

**PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance**

This is to certify that the thesis/dissertation prepared

By Abdallah Khreishah

Entitled Analysis and Design of Intersession Network Coding in Communication Networks

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

C-C. Wang, Co-Chair

Chair

X. Lin

N. B. Shroff, Co-Chair

T. J. Hacker

W. Szpankowski

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's Policy on Integrity in Research and the use of copyrighted material.

Approved by Major Professor(s): C-C. Wang

Approved by: V. Balakrishnan

Head of the Graduate Program

6/28/10

Date

**PURDUE UNIVERSITY
GRADUATE SCHOOL**

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation: Analysis and Design of Intersession Network Coding in
Communication Networks

For the degree of Doctor of Philosophy

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22*, September 6, 1991, *Policy on Integrity in Research*.*

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Abdallah Khreishah

Signature of Candidate

6/28/10

Date

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

ANALYSIS AND DESIGN OF INTERSESSION NETWORK CODING IN
COMMUNICATION NETWORKS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Abdallah Khreishah

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2010

Purdue University

West Lafayette, Indiana

UMI Number: 3444583

All rights reserved !

INFORMATION TO ALL USERS !

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion. !



UMI 3444583

Copyright 2011 by ProQuest LLC. !

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

To my parents especially my father who passed away while writing this dissertation.

ACKNOWLEDGMENTS

First of all, I thank my God for helping me to reach to this point. I thank him for providing me with all that I need, and for sustaining me throughout my graduate studies.

I would like to express my deepest gratitude to my co-advisors, Prof. Chih-Chun Wang and Prof. Ness B. Shroff for their constant encouragement, support, patience, and constructive criticism during my Ph.D study. This dissertation would be impossible without their guidance and support. They have provided me the first step towards being a successful researcher by revealing their experience in formulating problems, tackling these problems, and presenting them.

I am also grateful to Prof. Xiaojun Lin for his valuable comments and encouragement on my Ph.D work when he was a Ph.D student during my first semester as a graduate student and later as a member of my advisory committee. I am also privileged to have Prof. Wojciech Szpankowski and Prof. Tom Hacker in my doctoral committee and to work with Prof. Sonia Fahmy during my Masters study.

I would like to thank my officemates and friends I met at Purdue. Special thanks to Issa Khalil, Ibrahim Aly, JooHwan Kim, Gagan Gupta, Shahzada Rasool, Roman Chertov, Srikanth Hariharan, Sungoh Kwon, Venkata Ramanan, Changhee Joo, Sarah Sellke, Gyubum Kyung, Xiaohang Li, Longbi Lin, DongHoon Shin, Yan Wu, and Gaurav Sharma. I appreciate your support and help that makes my life at Purdue an enjoyable experience.

At this stage I do not forget all of my teachers during my life as student. Special thanks to my first teacher Ustadh Hussain.

Last but not least I would like to thank my parents and family. Without their support and progress, I would not have come this far in my life.

TABLE OF CONTENTS

	Page
LIST OF TABLES	ix
LIST OF FIGURES	x
ABSTRACT	xiv
1 INTRODUCTION	1
1.1 Overview of Network Coding	1
1.2 Intrasession and Intersession Network Coding	3
1.3 Optimization approaches for communication networks	5
1.3.1 Pairwise Intersession Network Coding	6
1.4 Optimization Based Approach for Pairwise Intersession Network Coding for Wireline Networks	8
1.5 Pairwise Intersession Network Coding for Wireless Networks with opportunistic scheduling	11
1.5.1 Path based approach	13
1.5.2 Back-pressure approach	13
1.6 Intersession Network Coding in Wireless Networks without Opportunistic Scheduling	13
2 RATE CONTROL WITH PAIRWISE INTERSESSION NETWORK CODING	15
2.1 Main Contribution	15
2.2 Settings and Preliminaries	16
2.2.1 System Settings	16
2.2.2 Superposition Approach	17
2.3 Problem Formulation	18
2.4 The Rate Control Algorithm	20
2.5 Implementation Details	24

	Page	
2.5.1	Collecting implicit costs	24
2.5.2	The Coding Scheme	25
2.5.3	Coding Scheme when $x_{ij}^l \neq x_{ji}^l$	28
2.5.4	Dealing with non-concave objective functions for real-time traffic	30
2.6	Complexity Reduction	30
2.6.1	Excluding redundant PICCs in the initialization step	30
2.6.2	Adaptive Algorithm	32
2.7	Simulation Results	35
2.7.1	Convergence	35
2.7.2	Gain and Fairness	37
2.7.3	Complexity	38
2.8	Conclusion	41
3	CROSS-LAYER OPTIMIZATION FOR WIRELESS MULTIHOP NETWORKS WITH PAIRWISE INTERSESSION NETWORK CODING	43
3.1	Main Contribution	43
3.2	Motivation	44
3.3	Preliminary Results	45
3.3.1	Analytical Framework for Wireless Multi-hop Networks — A Wireless to Wireline Conversion	45
3.3.2	Pairwise Intersession Network Coding for Wireless Networks	46
3.4	Optimal Joint Scheduling/Rate-Control With PINC	48
3.4.1	Convergence Analysis of Algorithm A	54
3.4.2	Stability of Algorithm A	54
3.5	Pairwise Intersession Network Coding with Imperfect Scheduling . .	55
3.5.1	Networks with Dynamic Arrivals and Departures	56
3.5.2	A Generalized Node Exclusive Interference Model With the WMA	57
3.6	Distributed Code Design for PINC	59
3.7	Numerical Experiments	63

	Page
3.8 Conclusion	67
4 BACK-PRESSURE ALGORITHM FOR FULL UTILIZATION OF WIRE- LESS MULTIHOP NETWORKS WITH PAIRWISE INTERSESSION NET- WORK CODING	69
4.1 Introduction and related work	69
4.2 Broadcast Link size with PINC	70
4.3 Problem Formulation	75
4.4 Energy minimization	80
4.4.1 Distributed Algorithm	80
4.4.2 Power Efficiency Ratio	92
4.5 Rate Control	93
4.6 Finding PICCs	95
4.7 Simulation Results	97
4.8 Conclusion	98
5 THE CAPACITY REGION OF 2-HOP RELAY NETWORK WITHOUT OPPORTUNISTIC SCHEDULING	101
5.1 Main Contribution	101
5.2 The Capacity region of two hop coding with two flows	102
5.2.1 The Setting	102
5.2.2 The Capacity Results	104
5.3 The Capacity of two hop coding with only XOR operations	108
5.3.1 The Settings	108
5.3.2 The Characterization	109
5.4 Two-hop Relay Network Simulations	112
5.4.1 The Variants of Optimal Intersession Network Coding (INC)	112
5.4.2 The Baseline Non-Coding Scheme and Its Variants	113
5.4.3 Schemes for More Than 2 Sessions	113
5.4.4 Practical, Low-Complexity Schemes for $M > 2$	114
5.4.5 Numerical Experiments	119

	Page
5.5 Conclusion	121
6 INTERSESSION NETWORK CODING FOR LOSSY WIRELESS MULTI-HOP NETWORKS	122
6.1 Main Contribution	122
6.2 Settings	122
6.3 Formulation without Intersession Network Coding	123
6.4 Extension to Intersession Network Coding	124
6.4.1 Single hop Coding Restrictions for Intersession Network Coding	124
6.4.2 The Capacity of 2-hop Relay Network	125
6.4.3 Formulation with Intersession Network Coding	128
6.5 Numerical Results	129
6.6 Conclusions	130
7 CONCLUSION AND FUTURE WORK	133
7.1 Summary of Contribution	133
7.2 Future Research Directions	135
7.2.1 Integrating Our Solution with TCP	135
7.2.2 Distributed Scheduling Algorithm for Pairwise Intersession Network Coding under more General Interference Models.	135
7.2.3 The Capacity Region of Single-hop Intersession Network Coding without Opportunistic Scheduling and More than Two sessions	136
7.2.4 More Complex Pairwise Intersession Network Coding	136
7.2.5 k -hop intersession network coding without opportunistic scheduling	137
7.2.6 Reduced Complexity and provably efficient coding scheme for 2-hop relay network with PEC model	138
7.2.7 Distributed Algorithms for Intersession Network Coding in lossy Multihop Wireless Networks.	139
LIST OF REFERENCES	140
A SUPPORTING RESULTS FOR CHAPTER 2	149

	Page
A.1 Notations used for the proof of Proposition 2.4.1	149
A.2 <i>Lemma A.2.1</i>	150
A.3 <i>Lemma A.3.1</i>	151
A.4 Proof of <i>Proposition 2.4.1</i>	153
A.5 Proposition A.5.1	155
A.6 Proof of Proposition 2.6.1	156
B SUPPORTING RESULTS FOR CHAPTER 3	158
B.1 Sketches of the proof of Proposition 3.5.1:	158
B.2 Sketches of the proof of Proposition 3.5.2:	159
B.3 Proof of Proposition 3.6.1	163
VITA	169

LIST OF TABLES

Table	Page
2.1 Rate R_i assigned for each session in Fig. 2.9 using routing and intersession network coding with the objective function $\sum_i \log_2(\delta + R_i)$ and different values of δ	39
B.1 The list of possible coding operations a node has to perform if the PICC satisfies Case 1	165
B.2 The list of possible coding operations a node has to perform if the PICC satisfies Case 3	165

LIST OF FIGURES

Figure	Page
1.1 The multicast butterfly network.	2
1.2 The butterfly network for two unicast sessions.	5
1.3 The butterfly topology and its capacity regions with and without network coding.	9
1.4 The grail topology and its capacity regions with and without network coding.	10
1.5 A cyclic butterfly network that satisfies condition 2 of Proposition 1.3.1 and the corresponding coding operations.	10
2.1 Applying both the random network coding and the pairwise random network coding to the grail structure.	27
2.2 (a) The assignment of the codes on the butterfly as an example of the l -th PICC between sessions i and j with unequal rates $x_{ij}^l = 3$ and $x_{ji}^l = 2$ during the transient state. (b) The upper sequence represents the sequence numbers of packets sent by s_i through the l -th PICC between sessions i and j , and the lower sequence represents the sequence numbers of packets sent by s_j through the same PICC. If a packet in the upper sequence is to be coded with another one in the lower sequence, there is a link between them. Packets without links are those for which no coding is performed.	29
2.3 Flow chart for the adaptive algorithm.	34
2.4 Convergence results for s_1 in the grail topology with different step sizes and K , the number of proximal iterations. Here, the rate corresponds to the optimal PICC.	36
2.5 Topology contains four source-sink pairs.	36
2.6 Convergence results for the topology in Fig. 3.8 with $\beta = 0.01$, and the number of proximal iterations $K = 5$. We plot the convergence rate vs. iterations for the optimal PICC and one insignificant PICC.	37
2.7 Gain for the topology in Fig. 3.8 with the objective function $\sum_i \log_2(\delta + R_i)$ and different values of δ	40

Figure	Page
2.8 Gain for the topology in Fig. 3.8 with the objective function $\sum_i \frac{R_i^{1-\sigma}}{1-\sigma}$ and different values of σ	40
2.9 Grid topology with three sessions	41
3.1 Modelling the wireless multicast advantage.	46
3.2 A simple wireless one-hop coding opportunity with two sessions: Session 1: $A \xrightarrow{X} C$; Session 2: $C \xrightarrow{Y} A$. (a) The slot-by-slot wireless transmission. (b) Wireless to wireline conversion. (c) Paths P_{s_1,d_1} , P_{s_2,d_2} , and P_{s_2,d_1} . (d) Paths Q_{s_2,d_2} , Q_{s_1,d_1} , and Q_{s_1,d_2}	47
3.3 The wireless cross flows with two sessions: Session 1: $A \xrightarrow{X} E$; Session 2: $B \xrightarrow{Y} D$. (a) The slot-by-slot wireless transmission. (b) Wireless to wireline conversion. (c) Paths P_{s_1,d_1} , P_{s_2,d_2} , and P_{s_2,d_1} . (d) Paths Q_{s_2,d_2} , Q_{s_1,d_1} , and Q_{s_1,d_2}	48
3.4 A new type of wireless cross flows with two sessions: Session 1: $A \xrightarrow{X} F$; Session 2: $C \xrightarrow{Y} D$. (a) The slot-by-slot wireless transmission. (b) Wireless to wireline conversion. (c) Paths P_{s_1,d_1} , P_{s_2,d_2} , and P_{s_2,d_1} . (d) Paths Q_{s_2,d_2} , Q_{s_1,d_1} , and Q_{s_1,d_2}	49
3.5 Illustration of the node exclusive model when the WMA is taken into consideration.	58
3.6 Illustration of an insignificant PICC that contains a pair of edge-disjoint paths as a strict subgraph.	60
3.7 Cases 1.1 to 1.4 of the new distributed code construction using only the binary XOR operation. M_1 and M_2 are the coded symbols along the two incoming edges of v . M_e is the outgoing coded symbol along edge e	62
3.8 The network topology, node locations, and three classes of unicast traffic used in the simulations.	64
3.9 The convergence results for the case of deterministic arrival with the linear SINR-based interference model.	65
3.10 The convergence results for the case of deterministic arrival with the $W \log(1 + SINR)$ -based interference model.	66
3.11 The number of users in the system versus the system load for the case of dynamic arrival with the linear SINR-based interference model.	67
4.1 Graphical representation of the different cases in the proof of Proposition 4.2.1	71

Figure	Page
4.2 The X-topology identified by the solid links with entry and exit nodes for session d_1 (d_2) being u_1, v_1 (u_2, v_2). The dotted lines represent the logical paths the information is flowing through. The logical paths might be non-coded paths or through PICCs	76
4.3 Four PICCs are using the X-topology. The four sets of coexisting sessions for these PICCs are (d_1, d_2) , (d_1, d_4) , (d_3, d_2) , (d_3, d_4) The dotted lines represent the logical paths the information is flowing through. The logical paths might be non-coded paths or through PICCs	77
4.4 Topology used to represent the linear formulation	78
4.5 A topology used to compare different capacity regions	80
4.6 A topology used to compare different capacity regions	80
4.7 Two I queues at node u for the received symbols X and Y in the input links and one O queue for the symbol Z sent through the output link. They grow up at possibly different times, but they shrink at the same time when a packet of the coded symbol Z is sent through the output link of u	84
4.8 A 3×3 grid topology used in the simulations.	98
4.9 Convergence for both the coded and non-coded versions of the CEEA Algorithm on a 3×3 grid topology.	99
4.10 The wireless butterfly topology used in the simulations	99
4.11 Convergence for both the coded and non-coded versions of the CRCA Algorithm.	100
5.1 Illustration of 2-session relay networks.	102
5.2 The overhearing probability versus the distance.	103
5.3 A broadcast PEC problem with side information.	105
5.4 (a) An example of random node placement for $M = 2$ sessions. (b) The constraint on the topological relationship between s_i and d_i	105
5.5 The average sum rates for different schemes combining INC, OpR, and CL.	116
5.6 The cumulative distribution function (CDF) of per topology improvement (PTI). The schemes in the legend corresponds to the curves from the rightmost to the leftmost when focusing on the intersection of CDF=0.6.	116
5.7 The average sum rates for different practical schemes without OpR. . .	117

Figure	Page
5.8 The PTI for 25 representative topologies. These 25 topologies have their sum rates being the 25-th quantile points when sorting the sum rates in the ascending order. The list of schemes in the legend corresponds to the curves from the topmost to the bottommost when focusing on the performance of of the 15/25-th quantile point.	118
6.1 An example of network with two flows to explain the single hop coding restrictions.	125
6.2 Average Gain of Intersession network coding for randomly generated topologies with respect to the number of sessions in the network. The Objective is to maximize the total throughput of the network.	130
6.3 Average Gain of Intersession network coding for randomly generated topologies with respect to the number of sessions in the network. The Objective is to achieve strict fairness among the flows.	131
7.1 An example where the superposition approach in Chapter 2 does not achieve the capacity of the network.	137
7.2 An example to illustrate the importance of characterizing intersession network coding for more than one hop. Every link has a delivery probability of 0.5.	138

ABSTRACT

Khreishah, Abdallah Ph.D., Purdue University, August 2010. Analysis and Design of Intersession Network Coding in Communication Networks. Major Professors: Chih-Chun Wang and Ness B. Shroff.

Network coding extends the functionality of networking nodes beyond the traditional store-and-forward operations. It allows information from different packets/flows to be mixed together at intermediate nodes using mathematical operations. There has been significant prior work that has explored “intrasession” network coding, where only packets of the same session or user are allowed to be mixed together. However, intrasession network coding, while providing substantial gains in the presence of multicast traffic, provides no performance gains for unicast traffic, which is the predominant traffic on the Internet. On the other hand, intersession network coding, where packets of different sessions or users are allowed to be mixed together, improves the capacity of the network in the presence of either unicast or multicast traffic. In this dissertation, we use optimization and information theoretic approaches to design intersession network coding schemes for both wireline and wireless networks. In wireline networks we design a distributed but joint rate control algorithm and coding scheme that are optimal when any coded symbol is formed by at most two original symbols. The proposed approach improves both the capacity and fairness over existing approaches. Due to their broadcast nature, wireless networks enjoy higher gains from network coding than their wireline counterpart. We explore two different wireless network settings in this dissertation and develop optimal intersession network coding schemes under these settings. The first setting is when channel conditions are known prior to transmission. In this case we develop a cross-layer framework that requires minimal interaction between layers and achieves the optimal solution when any

coded symbol is formed by at most two original symbols. We also provide a coding scheme that uses only XOR operations and achieves the same throughput as the one developed for wireline networks. We study the performance loss of our framework when using non-optimal but distributed scheduling algorithms. This framework is also extended to include energy minimization. In the second setting we assume that channel conditions are unknown prior to transmission. Without network coding the optimal solution can be achieved using opportunistic routing. With network coding we study the problem of two-hop relay network where the encoding and decoding nodes are neighbors. For this problem we provide a coding scheme that achieves the capacity of the network with two flows. We also use this result to maximize the throughput of a general lossy wireless network.

1. INTRODUCTION

1.1 Overview of Network Coding

The traditional way of sending information through computer networks is to organize them into data packets at the source and route the packets to the sink through intermediate nodes. This method is referred to as *store-and-forward* or *packet switching* [1]. We will refer to this method as the non-coded method. For the case of single source transmitting to a single sink, the maximum achievable rate equals to the min-cut max-flow value between the source and the sink. The min-cut max-flow is defined in [2]. For this unicast model, it is shown in [3] that the min-cut max-flow value is achievable by the non-coded method. In the case of a single source broadcasting information to more than one sink - known as single source multicast - non-coded method is not sufficient to achieve the maximum rate [4]. Furthermore, with the non-coded approach the problem is equivalent to steiner-tree packing [5] which is known to be NP-hard.

Ahlsweede et.al in [4] show that for the single source multicast case, the maximum achievable multicast rate is the minimum of the min-cut max-flow values between the source and each sink. It was also shown in [4] that this rate is achievable by coding. This type of coding is referred to as *network coding*. For example, in the butterfly network in Figure 1.1, we assume that each link can sustain a throughput of at most 1 packet per unit time. Source s wants to broadcast packets to sinks d_1 and d_2 , respectively. For this topology, if only non-coded solutions are permitted, we could achieve a multicast rate of 1.5 packets per unit time. However, using network coding, it can be easily seen that we can sustain a rate of 2 packets per unit time as shown by the coding scheme in the figure.

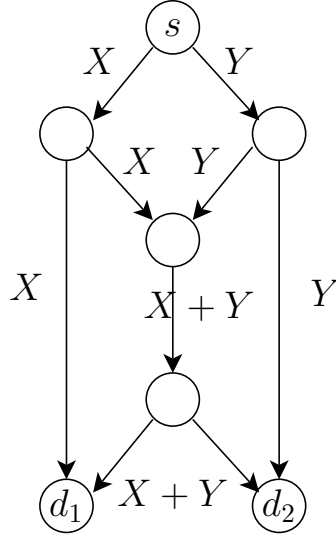


Fig. 1.1. The multicast butterfly network.

The simplicity of the single source multicast problem is due to two facts. First, Linear coding is sufficient to achieve the min-cut max-flow rate [6]. Second, there exists a polynomial time algorithm for the code construction [7]. A useful algebraic approach to network coding is found in [8]. Based on this approach, it was shown in [9] that choosing the coding coefficients randomly with a large enough field size achieves very close to optimal solution. The results in [9] facilitate distributed implementation of network coding.

The work on network coding conducted after these basic works are mainly in three different directions. The first direction is to provide bounds on the benefits of network coding and to study the capacity improvements provided by network coding. In [10], both graph and information theoretic characteristics of the network are used to provide upper bounds on the capacity of the network with multiple sources and sinks. Edge cut sets are used in [11, 12] for the same purpose. In [13] the capacity of the network is characterized when the number of nodes allowed to perform coding is limited and [14] studied the capacity of multiple unicast sessions in undirected networks. For wireless networks [15] showed that in the asymptotic case when the

number of nodes in the unit area becomes very large the gain of network coding in terms of throughput is bounded by a constant. The benefits of network coding in terms of both throughput and energy are also within a constant factor for the non-asymptotic case [16].

The second direction is to provide distributed algorithms that allocate resources at different links in the network. For wireline networks a distributed algorithm for the problem of minimum cost single-source multicast is developed in [17]. The same work characterizes the problem for wireless networks as a linear program. The congestion control problem of the single-source multicast for wireline networks is studied in [18–21], where different distributed algorithms are developed. A fast resource allocation algorithm is developed in [22] based on network coding with feedback. For wireless networks cross-layer algorithms for multicast are developed in [23–25], and opportunistic network coding algorithms are developed in [26, 27] for unicast sessions and in [28] for both unicast and multicast sessions.

The third direction is using network coding for problems other than resource allocation as peer-to-peer networks in [29, 30], content distribution [31], distributed storage [32], network tomography [33], code updates in sensor networks [34], and network security [35, 36].

1.2 Intrasession and Intersession Network Coding

When there are multiple sources in the network, the single source multicast problem can be generalized to *intrasession network coding*. The session is defined by a source and a set of sinks interested in receiving symbols from this source. In intrasession network coding, coding operations are limited to be between symbols of the same session. Intrasession network coding have demonstrated throughput improvement for multicast in both wireline [17–21, 28] and wireless networks [23–25]. The benefits of Intrasession network coding can also be in terms of energy savings [37]. When integrated with feedback messages in [22], intrasession network coding results in a faster

resource allocation algorithm than the back-pressure algorithm. In general, intrasession network coding does not achieve the capacity. Furthermore, if every session has only one sink, which is the typical scenario in communication networks referred to as *multiple-unicast-sessions*, the performance of non-coded solutions is the same as intrasession network coding. Therefore, in order to enhance the performance of communication networks, coding operations have to be performed among the symbols of different sessions which is referred to as *intersession network coding*.

In contrast to intrasession network coding, intersession network coding is less studied and understood. One reason for that is the result in [38] which shows that the general intersession network coding problem is NP-hard as the number of sessions becomes very large. Another reason for that is that in general linear coding is insufficient for the intersession network coding problem [39].

Different approaches have been used to study intersession network coding for both wireline and wireless networks. For wireline networks, the butterfly structure shown in Fig. 1.2, has been exploited in [40,41] to obtain a capacity region that is bigger than the non-coded capacity region. A game theoretic analysis of the butterfly capacity region is conducted in [42]. Genetics algorithms have been used in [43] to obtain another achievable capacity region for intersession network coding in wireline networks. For wireless networks, one approach is to use the butterfly to provide a good achievable capacity region as in [40,41,44]. Another approach is to limit the operations to be XOR and the decoding node to be the next hop of the encoding node as in [26,27]. Nonetheless, the butterfly is not the only structure that provide intersession network coding benefits and the approaches in [26,27] do not obtain the capacity region for single hop intersession network coding. In this dissertation we propose for wireline and wireless networks approaches that can achieve the capacity or find all intersession network coding opportunities under some assumptions and hence provide performance enhancement over existing techniques.

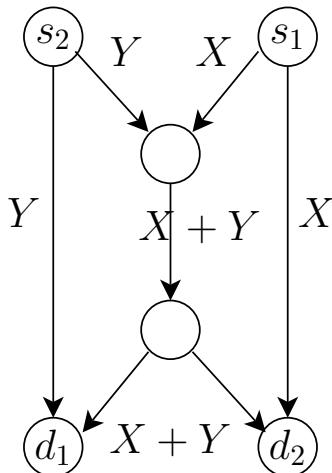


Fig. 1.2. The butterfly network for two unicast sessions.

1.3 Optimization approaches for communication networks

Optimization techniques are powerful tools that are used to allocate resources for the network. The objective of these optimization approaches is to come up with simple operations that can be performed in a distributed way at the network nodes and to show that these operations achieve the optimal solution of the given optimization objective. The optimization objective can be rate control, fairness, throughput maximization, energy or delay minimization, or any other objective function.

The problem of flow control in wireline networks with single path routing is studied in [45, 46]. The duality approach is used to model and study TCP as an optimization problem in [47–49]. The congestion control problem is extended for the multi-path routing case in [50, 51] and for the multicast case in [52]. Optimization techniques have been also used for network coding in [17–20]. For single-hop wireless networks, opportunistic scheduling have been modelled and solved as an optimization problem in [53]. For multihop wireless networks cross-layer frameworks for rate control have been proposed in [54–59]. In wireless multihop networks optimization techniques have been also used for energy minimization problems [60–62] and for network coding [23,

25,27,37]. There are several books that discuss the optimization approaches and their applications in communication networks as [63–65]. In this dissertation we will use optimization approaches to design efficient resource allocation methods for different network settings with intersession network coding.

1.3.1 Pairwise Intersession Network Coding

Since the general intersession network coding problem is NP-hard, and because linear coding is not sufficient for the problem as discussed in Section 1.2, more restrictions need to be put on the problem to make it more tractable. One of these restrictions is to limit the number of sessions to be two and to limit any coded symbol to be a result of at most two original symbols. We call this settings *pairwise intersession network coding*. In [66,67] pairwise intersession network coding was characterized. We use this characterization in Chapters 2 and 3 to develop optimal algorithms for both wireline and wireless networks. In the following we review the characterization of pairwise intersession network coding from [66,67].

Consider directed cyclic/acyclic wireline network $G = (V, E)$, in which each edge is able to carry one $\text{GF}(q)$ symbol per unit time (say a second) and the propagation delay is also one second. High-rate links are modelled by parallel edges and long-delay links are modelled by long paths with added auxiliary intermediate nodes. A pair of coexisting unicast sessions (s_1, d_1) and (s_2, d_2) would like to transmit two strings of independently distributed $\text{GF}(q)$ symbols X_1, \dots, X_T and Y_1, \dots, Y_T (one string for each session) simultaneously over a given duration of T seconds. Pairwise intersession network coding (PINC) is allowed and packets of these two strings $\{X_t, Y_t : t = 1, \dots, T\}$ can be arbitrarily mixed in a linear or non-linear fashion.

We say a PINC solution exists for transmitting two rate-1 strings of packets (over the given unit-edge-capacity network), if given any $\epsilon > 0$, there exists a sufficiently large T such that

$$\frac{1}{T}I([X]_1^T; [M_{d_1}]_1^T) > (1 - \epsilon) \log(q)$$

and

$$\frac{1}{T}I([Y]_1^T; [M_{d_2}]_1^T) > (1 - \epsilon) \log(q),$$

where $[X]_1^T \triangleq \{X_1, \dots, X_T\}$, $[Y]_1^T \triangleq \{Y_1, \dots, Y_T\}$, $I(\cdot; \cdot)$ is the mutual information and $[M_{d_i}]_1^T$ is the symbols received by destination d_i for $i = 1, 2$.

For the following, we use $P_{u,v}$ to represent a path connecting nodes u and v .

Proposition 1.3.1 *A PINC solution exists if and only if one of the following two conditions holds.*

- *Condition 1: There exist two edge-disjoint paths P_{s_1, d_1} and P_{s_2, d_2} .*
- *Condition 2: There exist six paths grouped into two sets $\mathcal{P} = \{P_{s_1, d_1}, P_{s_2, d_2}, P_{s_2, d_1}\}$ and $\mathcal{Q} = \{Q_{s_2, d_2}, Q_{s_1, d_1}, Q_{s_1, d_2}\}$ such that for all $e \in E$,*

$$1_{\{e \in P_{s_1, d_1}\}} + 1_{\{e \in P_{s_2, d_2}\}} + 1_{\{e \in P_{s_2, d_1}\}} \leq 2$$

and

$$1_{\{e \in Q_{s_2, d_2}\}} + 1_{\{e \in Q_{s_1, d_1}\}} + 1_{\{e \in Q_{s_1, d_2}\}} \leq 2,$$

where $1_{\{\cdot\}}$ is the indicator function.

If condition 1 of the proposition is satisfied, a non-coding solution is sufficient for the problem. On the other hand, if only condition 2 is satisfied, network coding is necessary to achieve simultaneous rate-1 transmission. For example, the acyclic network in Fig. 1.3(a) satisfies condition 2 by choosing $P_{s_1, d_1} = s_1 v_1 v_3 v_4 v_6 d_1$, $P_{s_2, d_2} = s_2 v_2 v_3 v_4 v_5 d_2$, $P_{s_2, d_1} = s_2 v_2 v_6 d_1$, $Q_{s_1, d_1} = s_1 v_1 v_3 v_4 v_6 d_1$, $Q_{s_2, d_2} = s_2 v_2 v_3 v_4 v_5 d_2$, and $Q_{s_1, d_2} = s_1 v_1 v_5 d_2$. A unit rate can be supported between (s_1, d_1) and (s_2, d_2) , if

the coding scheme represented in the figure is used. Another example where only condition 2 is satisfied is the grail network in Fig. 1.4(a) with

$$\begin{aligned}
 P_{s_1,d_1} &= s_1v_2v_3v_4v_5d_1, P_{s_2,d_2} = s_2v_1v_2v_3v_6d_2, \\
 P_{s_2,d_1} &= s_2v_1v_4v_5d_1, Q_{s_1,d_1} = s_1v_2v_3v_4v_5d_1, \\
 Q_{s_2,d_2} &= s_2v_1v_4v_5v_6d_2, Q_{s_1,d_2} = s_1v_2v_3v_6d_2.
 \end{aligned} \tag{1.1}$$

Fig. 1.5 contains a cyclic network that satisfies condition 2 of the proposition by choosing $P_{s_1,d_1} = s_1v_7v_6v_5v_2v_3v_4d_1$, $P_{s_2,d_1} = s_2v_1v_4d_1$, $P_{s_2,d_2} = s_2v_1v_2v_3v_6v_5v_8d_2$, $Q_{s_1,d_1} = s_1v_7v_6v_5v_2v_3v_4d_1$, $Q_{s_1,d_2} = s_1v_7v_8d_2$, and $Q_{s_2,d_2} = s_2v_1v_2v_3v_6v_5v_8d_2$. The corresponding coded symbols carried by each edge are also illustrated in Fig. 1.5, and rate-1 is sustainable for both unicast sessions (s_1, d_1) and (s_2, d_2) . Note that the (ϵ, T) is essential to take into account the delay of each edge as seen in Figs. 1.3(a), 1.4(a) and 1.5. Proposition 1.3.1 shows that as the existence of non-coding solutions is equivalent to finding edge-disjoint paths, the existence of PINC solutions is equivalent to finding paths with *controlled edge overlap*. The intuition is that when the packets/paths are not overly using any bottleneck edge (those edges used by three paths), coding enables the information to be transmitted simultaneously for both sessions. The subgraph G' induced by any six paths satisfying condition 2 of Proposition 1.3.1 will be referred as a pairwise intersession coding configuration (PICC). *In a broad sense, a path is the smallest “graph unit” for non-coding multiple session communications while a PICC is the smallest “graph unit” when coding across two sessions is permitted.*

Next we will give an overview of our contribution in this dissertation.

1.4 Optimization Based Approach for Pairwise Intersession Network Coding for Wireline Networks

Fairness among sessions can be modelled by associating a utility function with each session. Maximizing the sum of these utility functions achieves specific type of fairness as shown in [68]. Based on this modelling, distributed algorithms that

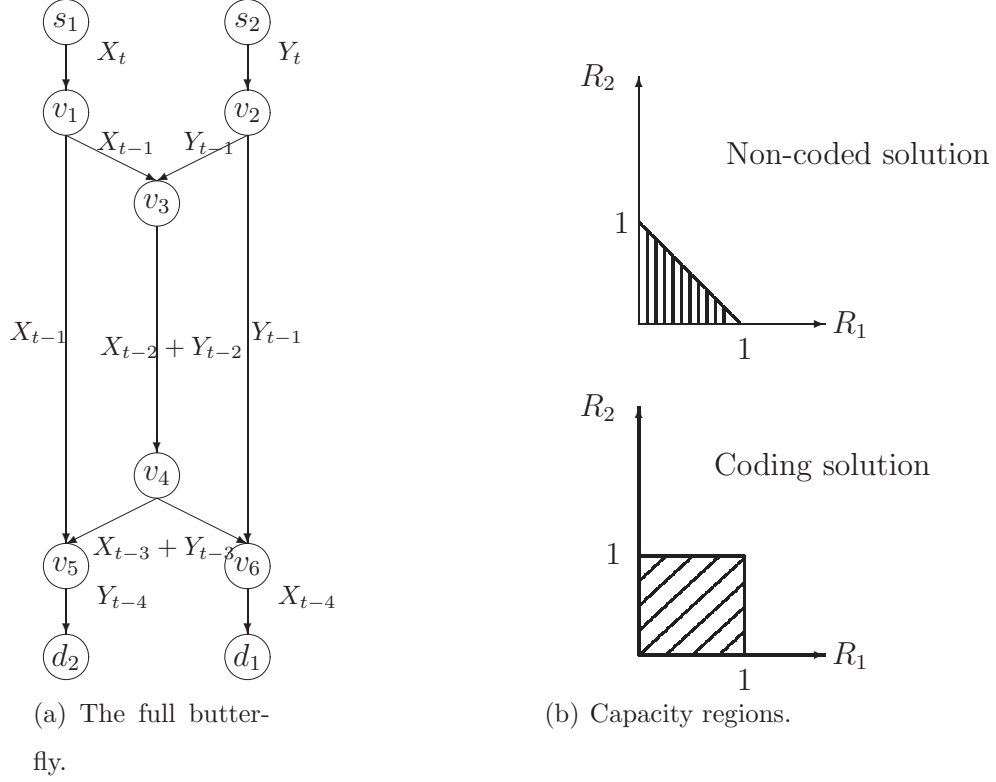


Fig. 1.3. The butterfly topology and its capacity regions with and without network coding.

achieve fairness have been developed for the non-coded solution as in [45, 51, 69, 70], which were extended for intrasession network coding in [18–20, 71]. In Chapter 2 we extend these results to *pairwise intersession network coding*, i.e. the setting where any coded packet can be generated from only two original packets. Based on the PINC results, we represent the network with arbitrary number of sessions as a superposition of intersession coded traffic and non-coded traffic. The intersession coded traffic satisfies the necessary and sufficient conditions for PINC. The rate control algorithm we obtain is a fully distributed queue length algorithm that selects good paths in the network and perform coding along these paths. Due to its path based nature, the complexity of our scheme is much lower than that of other existing schemes that search for butterfly structures in the network [40, 41]. Simulation results also confirm

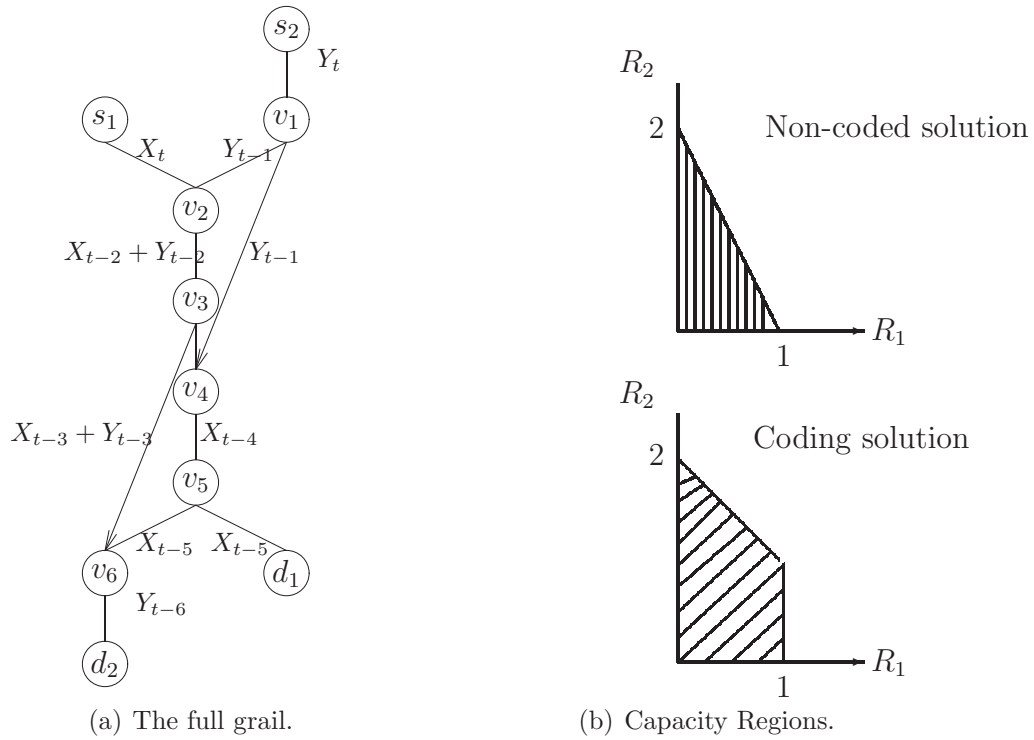


Fig. 1.4. The grail topology and its capacity regions with and without network coding.

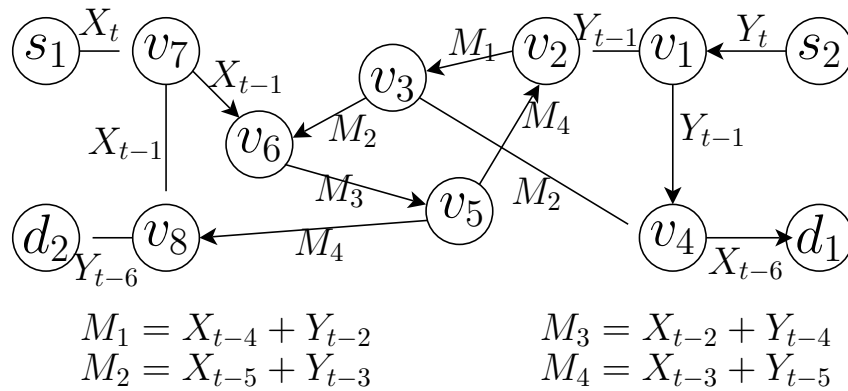


Fig. 1.5. A cyclic butterfly network that satisfies condition 2 of Proposition 1.3.1 and the corresponding coding operations.

the benefits of the proposed approach over the path based one in terms of fairness and throughput. We also provide in Chapter 2 a distributed coding scheme for pairwise intersession network coding that perform random coding operations on all the links except for selected ones. Decoding operations are performed on the selected links and the determination of these selected links are done in a distributed way. The proposed coding scheme achieves the optimal solution with a moderate field size, typically 2^{16} .

1.5 Pairwise Intersession Network Coding for Wireless Networks with opportunistic scheduling

One of the key features in wireless multihop networks is their broadcast nature. Due to this feature, wireless networks observe larger gain from network coding than their wireline counterpart. If unicast sessions are present in the wireless network, most coding opportunities involve only two or three sessions as coding across many sessions requires greater transmission power to broadcast the coded symbol to many receivers, which enhances interference as observed in [26, 27]. Therefore, in Chapters 3 and 4 we study cross-layer optimization for wireless multihop networks with pairwise intersession network coding. Pairwise intersession network coding as defined in Section 1.4 means that any coded packet is made of at most two original packets. The assumption we make here is the ability for any node to perform opportunistic scheduling, i.e., the channel conditions are known prior to transmission. In Chapter 5 we remove this assumption and design suitable intersession network coding schemes.

The solutions we have in Chapters 3 and 4 are cross-layer solutions that satisfy the loose coupling principle between layers, introduced in [54]. This means that while the solutions jointly optimize through all the network layers, they do not lose modularity by having a separate optimization problem for each layer and minimum number of variables are used across the layers. The provided solutions can be considered as extensions of the results in Chapter 2 to wireless networks. However, these extensions are not trivial due to the unique features of wireless networks. Therefore, the solutions

take into account the transmission power, broadcast nature, scheduling, and rate control.

Two basic principles are used to develop the solutions in Chapters 3 and 4. The first principle is the wireless to wireline conversion. The conversion is performed by introducing for each original node a set of auxiliary nodes, each auxiliary node is connected by directed links to a set of the next-hop nodes of the original node. Therefore, designing a wireless transmission scheme that exploits the broadcast advantages of wireless networks is equivalent to designing a good routing/scheduling algorithm on its wireline counterpart with the additional node-exclusive scheduling constraints that auxiliary nodes corresponding to the same auxiliary node cannot be active simultaneously. The wireless to wireline conversion is used to allocate rates for individual nodes. The second principle is that of a hyperlink. The hyperlink is defined as the set of auxiliary links adjacent to an auxiliary node. The hyperlink principle is used to make scheduling decisions. Therefore, the scheduling algorithms we obtain consider hyperlinks instead of links.

Wireless networks have limited processing capacity. Therefore, any developed coding scheme should not use complex mathematical operations. We design a distributed XOR based coding scheme that uses the knowledge of the nodes about the paths they belong to to decide whether to XOR two incoming packets or to forward only one of them. The performance of the proposed coding scheme is the same as that of the coding scheme developed in Chapter 2, i.e., it achieves the optimal solution for pairwise intersession network coding.

We have used two approaches to develop distributed algorithms for wireless multi-hop networks with pairwise intersession network coding. These are, path based and back-pressure approaches. In the following, we discuss the unique features of these approaches.

1.5.1 Path based approach

In the path based approach presented in Chapter 3, paths are found before running the algorithm and the transmission operations should be performed on these paths when running the algorithm. This approach permits us to analyze the performance loss of rate control for both deterministic and stochastic arrivals due to the use of suboptimal but distributed and less complex scheduling algorithms. This analysis is important because optimal scheduling policies are always centralized with high complexity.

1.5.2 Back-pressure approach

In this approach presented in Chapter 4, the routes that the packets follow are not predetermined and decided dynamically while the algorithm is running. Here, we allow the decoded packet to be re-encoded again in contrast to the path-based approach. This provides a larger capacity region. It is computationally expensive to find all the pairwise intersession network coding opportunities in the network. Therefore, we specify a parameter κ and allow coding to be performed within κ -hops. The parameter κ is an arbitrary integer, balancing the tradeoff between complexity and the achievable capacity. The back-pressure approach allows the extension of our scheme to energy minimization framework and studying the increase of power consumption due to the use of imperfect scheduling. The back-pressure approach on the other hand incurs more delay and takes more time to converge as observed in [72–74].

1.6 Intersession Network Coding in Wireless Networks without Opportunistic Scheduling

In Chapter 5 we remove the assumption of opportunistic scheduling. This is a more realistic setting, because the channel states change rapidly over time. With

this assumption the wireless to wireline conversion in Chapters 3 and 4 does not work, because the nodes cannot control which one of their neighbors receive the transmitted packet. Without network coding the optimal solution can be achieved using opportunistic routing introduced in [75]. Intrasession network coding can be combined with opportunistic routing as in [28].

Most of previous works on intersession network coding either ignore the assumption of no opportunistic scheduling [26, 76, 77] or design suboptimal policies under the no opportunistic scheduling case [27]. These works were conducted under the *single-hop intersession network coding* settings, i.e., the encoding and decoding nodes are one-hop away from each other. In Chapter 5, we provide a coding scheme for single-hop intersession network coding with the erasure channel model that achieves the capacity of the network with two flows. The capacity region obtained by the provided coding scheme can be characterized by linear equations. The coding scheme is not limited to XOR operations and uses random network coding. In Chapter 6 we use the capacity results to enhance the throughput of a general lossy wireless multihop network.

The rest of this dissertation is organized as follows: In Chapter 2, we develop a distributed rate control algorithm and distributed coding scheme for pairwise intersession network coding. We then turn to wireless networks in Chapter 3 and develop a path based cross layer rate control algorithm for wireless multi-hop networks with pairwise intersession network coding and the associated XOR-based coding scheme. In Chapter 4 we use the back-pressure technique to obtain distributed algorithms for rate control and energy minimization. Chapter 5 presents coding schemes for single hop intersession network coding when the opportunistic scheduling assumption is removed and Chapter 6 uses these results for multihop wireless networks. We briefly summarize our work in this dissertation and provide future research directions in Chapter 7.

2. RATE CONTROL WITH PAIRWISE INTERSESSION NETWORK CODING

2.1 Main Contribution

Wireline network is the simplest multihop network model. This is due to the absence of interference and the high delivery rate of links. Due to this fact, we start our study of intersession network coding by considering a wireline network with arbitrary number of sessions. We represent the network by a superposition of PICCs and routing paths with non-coded traffic to obtain a larger capacity region than existing works. Our main contributions in this chapter are as follows:

1. The development of a distributed algorithm with rate control and utility maximization for *intersession* network coding for multiple unicast flows, which can be easily generalized for the case of multiple multicast flows [78]. Our results show that the utility-optimization-based rate-control algorithm, originally designed for non-coded transmissions [45, 69, 70, 79] and later generalized for *intrasession network coding* [18–20, 71, 80, 81] can be extended to *pairwise intersession network coding* for the first time. This is a non trivial generalization considering the characteristic difference between inter and intrasession network coding.
2. Our result is developed based on finding good paths rather than finding specific structures in the network (such as the butterfly structures in [82]). This enables more efficient solutions since one can leverage upon existing work on how to choose good paths through the network. Further, we show empirically that the capacity region obtained via our approach can be considerably larger than those obtained via the pattern search algorithm [40, 41, 82].

3. A pairwise random coding scheme is proposed, which is a modified version of the random linear coding scheme in [9]. The pairwise random coding scheme decouples the coding and rate-control decisions and facilitates the development of a fully distributed algorithm. Combining the distributed rate control and the decentralized coding scheme, we eliminate unnecessary queue length information exchange among intermediate nodes, which results in improved efficiency of the overall scheme compared to the back-pressure algorithms in [41] and [40].

The rest of the chapter is organized as follows. In Section 2.2, the graph theoretic characterization of pairwise intersession network coding is reviewed for completeness. In Section 2.3, we describe the system settings and the formulation of our optimization problem. In Section 2.4, we solve the dual problem to obtain the optimal distributed rate control algorithm for pairwise intersession network coding. Several practical implementation issues are in Section 2.5 including the pairwise random coding scheme. In Section 2.6 we propose two approaches to reduce the complexity of the rate control algorithm. Section 2.7 is devoted to simulation results. We conclude the chapter in Section 2.8.

2.2 Settings and Preliminaries

2.2.1 System Settings

We model the network by a directed graph $G = (V, E)$, where V and E are the sets of all nodes and links, respectively. We use $\text{In}(v)$ to represent the set of all incoming links to node v and $\text{Out}(v)$ to represent the set of all outgoing links from node v . Two types of graphs are considered depending on the corresponding edge capacity: graphs with integral edge-capacity and graphs with fractional edge-capacity. For the former type, each edge has unit capacity and carries either one or zero packet per unit time. (No fractional packets are allowed.) For the latter type, each edge e has a fractional capacity, denoted by C_e , and can transmit at any rate between 0 and C_e . An integral graph models the packet-based transmission in a network, for which a high-rate link is

represented by parallel edges. On the other hand, the fractional graph can be viewed as a time-averaged version of the integral graph, which focuses on the “transmission rates” rather than the packet-by-packet behavior. For the rate-control algorithm in this chapter, we use the fractional graph model. When discussing the detailed coding operations among different packets, we use the integral graph model.

For networks modelled by fractional graphs, the rate control problem is defined by the set of tuples $(s_i, d_i, U_i(R_i))$ $i \in 1, 2, \dots, N$, where N is the number of coexisting unicast sessions. s_i and d_i are the source and destination nodes of session i and $U_i(\cdot)$ is the utility function of session i that is concave and monotonically increasing. R_i is the transmission rate supported in the i -th session.

Some graph-theoretic definitions will also be used in this work. We use $P_{v,w}$ or $Q_{v,w}$ to denote paths from nodes v to w . Here, we used two different notations to describe a path from u to v to make the characterization in Proposition 1.3.1 easier to understand. We use \mathcal{P} to represent a set of paths.

2.2.2 Superposition Approach

Proposition 1.3.1 serves as the building foundation of intersession network coding over pairs of unicast sessions.

Consider N source-&-sink pairs and each source s_i would like to transmit at rate R_i packets per unit time to the corresponding sink d_i over a fractional directed graph. The rate vector (R_1, \dots, R_N) is feasible if the original graph G can be viewed as the superposition of one graph G' and many PICCs such that (i) non-coded transmission is performed for every (s_i, d_i) pair in G' , (ii) pairwise linear network coding across (s_i, d_i) and (s_j, d_j) , $i \neq j$ is performed in each PICC individually, and (iii) the transmission rates (R_1, \dots, R_N) can be supported. Here, R_i is the sum of the non-coding transmission rate between s_i and d_i through G' and all the rates supported in any PICC where intersession network coding is performed between session i and some other session j .

Based on this superposition principle the above construction describes the achievable rate region of *Pairwise Intersession Network Coding* (PINC). Figs 1.3(b) and 1.4(b) compare the capacity regions obtained using the superposition approach with the non-coded capacity regions for both the butterfly and the grail. For these two topologies the capacity region obtained using the superposition approach is the capacity region of the network. In the next section we will describe the corresponding PINC achievable rate region by a set of constraints.

2.3 Problem Formulation

Since in the PINC region, the rate R_i is expressed as the sum of rates with/without intersession network coding, two sets of parameters and variables will be used in our formulation. Some parameters and variables are for the non-coded transmission and the others capture the intersession network coding performed on the PICCs. For the non-coded transmission, we define the parameters \mathcal{P}_i and $H_i^k(e)$, and the variable x_i^k . Let \mathcal{P}_i represent the collection of all paths between s_i and d_i . If link e is used by the k -th path between s_i and d_i , where k ranges from 1 to $|\mathcal{P}_i|$, then the indicator function $H_i^k(e) = 1$. Otherwise it is set to zero. x_i^k represents the uncoded transmission rate supported through the k -th path between s_i and d_i in G' . For the coded transmission through the PICCs, we define the parameters $\mathcal{P}(i, j)$, and $\mathcal{E}_{ij}^p(e)$, and the variable x_{ij}^{pm} . $\mathcal{P}(i, j)$ is the set of all tuples containing all possible choices of paths $\{P_{s_i, d_i}, P_{s_j, d_j}, P_{s_j, d_i}\}$. Because each PICC contains two sets of 3 paths, therefore, any PICC between sessions i and j can be indexed by p and m jointly, where the p and m means that the p -th tuple in $\mathcal{P}(i, j)$ and the m -th tuple in $\mathcal{P}(j, i)$ are used to generate the PICC of interest. The rate supported for sessions i and j over that PICC is denoted by x_{ij}^{pm} . We also define \vec{x} as a column vector containing x_i^k , $\forall i, k$ and x_{ij}^{pm} , $\forall i, j, p, m$. Therefore, the total supported rate for session i becomes

$$R_i = \sum_{k=1}^{|\mathcal{P}_i|} x_i^k + \sum_{j:j \neq i} \sum_{p=1}^{|\mathcal{P}(i,j)|} \sum_{m=1}^{|\mathcal{P}(j,i)|} x_{ij}^{pm}.$$

Consider a specific link e . The capacity consumed by pure routing traffic is: $\sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e)x_i^k$. For the PICC between sessions i and j , indexed by p and m , the capacity consumed by the path selection \mathcal{P} is $\mathcal{E}_{ij}^p(e)x_{ij}^{pm}$, where $\mathcal{E}_{ij}^p(e)$ is defined in the following manner:

$$\mathcal{E}_{ij}^p(e) = \begin{cases} 0 & \text{if no path in the } p\text{-th tuple in } \mathcal{P}(i, j) \\ & \text{uses link } e \\ 1 & \text{if 1 or 2 paths in the } p\text{-th tuple in} \\ & \mathcal{P}(i, j) \text{ use link } e \\ 2 & \text{if 3 paths in the } p\text{-th tuple in } \mathcal{P}(i, j) \\ & \text{use link } e. \end{cases}$$

This is because by Proposition 1.3.1, successful pairwise network coding requires that $1_{\{e \in P_{s_1, d_1}\}} + 1_{\{e \in P_{s_2, d_2}\}} + 1_{\{e \in P_{s_2, d_1}\}} \leq 2$. If all three paths in \mathcal{P} use link e , then the traffic along these three paths must use two parallel edges instead of a single one. Otherwise, $1_{\{e \in P_{s_1, d_1}\}} + 1_{\{e \in P_{s_2, d_2}\}} + 1_{\{e \in P_{s_2, d_1}\}} = 3$, which violates the necessary condition for pairwise intersession network coding. The same argument holds for the traffic along the paths in \mathcal{Q} , the m -th tuple in $\mathcal{P}(j, i)$, for which the network coded traffic consumes $\mathcal{E}_{ji}^m(e)x_{ij}^{pm}$. From the above reasoning, the total capacity consumed by intersession coding for the PICC between sessions i and j , indexed by p and m is the maximum of the two which is formally expressed as $\max(\mathcal{E}_{ij}^p(e), \mathcal{E}_{ji}^m(e))x_{ij}^{pm}$. Summing over all pairs of sessions $i \neq j$, and all p -th and m -th tuples of $\mathcal{P}(i, j)$ and $\mathcal{P}(j, i)$, the total capacity consumed by intersession network coding becomes $\sum_{(i,j):i < j} \sum_{p=1}^{|\mathcal{P}(i,j)|} \sum_{m=1}^{|\mathcal{P}(j,i)|} \max(\mathcal{E}_{ij}^p(e), \mathcal{E}_{ji}^m(e))x_{ij}^{pm}$.

Let \mathcal{PICC}_{ij} represent the collection of all PICCs between sessions i and j . For simplicity we use x_{ij}^l instead of x_{ij}^{pm} , where l is the index indicating that the l -th PICC of \mathcal{PICC}_{ij} is used. Since any union of the p -tuple and the m -th tuple of $\mathcal{P}(i, j)$ and $\mathcal{P}(j, i)$ can be mapped to the l -th PICC between i and j . We can also define

$$H_{ij}^l(e) = \frac{1}{2} \max(\mathcal{E}_{ij}^p(e), \mathcal{E}_{ji}^m(e)). \quad (2.1)$$

From the above discussion, the following constraints represent the PINC capacity region.

$$\sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e) x_i^k + 2 \sum_{(i,j):i<j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} H_{ij}^l(e) x_{ij}^l \leq C_e, \quad \forall e \in E \quad (2.2)$$

$$x_{ij}^l = x_{ji}^l, \quad \forall i < j, l. \quad (2.3)$$

Thus, our optimization problem becomes:

$$\max_{\vec{x} \geq 0} \sum_{i=1}^N U_i \left(\sum_{k=1}^{|\mathcal{P}_i|} x_i^k + \sum_{j:i \neq j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} x_{ij}^l \right) \quad (2.4)$$

subject to \vec{x} satisfying (2.2) and (2.3).

By change of variable indices i and j we have

$$\sum_{(i,j):i<j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} H_{ij}^l(e) x_{ij}^l = \sum_{(i,j):j<i} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} H_{ji}^l(e) x_{ji}^l.$$

Since $x_{ij}^l = x_{ji}^l$ according to (2.3), the constraints in (2.2) can be rewritten as:

$$\sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e) x_i^k + \sum_{(i,j):i \neq j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} H_{ij}^l(e) x_{ij}^l \leq C_e, \quad \forall e \in E \quad (2.5)$$

For the following we focus on the rate control problem satisfying constraints (2.3) and (2.5) with the objective function being (2.4).

2.4 The Rate Control Algorithm

Note that even if every utility function $U_i(\cdot)$ is strictly concave, the objective function in (2.4) may not be strictly concave due to the presence of the linear terms $\sum_{k=1}^{|\mathcal{P}_i|} x_i^k + \sum_{j:i \neq j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} x_{ij}^l$. Thus, a direct application of standard convex optimization techniques might lead to multiple solutions, for which the output of an iterative method may oscillate. However, we can apply the “proximal method” described in [63] page 233 to ensure convergence. The idea behind the proximal method is to solve a series of problems, each of which has a strictly concave objective function.

The limit of the series approaches a single solution of the original problem. A detailed description of the proximal method is in [63]. To implement the proximal method, we now introduce auxiliary variables $\vec{y} = \{y_i^k, y_{ij}^l\}$ with the same size of \vec{x} . The intermediate optimization problem of the proximal method becomes:

$$\begin{aligned} \max_{\vec{x} \geq 0} & \sum_{i=1}^N U_i \left(\sum_{k=1}^{|\mathcal{P}_i|} x_i^k + \sum_{j:i \neq j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} x_{ij}^l \right) \\ & - \sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} \frac{\alpha_i}{2} (x_i^k - y_i^k)^2 - \sum_{(i,j):i \neq j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} \frac{\alpha_i}{2} (x_{ij}^l - y_{ij}^l)^2 \end{aligned} \quad (2.6)$$

subject to \vec{x} satisfying (2.3) and (2.5), where α_i is a positive constant.

In the following, we focus on the dual of the intermediate maximization problem. Since the Slater condition holds (see for reference [65]), there is no duality gap between the primal and the dual problems. Hence, we can use the dual approach to solve the problem.

Associate Lagrange multiplier λ_e with each link e , and μ_{ij}^l with the l -th PICC between sessions i and j . Also, let $\vec{\lambda}$ and $\vec{\mu}$ be two column vectors with elements λ_e and μ_{ij}^l , respectively. The Lagrange function of the above primal intermediate problem is:

$$\begin{aligned} L(\vec{x}, \vec{\lambda}, \vec{\mu}, \vec{y}) &= \sum_{i=1}^N U_i \left(\sum_{k=1}^{|\mathcal{P}_i|} x_i^k + \sum_{j:i \neq j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} x_{ij}^l \right) - \sum_{(i,j):i \neq j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} \frac{\alpha_i}{2} (x_{ij}^l - y_{ij}^l)^2 \\ & - \sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} \frac{\alpha_i}{2} (x_i^k - y_i^k)^2 - \sum_{(i,j):i < j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} \mu_{ij}^l x_{ij}^l + \sum_{(i,j):i < j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} \mu_{ij}^l x_{ji}^l \\ & + \sum_e \lambda_e C_e - \sum_e \lambda_e \left\{ \sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e) x_i^k + \sum_{(i,j):i \neq j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} H_{ij}^l(e) x_{ij}^l \right\} \end{aligned}$$

Since $\sum_{(i,j):i < j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} \mu_{ij}^l x_{ji}^l = \sum_{(i,j):j < i} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} \mu_{ji}^l x_{ij}^l$, by a simple change of variables the Lagrange function is *separable* and we can rewrite it as:

$$L(\vec{x}, \vec{\lambda}, \vec{\mu}, \vec{y}) = \sum_{i=1}^N B_i(\vec{x}, \vec{\lambda}, \vec{\mu}, \vec{y}) + \sum_e \lambda_e C_e.$$

Here,

$$\begin{aligned}
B_i(\vec{x}, \vec{\lambda}, \vec{\mu}, \vec{y}) = & U_i \left(\sum_{k=1}^{|\mathcal{P}_i|} x_i^k + \sum_{j:i \neq j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} x_{ij}^l \right) - \sum_{k=1}^{|\mathcal{P}_i|} \frac{\alpha_i}{2} (x_i^k - y_i^k)^2 \\
& - \sum_{j:i \neq j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} \frac{\alpha_i}{2} (x_{ij}^l - y_{ij}^l)^2 - \sum_{k=1}^{|\mathcal{P}_i|} \left(\sum_e H_i^k(e) \lambda_e \right) x_i^k \\
& - \sum_{j:i \neq j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} \left(\sum_e H_{ij}^l(e) \lambda_e \right) x_{ij}^l - \sum_{j:i < j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} \mu_{ij}^l x_{ij}^l \\
& + \sum_{j:i > j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} \mu_{ji}^l x_{ij}^l.
\end{aligned}$$

The objective function of the dual problem is

$$D(\vec{\lambda}, \vec{\mu}, \vec{y}) = \max_{\vec{x} \geq 0} L(\vec{x}, \vec{\lambda}, \vec{\mu}, \vec{y}),$$

and the dual problem is:

$$\min_{\vec{\lambda} \geq 0, \vec{\mu}} D(\vec{\lambda}, \vec{\mu}, \vec{y}).$$

The dual optimization problem can be solved using the gradient method.

Based on the above discussion we have the following distributed rate control algorithm (Algorithm A).

Algorithm A:

- Initialization phase: Find all paths between all sources and destinations. This can be done using any routing protocol that finds multiple paths in a distributed way as in [83, 84]. After this, sources send control messages to every link e to set the values of $H_i^k(e)$ and $H_{ij}^l(e)$. Each link sets its corresponding $\lambda_e(0)$ to zero, each destination d_i sets its corresponding $\mu_{ij}^l(0)$ to zero, and each source s_i chooses the values of $y_i^k(0)$, $y_{ij}^l(0)$, $x_i^k(0)$ and $x_{ij}^l(0)$ arbitrarily.
- Iteration phase: At the t -th iteration:
 1. Fix $\vec{\lambda}(t, 0) = \vec{\lambda}(t)$, $\vec{\mu}(t, 0) = \vec{\mu}(t)$, and $\vec{x}(t, 0) = \vec{x}(t)$.
 2. perform the following steps sequentially for $\kappa = 0, \dots, K - 1$.

- Update the dual variables at each link e by:

$$\lambda_e(t, \kappa + 1) = \left[\lambda_e(t, \kappa) + \beta_e \sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e) x_i^k(t, \kappa) \right. \quad (2.7)$$

$$\left. + \sum_{(i,j):i=j}^{|\mathcal{P}ICC_{ij}|} \sum_{l=1} H_{ij}^l(e) x_{ij}^l(t, \kappa) - C_e \right]^+. \quad (2.8)$$

Here, $[\cdot]^+$ is a projection on $[0, \infty)$ and β_e is a positive step size. Also, $(\sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e) x_i^k(t, \kappa) + \sum_{(i,j):i=j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} H_{ij}^l(e) x_{ij}^l(t, \kappa) - C_e)$ is the queue length change at link e during the time from the κ -th to the $(\kappa + 1)$ -th step.

- Set

$$\mu_{ij}^l(t, \kappa + 1) = \mu_{ij}^l(t, \kappa) + \beta_{ij}^l (x_{ij}^l(t, \kappa) - x_{ji}^l(t, \kappa)), \quad \forall i < j. \quad (2.9)$$

This can be implemented at each destination d_i , where β_{ij}^l is a positive step size.

- Let $\vec{x}(t, \kappa + 1) = \arg \max_{\vec{x} \geq 0} L(\vec{x}, \vec{\lambda}(t, \kappa + 1), \vec{\mu}(t, \kappa + 1), \vec{y}(t))$.

This can be computed in a distributed way at each source since the L function is separable. It is worth noting that computing $\vec{x}(t, \kappa + 1)$ needs the values of (i) $\sum_e H_{ij}^l(e) \lambda_e(t, \kappa + 1)$, $\forall i < j, l, m$, which can be computed along the paths, (ii) $\mu_{ij}^l(t, \kappa + 1)$, $\forall i < j, l$, and (iii) $\mu_{ji}^l(t, \kappa + 1)$, $\forall i > j, l$. All of this information can be sent back to the source using an acknowledgment message as will be explained in Section 2.5.1.

- Let $\vec{\lambda}(t + 1) = \vec{\lambda}(t, K)$ and $\vec{\mu}(t + 1) = \vec{\mu}(t, K)$. Set

$$\vec{y}(t + 1) = \vec{x}(t, K)$$

and

$$\vec{x}(t + 1) = \vec{x}(t, K).$$

For sufficiently large K and sufficiently large number of iterations, $\vec{x}(t)$ converges to the optimizing \vec{x}^* for the original problem with the objective function in (2.4) and the constraints (2.3) and (2.5).

Proposition 2.4.1 *As $K \rightarrow \infty$, with the step sizes (β_e, β_{ij}^l) satisfying the following: $(\mathcal{L} \cdot \max_e \beta_e + 2 \max_{\{i,j,l\}} \beta_{ij}^l) < 2 \min_i(\alpha_i)$, where*

$$\mathcal{L} = \left(\sum_e \sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e) + \sum_{(i,j):i=j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} (H_{ij}^l(e))^2 \right),$$

Algorithm A converges to the optimal solution of (2.4) subject to the constraints (2.3) and (2.5).

The proof is provided in Appendix A.4. For the case when K is bounded away from infinity, the convergence of Algorithm A is verified by simulations. Similar proofs to those in [79] can be used to rigorously prove the convergence of Algorithm A with fixed K and with noisy and delayed measurements. This makes Algorithm A suitable for practical implementation.

2.5 Implementation Details

In this section, we discuss several practical issues that may impact the implementation of our algorithm. In Section 2.5.1 we show how to collect the implicit costs needed for Algorithm A. The pairwise random coding scheme is introduced in Section 2.5.2, followed by a discussion of the transient behavior before Algorithm A converges in Section 2.5.3. A brief discussion of how to deal with non-concave objective functions for real-time traffic is in Section 2.5.4.

2.5.1 Collecting implicit costs

Each source s_i needs to collect $\sum_e \lambda_e H_{ij}^l(e)$, $\forall j \neq i, l \in \{1, \dots, |\mathcal{P}ICC_{ij}^l|\}$ in order to compute the update rate x_{ij}^l . To do so, special control messages $\mathcal{S}_{ij}^l(u, e)$ and $\mathcal{S}_{ij}^l(e, v)$ are used. $\mathcal{S}_{ij}^l(u, e)$ is the control message sent from node u to link e to collect

$\sum_e \lambda_e H_{ij}^l(e)$. Similarly, $\mathcal{S}_{ij}^l(e, v)$ is the control message sent from link e to node v to collect $\sum_e \lambda_e H_{ij}^l(e)$. More explicitly, collecting $\sum_e \lambda_e H_{ij}^l(e)$, $\forall j \neq i$ is done according to the following.

- Each source s_i sets $\mathcal{S}_{ij}^l(s_i, e) = 0$ to all of its outgoing links e that satisfy $H_{ij}^l(e) \neq 0$.
- Assuming $e = (u, v)$, then at link e , $\mathcal{S}_{ij}^l(e, v) = \mathcal{S}_{ij}^l(u, e) + \lambda_e H_{ij}^l(e)$.
- At every intermediate node v , let $\text{In}_{ij}^l(v)$ be the set of incoming links to node v such that $H_{ij}^l(e) \neq 0$, and $\text{Out}_{ij}^l(v)$ be the set of outgoing links from node v such that $H_{ij}^l(e) \neq 0$. Then node v arbitrarily chooses one $e_v \in \text{Out}_{ij}^l(v)$ and sets $\mathcal{S}_{ij}^l(v, e_v) = \sum_{e \in \text{In}_{ij}^l(v)} \mathcal{S}_{ij}^l(e, v)$ and $\mathcal{S}_{ij}^l(v, e) = 0$ for all links $e \in \text{Out}_{ij}^l(v) \setminus e_v$.

The third step avoids overcounting the implicit costs. In the end, $\sum_{e \in \text{In}_{ij}^l(d_i)} \mathcal{S}_{ij}^l(e, d_i) + \sum_{e \in \text{In}_{ij}^l(d_j)} \mathcal{S}_{ij}^l(e, d_j) = \sum_e \lambda_e H_{ij}^l(e)$. The first term of the left hand side can be obtained at d_i while the second can be obtained at d_j . Both of them can be sent back using the acknowledge messages and s_i can obtain $\sum_e \lambda_e H_{ij}^l(e)$.

2.5.2 The Coding Scheme

The optimization problem and the solution described thus far allocate rates at each link so that the utility function can be optimized subject to \vec{x} being in the PINC region. The next question is what is the network coding scheme that can achieve the optimal rate assignment? In this section, we propose the use of a scheme we call the pairwise random coding scheme. Suppose rate x_{ij}^l is sustained along the l -th PICC between sessions i and j . From a packet-by-packet perspective it means that every $\frac{1}{x_{ij}^l}$ unit time one packet will be sent from s_i to d_i and another packet will be sent from s_j to d_j . Therefore, we can focus on coding over those two packets (every $\frac{1}{x_{ij}^l}$ unit time) along the corresponding PICC. Let the integral graph G'' represent the underlying PICC. Without loss of generality we assume that G'' is for the session

pair (s_1, d_1) and (s_2, d_2) . We further assume that the packets for the unicast sessions (s_1, d_1) , (s_2, d_2) are X , Y , respectively.

One choice of the coding scheme that is widely used is the random linear coding scheme, as in [9]. Unfortunately directly using random network coding for *pairwise intersession network coding* without modification is infeasible. Take Fig. 2.1(a) for example, which is a typical choice of random network coding over $\text{GF}(17)$ where the vector (θ_1, θ_2) at edge e in the figure represents that the packet at link e contains $\theta_1 X + \theta_2 Y$. In this case d_1 will not be able to decode both X and Y , as random mixing is performed at v_3 and d_1 will receive $9X + 2Y$. If both d_1, d_2 have min-cut max-flow values being ≥ 2 , random network coding is sufficient for PINC, because both d_1, d_2 can decode both symbols. The infeasibility of random network coding is caused by the min-cut max-flow value from s_1 and s_2 to either d_1 or d_2 being 1. If the min-cut max-flow value from s_1 and s_2 to either d_1 or d_2 is 1, either the paths in the set $\mathcal{Q}_1 = \{P_{s_1, d_1}, P_{s_2, d_1}, Q_{s_1, d_1}\}$ or the paths in the set $\mathcal{Q}_2 = \{P_{s_2, d_2}, Q_{s_1, d_2}, Q_{s_2, d_2}\}$ share the same edge in G'' based on the path selection in (1.1). For example, in Fig. 2.1(c) all paths in the set \mathcal{Q}_1 share edge (v_3, v_4) . Motivated by this observation, the pairwise random coding scheme performs pure routing and random network coding on most part of the network and performs decoding on only two nodes. The pairwise random network coding is described as follows.

Find the furthest edge $e_1 = (u_1, v_1)$ from d_1 such that (i) $1_{e_1 \in P_{s_1, d_1}} + 1_{e_1 \in P_{s_2, d_1}} + 1_{e_1 \in Q_{s_1, d_1}} = 3$. (ii) For all paths in \mathcal{Q}_1 the segments from v_1 to d_1 are edge disjoint from the path P_{s_2, d_2} . Also find the furthest edge $e_2 = (u_2, v_2)$ from d_2 such that (a) $1_{e_2 \in P_{s_2, d_2}} + 1_{e_2 \in Q_{s_1, d_2}} + 1_{e_2 \in Q_{s_2, d_2}} = 3$. (b) For all paths in \mathcal{Q}_2 the segments from v_2 to d_2 are edge disjoint from the path P_{s_1, d_1} . After that perform random linear network coding through all the edges of G'' except edges e_1 and e_2 . Decode X on e_1 and forward it to d_1 through the segment of path Q_{s_1, d_1} that goes from u_1 to d_1 , decode Y on e_2 and forward it to d_2 through the segment of path P_{s_2, d_2} that goes from u_2 to d_2 . For example, if we use pairwise random coding in Fig. 2.1(b), (v_3, v_4) will be the first edge that satisfies the conditions for e_1 in the pairwise coding scheme as is clear

from Fig. 2.1(c). Therefore, v_3 will decode X instead of random mixing and forward it to d_2 .

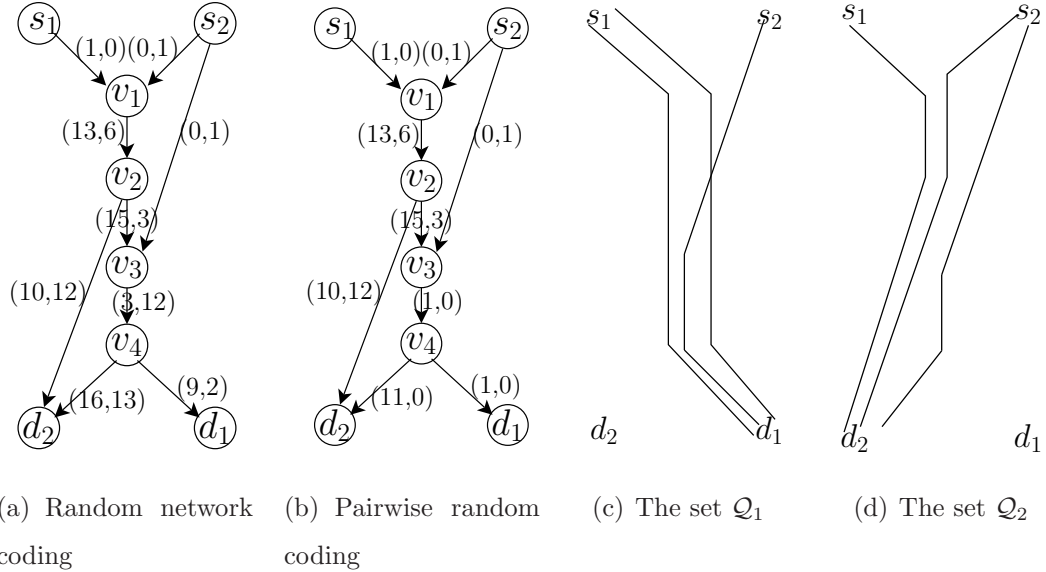


Fig. 2.1. Applying both the random network coding and the pairwise random network coding to the grail structure.

Proposition 2.5.1 *Given that pairwise network coding is feasible on the directed acyclic PICC G'' as in Proposition 1.3.1, the probability that the pairwise random coding scheme is able to transmit X and Y successfully for sessions (s_1, d_1) and (s_2, d_2) , is lower bounded by $Pr(\text{success}) \geq (1 - \frac{2}{q})^{|E''|}$. Here, q is the field size and $|E''|$ is the number of edges in G'' .*

Proof $Pr(\text{success})$ is lower bounded by the probability that both u_1 and u_2 recover both X and Y successfully. Because (i) G'' is directed acyclic, (ii) the min-cut from s_1 and s_2 to u_1 is ≥ 2 , (iii) the min-cut from s_1 and s_2 to u_2 is ≥ 2 , we have three cases. Case 1: There is no path from v_1 to u_2 nor there is a path from v_2 to u_1 . The problem is the same as multicasting both X and Y to both u_1 and u_2 when all the coding coefficients are random and the inequality holds. Case 2: There exists a path from v_1 to u_2 . Here we construct another graph F'' from G'' by removing all outgoing

edges from v_1 and replacing them by new edges from s_1 to the same vertices that the removed edges were going to. We send X through these edges and perform pairwise random coding through the rest of the edges in F'' . The probability that both u_1 and u_2 recover both X and Y on G'' is the same of that on F'' , which satisfies the inequality. Case 3: There exists a path from v_2 to u_1 . This case is symmetric to case 3 and so we remove the outgoing edges of v_2 and replace them by new edges from s_2 to show that the inequality holds. ■

The pairwise random coding scheme can be implemented in a distributed way. Two trace messages can be sent back by the destinations d_1 and d_2 during the initialization phase to identify edges e_1 and e_2 to perform decoding. Furthermore, by Proposition 2.5.1, we can see that the success probability of pairwise random coding scheme approaches one when the size of the finite field is sufficiently large. In practice [85] moderately-sized $q = 2^{16}$ or $q = 2^8$ is sufficient without incurring too much overhead (generally 3-6%).

2.5.3 Coding Scheme when $x_{ij}^l \neq x_{ji}^l$

The above pairwise random coding scheme assumes that two sessions (s_i, d_i) and (s_j, d_j) share the same pairwise coding rates $x_{ij}^l = x_{ji}^l$, which is achieved after the convergence of the Algorithm A (as proven in Proposition 2.4.1). However, during the transient time before convergence, we might have unequal cross-coding rates assigned by each individual session respectively. Furthermore, it is difficult to know when Algorithm A converges. To overcome these difficulties the coding scheme can be modified in the transient state when $x_{ij}^l \neq x_{ji}^l$. The basic idea is if $x_{ij}^l(t) > x_{ji}^l(t)$, we perform pairwise network coding at the smaller rate $x_{ji}^l(t)$ and send uncoded packets at rate $(x_{ij}^l(t) - x_{ji}^l(t))$. Therefore, d_j can receive coded packets at rate x_{ji}^l while d_i can receive coded packets at rate x_{ji}^l and uncoded packets at rate $x_{ij}^l - x_{ji}^l$ (the total rate is still x_{ij}^l). For example, assume that $x_{ij}^l(t) = 3$ and $x_{ji}^l(t) = 2$, then packets with sequence numbers 1,2,4,5,7,8,10,11 of s_i will be coded with packets with

sequence numbers 1,2,3,4,5,6,7,8 of s_j . Packets with sequence numbers 3,6,9,12 of s_i will be forwarded as shown in Fig. 2.2(b). Take the well studied butterfly structure as an example, the assignment of the code on the edges of this PICC in the transient state is represented in Fig. 2.2(a). In the figure we assume that the rate is 3 for s_1 and 2 for s_2 . Therefore, in one time slot s_1 will send X , Y , and X_3 and s_2 will send Y_1 and Y_2 . Packets X and Y will be coded with packets Y_1 and Y_2 , respectively, and packet X_3 will be forwarded without coding. In this way d_1 is able to receive at rate 3 and d_2 is able to receive at rate 2.

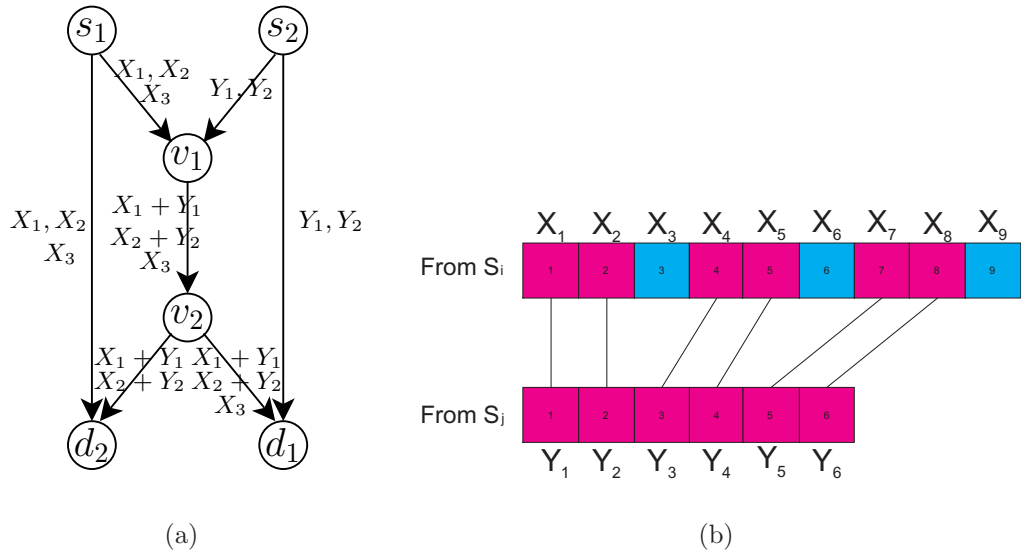


Fig. 2.2. (a) The assignment of the codes on the butterfly as an example of the l -th PICC between sessions i and j with unequal rates $x_{ij}^l = 3$ and $x_{ji}^l = 2$ during the transient state. (b) The upper sequence represents the sequence numbers of packets sent by s_i through the l -th PICC between sessions i and j , and the lower sequence represents the sequence numbers of packets sent by s_j through the same PICC. If a packet in the upper sequence is to be coded with another one in the lower sequence, there is a link between them. Packets without links are those for which no coding is performed.

2.5.4 Dealing with non-concave objective functions for real-time traffic

Our assumption here is that the utility function is concave which does not hold for transmission scenarios like real-time traffic. Real-time traffic can be modelled by sigmoidal functions. In this case Algorithms A can be modified as in [86] to achieve the optimal rate control when the number of sessions is sufficiently large. This is possible because of the path based formulation we use.

2.6 Complexity Reduction

The complexity of Algorithm A depends on the number of PICCs in the network. This is because for the l -th PICC between sessions i and j we assign two primal variables x_{ij}^l and x_{ji}^l . Also every link e has to maintain variables of the form $H_{ij}^l(e)$ and each destination d_i has to maintain variables of the form μ_{ij}^l . Furthermore, to compute x_{ij}^l in every update, source s_i has to collect $\sum_e \lambda_e H_{ij}^l(e)$ as explained in Section 2.5.1. Using the approach in Section 2.3, the number of PICCs between session i and j is $(|\mathcal{P}_{ii}|^2 \cdot |\mathcal{P}_{jj}|^2 \cdot |\mathcal{P}_{ij}| \cdot |\mathcal{P}_{ji}|)$, and so the total number of PICCs in the network is $\sum_{(i,j):i=j} |\mathcal{P}_{ii}|^2 \cdot |\mathcal{P}_{jj}|^2 \cdot |\mathcal{P}_{ij}| \cdot |\mathcal{P}_{ji}|$. Here, $|\mathcal{P}_{ij}|$ represents the number of paths between s_i and d_j . From the above discussion, reducing the number of PICCs in the network plays a major role in the practical implementation of the proposed algorithm. In this section we provide two approaches to reduce the complexity of Algorithm A. The first approach reduces the number of PICCs without sacrificing performance, while the second chooses paths and PICCs adaptively and may sacrifice performance.

2.6.1 Excluding redundant PICCs in the initialization step

By construction, the l -th PICC between sessions i and j satisfies condition 2 of Proposition 1.3.1 and supports the coded traffic rate x_{ij}^l . In this section we provide rules to see whether condition 1 of Proposition 1.3.1 is also satisfied on that PICC at

rate x_{ij}^l . If so, we can remove x_{ij}^l and $H_{ij}^l(e)$, $\forall e$ from the optimization problem (2.4) and still achieve the same optimal solution. This is because the 2EDPs in that PICC can be used to send uncoded traffic which has already been characterized by x_i^k , the uncoded data rate in (2.2). The rules also help detecting whether a given PICC contains redundant links such that there exists another PICC whose links are a proper subset of the links used by the given PICC. Since for these PICCs, we can use a strictly smaller part of the PICC while still providing the same throughput improvement, we term those PICCs redundant PICCs. The following rules remove redundant PICC and result in a tremendous reduction in the complexity of Algorithm A. In the following we assume the same simplification as in section 2.5.2 by considering the integral graph G'' for session pair $(s_1, d_1), (s_2, d_2)$. To identify the PICCs that should be removed from the optimization problem, we have the following rules for directed acyclic PICCs:

- Rule 1: For G'' , if neither $P_{s_1, d_1} = Q_{s_1, d_1}$ nor $P_{s_2, d_2} = Q_{s_2, d_2}$, then G'' is a redundant PICC.
- Rule 2: For a given path P , let $E(P)$ be the set of edges in path P . If the two sets of edges $E(P_{s_1, d_1}) \cup E(Q_{s_1, d_1})$ and $E(P_{s_2, d_2}) \cup E(Q_{s_2, d_2})$ are disjoint, then G'' is a redundant PICC.
- Rule 3: If $H_{ij}^l(e) = 1$ for some link e , then the l -th PICC between sessions i and j is redundant.

If a PICC is classified as redundant by a rule, we say that the PICC is declared redundant by that rule. Otherwise, we say that the PICC passes that rule.

Proposition 2.6.1 *Rules 1-3 identify redundant PICCs. All redundant PICCs can be removed without sacrificing the achievable rate.*

Based on Proposition 2.6.1, we have the following reduced-complexity algorithm which achieves the same capacity region as by Algorithm A.

(Algorithm B):

1. *Path Finding*: Every source finds a set of paths to every destination, and announces the sizes of these sets to the links in these paths and to other sources. Every PICC that passes Rule 1 will have a unique ID number which can be computed locally at every link in the network.
2. Source s_i sends trace message through all paths $P_{s_i d_j}$. This message includes i, j , and the path number.
3. Link e sets up $H_{ij}^l(e)$ for all PICCs that pass Rule 1.
4. The following steps are executed at every link e in parallel
 - If P_{s_i, d_i}^l or Q_{s_i, d_i}^l share link e with P_{s_j, d_j}^l or Q_{s_j, d_j}^l a notification message of type \mathcal{X} is sent back to s_i, s_j , with the ID number of the l -th PICC between sessions i and j . This means that the PICC passes Rule 2. Here, P_{s_i, d_i}^l represents the path from s_i to d_i in the set \mathcal{P} in Proposition 1.3.1 for the l -th PICC between sessions i and j . $Q_{s_i, d_i}^l, Q_{s_j, d_j}^l$, and P_{s_j, d_j}^l are defined in the same way.
 - If $H_{ij}^l(e) = 1$, link e sends a notification message of type \mathcal{Y} to s_i, s_j , with the ID number of the l -th PICC between sessions i and j . This means the l -th PICC between i and j is redundant by Rule 3.
5. Sources s_i and s_j delete all PICC for which a notification message of type \mathcal{Y} is received or no notification message of type \mathcal{X} is received.
6. Run Algorithm A on the non deleted PICCs.

Steps (1)-(5) are initialization steps and executed only once. Also, they can be executed in a distributed manner.

2.6.2 Adaptive Algorithm

The reductions in Section 2.6.1 reduce the number of PICCs in the initialization step without sacrificing the performance by eliminating redundant PICCs. To further

reduce the complexity, we propose another type of reduction. As observed by our simulations the rates on some PICCs converge to zero very quickly (generally after only a few iterations), which means that network coding over those PICCs provides no positive gain when compared to the optimal rate-control solution. This may be due to that non-coded solution is sufficient for those PICCs because they satisfy both conditions of Proposition 1.3.1. It may also be because the links used by the PICCs can be used by more significant PICCs. We termed those PICCs as *insignificant* PICCs. The adaptive scheme we propose in this section works initially on a small number of PICCs. While the algorithm is run on these PICCs, insignificant PICCs among them will be deleted and new PICCs formed by the newly found paths will be added adaptively. This approach reduces the variable space of the optimization problem. Fig. 2.3 contains a detailed description of the above scheme with an adaptive path search mechanism.

In the flow chart in Fig. 2.3, every source maintains a collection of paths \mathcal{P}_{found} and every source pair maintains collections of PICCs $\mathcal{P}ICC_{found}$ and $\mathcal{P}ICC_{active}$. Every time new paths are found, the source puts them in \mathcal{P}_{found} which is executed in parallel to the steps in Fig. 2.3. When $\mathcal{P}ICC_{found}$ becomes empty, all possible PICCs that can be formed by the paths in the \mathcal{P}_{found} and have not been used by the algorithm yet, are put in $\mathcal{P}ICC_{found}$. Each pair of sessions i and j is assigned a value ϕ_{ij} . The constant ϕ_{ij} represents the maximum number of PICCs (for sessions i and j) that can be included in the maximization problem simultaneously. If the rate of a given PICC converges to zero, it is identified as an insignificant one. Convergence to zero is detected when the rate of the PICC goes below a threshold value. Every time insignificant PICCs (for sessions i and j) are removed from the optimization problem, the complexity of the algorithm reduces and we can afford to include one new PICC when solving the optimization problem. Therefore, s_i moves some of the PICCs (for sessions i and j) from $\mathcal{P}ICC_{found}$ to $\mathcal{P}ICC_{active}$. This is done in a way such that the total number of PICCs for i and j in $\mathcal{P}ICC_{active}$ does not exceed ϕ_{ij} .

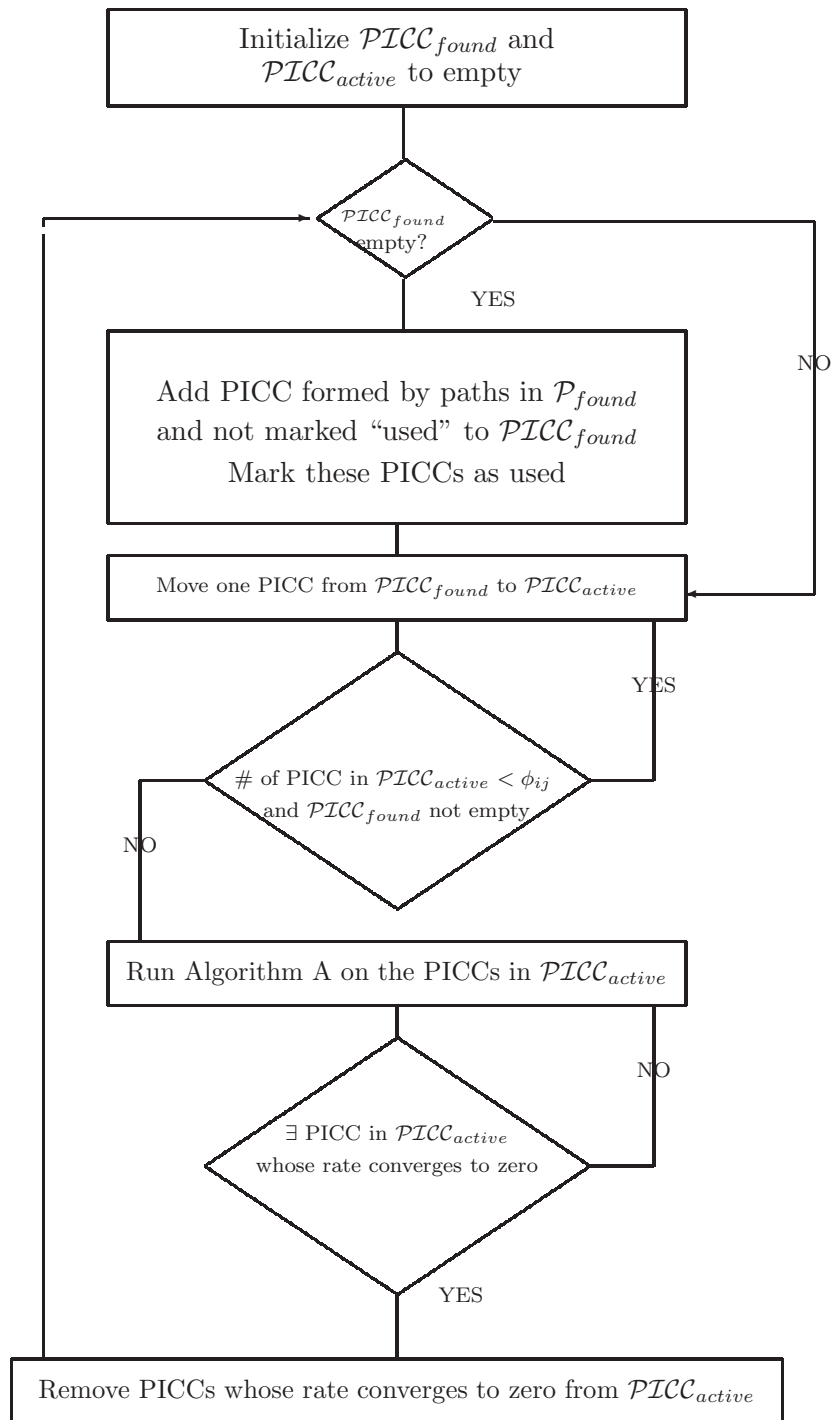


Fig. 2.3. Flow chart for the adaptive algorithm.

For distributed implementation one source of each pair is assigned as a controller to ensure the consistency of $\mathcal{P}ICC_{found}$ and $\mathcal{P}ICC_{active}$ for the two sources.

2.7 Simulation Results

The objectives of the simulations are to verify the convergence of Algorithms A and B, and to show the benefits of the proposed intersession network coding solution in terms of throughput, fairness, and its complexity advantage over existing intersession coding rate-control solutions.

2.7.1 Convergence

To study the convergence of Algorithms A and B, we run simulations on the so called grail topology in Fig. 1.4(a) with the utility function of each source s_i being $\log_2(R_i)$. As is evident from the grail topology (Fig. 1.4(a)), there are three paths connecting (s_2, d_2) , two paths connecting (s_1, d_2) , two paths connecting (s_2, d_1) , and one path connecting (s_1, d_1) . Therefore, there are six different path collections \mathcal{P} and six different path collections \mathcal{Q} . Totally, there are 36 possible PICCs. We assign the initial rates of each PICC randomly, and vary β_e , β_{ij}^l and the number of proximal iterations K to test the speed of convergence of Algorithm A.

The optimal solution for the grail topology is to assign unit rate to the optimal PICC that uses the paths in (1.1), and zero rates to all of the other PICCs. These are the paths that satisfy condition 2 in Proposition 1.3.1 as explained in section 2.2. In Fig. 2.4 we show the rates for the optimal PICC with different step sizes. Every outer iteration contains K proximal iterations. Our algorithm converges even with a very small number of proximal iterations. As expected, increasing the step size up to a specific value will make the algorithm converge faster. A bigger topology in Fig. 3.8 with 36 nodes and unit capacity links is used in our simulations. This topology has four unicast sessions. The convergence results for one of the optimal PICCs and one

of the insignificant PICCs in the topology in Fig. 3.8 are shown in Fig. 2.6. In Fig. 2.6, the rate of the insignificant PICC converges quickly to zero.

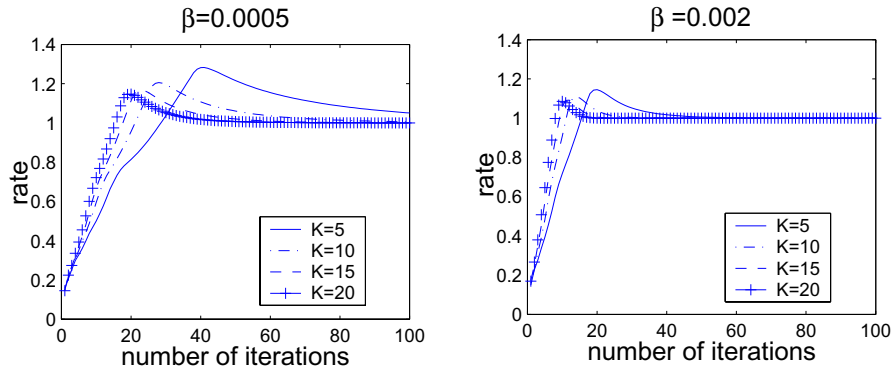


Fig. 2.4. Convergence results for s_1 in the grail topology with different step sizes and K , the number of proximal iterations. Here, the rate corresponds to the optimal PICC.

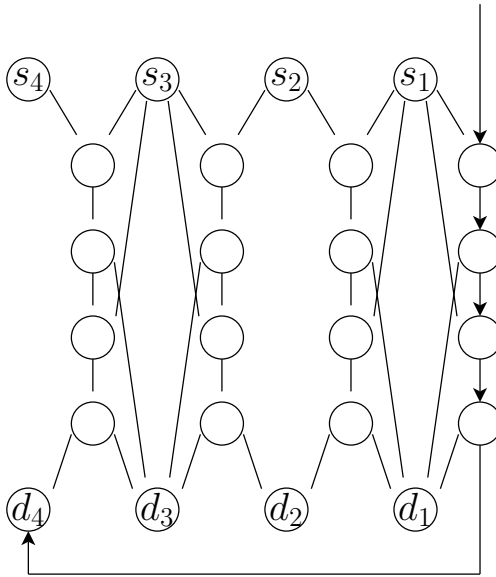


Fig. 2.5. Topology contains four source-sink pairs.

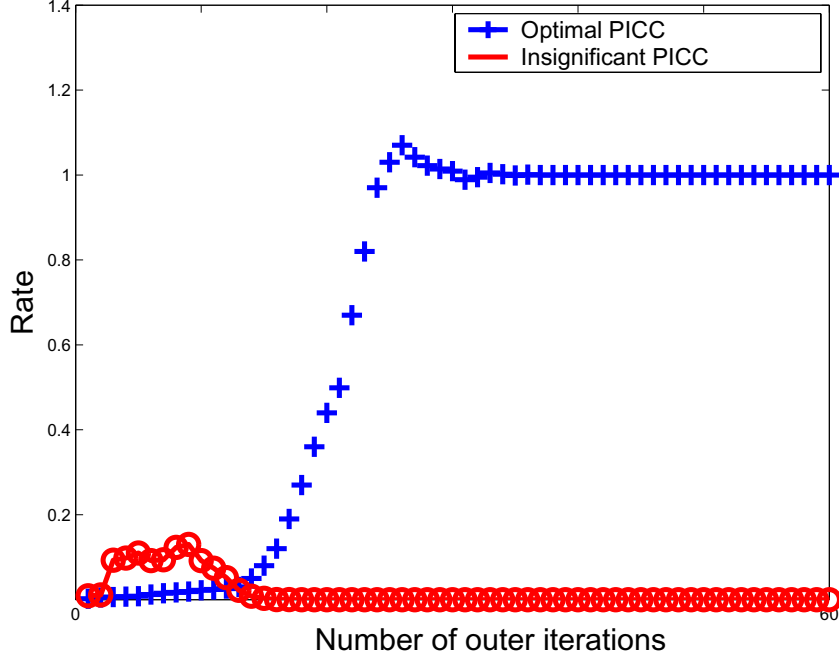


Fig. 2.6. Convergence results for the topology in Fig. 3.8 with $\beta = 0.01$, and the number of proximal iterations $K = 5$. We plot the convergence rate vs. iterations for the optimal PICC and one insignificant PICC.

2.7.2 Gain and Fairness

We compare Algorithm A with existing algorithms and quantify the benefits of intersession network coding over non-coded solutions. The simulation is conducted on a graph depicted in Fig. 3.8. For this topology, the butterfly-based work [82] and its distributed implementation in [41] and [40] cannot realize any throughput benefits of network coding and the performance of these algorithms is the same as that of non-coded solutions since there is no butterfly substructure in Fig. 3.8. It is worth noting that the distributed implementations of the butterfly-based region in [40, 41] focus on stabilizing the given traffic load instead of maximizing the utility function. We define the utility gain of pairwise intersession network coding PINC, \mathcal{UG} as

$$\mathcal{UG} = \frac{Utility(\text{PINC}) - Utility(\text{non-coded})}{Utility(\text{non-coded})}.$$

We denote the total throughput of the network when the optimal utility is achieved under the PINC and the non-coded solutions by $\sum_i R_i(\text{PINC})$ and $\sum_i R_i(\text{non-coded})$, respectively. The throughput gain, \mathcal{TG} is defined as

$$\mathcal{TG} = \frac{\sum_i R_i(\text{PINC}) - \sum_i R_i(\text{non-coded})}{\sum_i R_i(\text{non-coded})}.$$

We evaluate the gains of Algorithm A using different utility functions presented in [68] and [69]. The first type of utility function is $\log_2(\delta + R_i)$, where δ is a constant in the range $[0, 1]$. The second type of utility function is of the form $\frac{R_i^{1-\sigma}}{1-\sigma}$, where σ is a constant in the range $(0, 1)$. The results are shown in Figs. 2.7 and 2.8. Algorithm A provides strict performance gains over both non-coded and butterfly-based capacity region on this topology. Moreover, the largest throughput gain happens when fairness is the design criteria for the network, i.e, when δ is small and when σ is large. This is the same conclusion drawn from the capacity regions in Figs. 1.3(b) and 1.4(b).

We also run simulations on the grid topology in Fig. 2.9. In this topology there are four paths between s_1 and d_1 , four paths between s_2 and d_2 , and only one path between s_3 and d_3 . Also, the path between s_3 and d_3 overlaps with all the other eight paths. Therefore, without network coding the rates of sessions 1 and 2 are about 3.7 times the rate of session 3 when δ is small 0.1 (fairness is of high priority) as shown in Table B.1. Using network coding the rate ratio is reduced to about 2.2 times with a small decrease in rates R_1 and R_2 and a considerable increase in R_3 from 0.29 to 0.47. When δ is relatively large 0.6, the rate ratio without network coding is about 21, because less emphasis is put on the smaller rate (rate of session 3 in this case.) Surprisingly with network coding the rate ratio is reduced to 4. This decrease in the ratio is due to that network coding resolves the bottlenecks. The fairness is thus improved as network coding removes the bottleneck for the smallest-rate session 3.

2.7.3 Complexity

In terms of computational complexity of Algorithm A and the existing butterfly-based method [82], the path based Algorithm A solves a maximization problem of

ing interference model for the former is absorbed as scheduling constraints Θ for the latter problem, which will be clear in the later section.

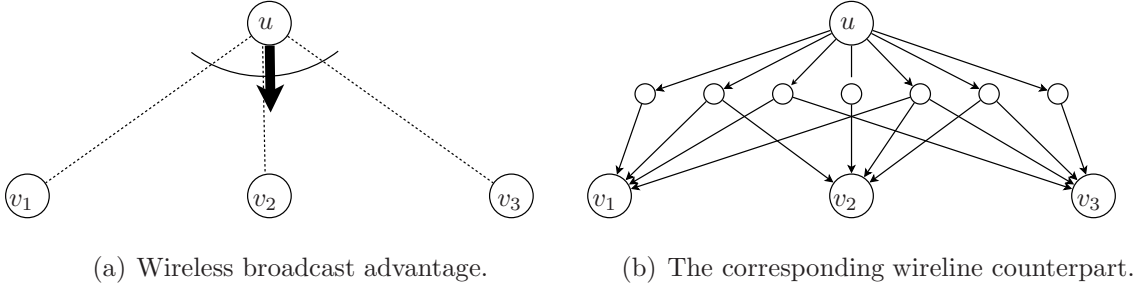


Fig. 3.1. Modelling the wireless multicast advantage.

3.3.2 Pairwise Intersession Network Coding for Wireless Networks

Based on a necessary and sufficient condition, the six-path-based PICC captures *all pairwise coding opportunities* once the aforementioned wireless to wireline conversion is properly exploited, which include the widely studied butterfly structure [40, 41, 44, 82, 87] and the one-hop coding opportunities [26, 27, 76, 77, 88] as special cases when coding is permitted only between two sessions.

Fig. 3.2(a) is a classic example of the one-hop intersession coding opportunity for wireless networks. Node A would like to send symbol X to node C while C intends to send Y to A . Fig. 3.2(a) depicts how to send two symbols in three time slots. The 1 and 2 in the small boxes indicate that A sends X to B in the first time slot while C sends Y to B in the second time slot. In the third time slot, B broadcasts coded symbol $X + Y$ to A and C using the WMA. If we follow the wireless to wireline conversion, Fig. 3.2(a) is transformed to Fig. 3.2(b). By noticing the existence of the \mathcal{P} and \mathcal{Q} paths with controlled edge-overlap as in Figs. 3.2(c) and 3.2(d), this one-hop coding opportunity for coding across two sessions is captured by Proposition 1.3.1 and corresponds to an instance of PICC. Note that Fig. 3.2(b) also indicates that under

a *node exclusive* model at least three time slots are necessary as the three involved auxiliary nodes cannot be scheduled simultaneously.

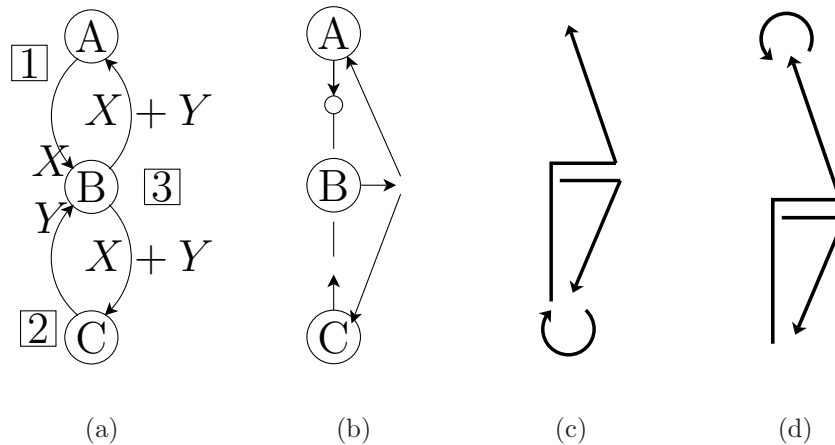


Fig. 3.2. A simple wireless one-hop coding opportunity with two sessions: Session 1: $A \xrightarrow{X} C$; Session 2: $C \xrightarrow{Y} A$. (a) The slot-by-slot wireless transmission. (b) Wireless to wireline conversion. (c) Paths P_{s_1,d_1} , P_{s_2,d_2} , and P_{s_2,d_1} . (d) Paths Q_{s_2,d_2} , Q_{s_1,d_1} , and Q_{s_1,d_2} .

Similarly Fig. 3.3(a) describes the classic wireless cross-flows in which symbols X and Y can be sent from A to E and from B to D in three time slots. D and E use the overheard packets X and Y for decoding. Figs. 3.3(b) to 3.3(d) depict the corresponding wireless to wireline conversion and show that the wireless cross flows can again be captured as a special instance PICC (which is actually a butterfly in the corresponding wireline network). Since the path-based characterization of PINC does not require that encoding and decoding happen at nodes that are 1-hop apart from each other, our formulation naturally takes into account coding opportunities over subgraphs of different “sizes,” e.g. 2-hop butterflies used in [44]. Capturing *all* pairwise coding opportunities, the PICCs also prompt new wireless intersession coding opportunities different from Figs. 3.2 and 3.3. For example, in Fig. 3.4(a), a new type of wireless cross flows is identified, for which A sends symbol X to F while C sends symbol Y to D . This example is not captured by the traditional one-

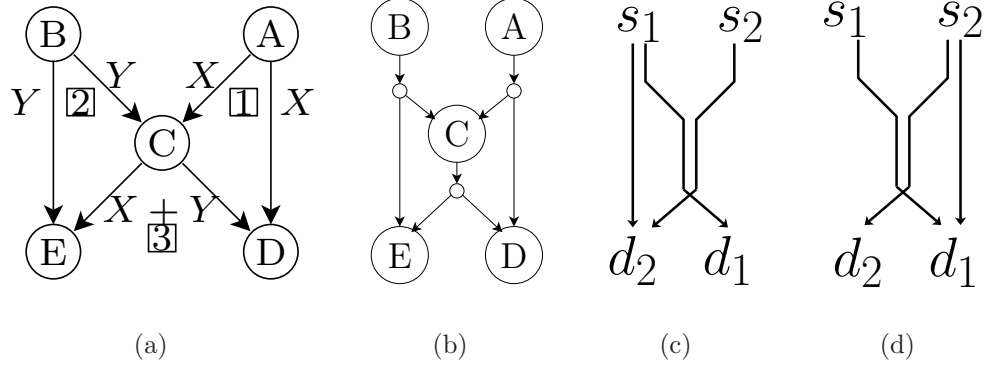


Fig. 3.3. The wireless cross flows with two sessions: Session 1: $A \xrightarrow{X} E$; Session 2: $B \xrightarrow{Y} D$. (a) The slot-by-slot wireless transmission. (b) Wireless to wireline conversion. (c) Paths P_{s_1,d_1} , P_{s_2,d_2} , and P_{s_2,d_1} . (d) Paths Q_{s_2,d_2} , Q_{s_1,d_1} , and Q_{s_1,d_2} .

hop coding opportunity as node D does not overhear the original symbol X sent by A but overhears the reconstructed symbol X decoded and sent by E . Figs. 3.4(c) and 3.4(d) illustrate the corresponding \mathcal{P} and \mathcal{Q} paths, which verify that this new type of wireless cross flows is a special instance of PICC (that is different than the classic butterfly structure). By including the existing coding opportunities as special cases and capturing additional ones, our PICC-based solution will enhance further the achievable capacity region of intersession network coding.

3.4 Optimal Joint Scheduling/Rate-Control With PINC

Following Section 3.3.1, we model a wireless network by its wireline counterpart denoted by $G = (V, E)$ where V is the set of network nodes plus auxiliary nodes and E is the edge set. Consider slotted transmission, a scheduling policy Θ is a collection of active edges and the associated power levels. Under a given interference model, we use r_e^Θ to denote the rate that can be supported on edge e under the scheduling policy Θ , and we often use \mathbf{r}^Θ for the collective rate vector. Let Θ denote the collection of all policies and let $\mathcal{R} \triangleq \{\mathbf{r}^\Theta : \forall \Theta \in \Theta\}$ denote the corresponding rates. Any rate vector $\mathbf{r} \in \text{Co}(\mathcal{R})$, the convex hull of \mathcal{R} , can be achieved via time sharing. Without

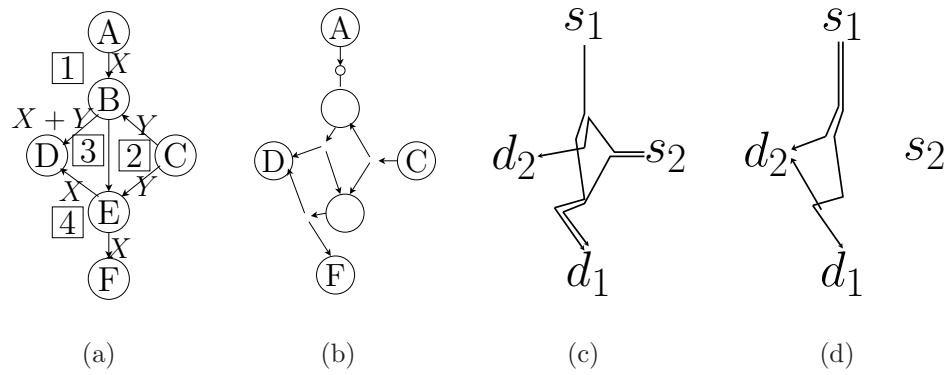


Fig. 3.4. A new type of wireless cross flows with two sessions: Session 1: $A \xrightarrow{X} F$; Session 2: $C \xrightarrow{Y} D$. (a) The slot-by-slot wireless transmission. (b) Wireless to wireline conversion. (c) Paths P_{s_1,d_1} , P_{s_2,d_2} , and P_{s_2,d_1} . (d) Paths Q_{s_2,d_2} , Q_{s_1,d_1} , and Q_{s_1,d_2} .

loss of generality, we assume the rate region is bounded. There are N coexisting unicast sessions using the network to send data from source s_i to destination d_i where $i = 1, \dots, N$. The utility function $U_i(x)$ for each session is strictly concave and monotonically increasing, where x is the end-to-end data rate for the session.

The utility optimization for multiple unicast sessions using PINC can be cast as follows.

$$\max_{0 \leq \mathbf{x} \leq M_X, \mathbf{r} \in \text{Co}(\mathcal{R})} \sum_{i=1}^N U_i \left(\sum_{k=1}^{|\mathcal{P}_i|} x_i^k + \sum_{\substack{j:j=i \\ l=1}}^{|\mathcal{P}ICC_{ij}|} x_{ij}^l \right) \quad (3.1)$$

$$\text{subject to} \quad \sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e) x_i^k + \sum_{\substack{(i,j):i=j \\ l=1}}^{|\mathcal{P}ICC_{ij}|} \frac{H_{ij}^l(e) x_{ij}^l}{2} \leq r_e, \forall e \in E \quad (3.2)$$

$$x_{ij}^l = x_{ji}^l, \quad \forall (i, j) : i < j, \forall l \quad (3.3)$$

where \mathcal{P}_i is the collection of paths from s_i to d_i along which packets will be routed without any coding operations and x_i^k is the rate assigned for the k -th path. $\mathcal{P}ICC_{ij}$ is the collection of PICCs between sessions i and j on which intersession network coding will be performed and x_{ij}^l is the packet rate of source s_i that will be network coded using the l -th PICC of $\mathcal{P}ICC_{ij}$. Without loss of generality, we further assume the indices of $\mathcal{P}ICC_{ij}$ and of $\mathcal{P}ICC_{ji}$ are consistent. Namely, for all l , the l -th PICC of $\mathcal{P}ICC_{ij}$ is also the l -th PICC of $\mathcal{P}ICC_{ji}$. Since in PINC, packets from s_i and s_j are coded bijectively with each other, the system requires the equal-rate constraint (3.3). Without loss of generality, we also assume that the rate vector \mathbf{x} is bounded by a finitely large constant M_X .

In (3.2), $H_i^k(e)$ is the indicator function whether the k -th path in \mathcal{P}_i uses edge e . $H_{ij}^l(e)$ is the indicator function whether the l -th PICC in $\mathcal{P}ICC_{ij}$ uses edge e . PINC ensures that the two packet flows (with rates x_{ij}^l and x_{ji}^l respectively) jointly use only the max rate $\max(H_{ij}^l(e)x_{ij}^l, H_{ji}^l(e)x_{ji}^l)$ instead of the sum rate $H_{ij}^l(e)x_{ij}^l + H_{ji}^l(e)x_{ji}^l$. By the fact that the indicator function is symmetric by definition, i.e. $H_{ij}^l(e) = H_{ji}^l(e)$, and by the equal-rate constraint in (3.3), the rate consumption becomes $\max(H_{ij}^l(e)x_{ij}^l, H_{ji}^l(e)x_{ji}^l) = \frac{H_{ij}^l(e)x_{ij}^l + H_{ji}^l(e)x_{ji}^l}{2}$. For each edge e , summing over

rates consumed by multi-path routing and by multi-PICC network coding leads to the capacity constraint (3.2). The non-negative rate vector \mathbf{x} , including all x_i^k and x_{ij}^l , is the subject of rate control and the edge rate vector $\mathbf{r} \in \text{Co}(\mathcal{R})$ is the subject of optimal scheduling and time-sharing. One can use any path search algorithm to initialize the paths collection \mathcal{P}_i . One advantage of considering PICC is that unlike the existing butterfly-search approach [40, 41], the characterization of PICC is *path-based* rather than structure-based. One can thus use any path search algorithm to identify possible constituent paths and any six paths $P_{s_1, d_1}, P_{s_2, d_2}, \dots, Q_{s_1, d_2}$ can serve as a PICC. It is worth pointing out that there is no need to strictly enforce Condition 2 of Proposition 1.3.1 during implementation. More explicitly, since each edge e knows whether itself participates in a given path during the path-search phase, e also knows its edge-overlap in the given six paths. Consider an edge e that is a *bottleneck*, i.e.

$$\begin{aligned} &1_{\{e \in P_{s_1, d_1}\}} + 1_{\{e \in P_{s_2, d_2}\}} + 1_{\{e \in P_{s_2, d_1}\}} = 3 \\ \text{or } &1_{\{e \in Q_{s_1, d_1}\}} + 1_{\{e \in Q_{s_2, d_2}\}} + 1_{\{e \in Q_{s_1, d_2}\}} = 3. \end{aligned} \quad (3.4)$$

We can still treat the given six paths as a PICC in our optimization problem (3.1) to (3.3) even though they do not satisfy Condition 2 of Proposition 1.3.1. To that end, we simply need to generalize the indicator function $H_{ij}^l(e)$ and let $H_{ij}^l(e) = 2$ (rather than 1) for any bottleneck edge. In this way, we allocate *double* the capacity for such e (see (3.2)), which thus resolves the corresponding bottleneck caused by (3.4). With the use of a generalized indicator $H_{ij}^l(e)$, searching for PICCs is equivalent to searching for paths plus combining six paths as a group, which can be achieved by any path-search algorithms. Note that the larger the path collection \mathcal{P}_i and the PICC collection \mathcal{PICC}_{ij} , the higher achievable throughput will be. Depending on the available resources, there thus exists a complexity-performance tradeoff on how *exhaustive* the path-search algorithm should be.

The optimal solution of (3.1–3.3) can be achieved in a decoupled way by solving its dual problem via the sub-gradient method.

Algorithm A:

Rate Update For each s_i , update its rate vector $\mathbf{x}_i[t] = \{x_i^k[t], x_{ij}^l[t] : \forall k, j, l\}$ for the t -th time slot by

$$\begin{aligned} \mathbf{x}_i[t] = \arg \max_{0 \leq \mathbf{x}_i \leq M_X} U_i & \left(\begin{array}{c} |\mathcal{P}(i)| \\ x_i^k + \\ k=1 \end{array} \begin{array}{c} |\mathcal{P}ICC_{ij}| \\ x_{ij}^l \\ j:j=i \quad l=1 \end{array} \right) \\ - q_e[t] & \left(\begin{array}{c} |\mathcal{P}_i| \\ H_i^k(e)x_i^k + \\ k=1 \end{array} \begin{array}{c} |\mathcal{P}ICC_{ij}| \\ \frac{H_{ij}^l(e)x_{ij}^l}{2} \\ j:j=i \quad l=1 \end{array} \right) \\ - & \begin{array}{c} |\mathcal{P}ICC_{ij}| \\ q_{ij}^l[t]x_{ij}^l - q_{ji}^l[t]x_{ij}^l \\ l=1 \quad j:j>i \quad j:j<i \end{array} \\ - \alpha_i & \left(\begin{array}{c} |\mathcal{P}_i| \\ (x_i^k - y_i^k)^2 + \\ k=1 \end{array} \begin{array}{c} |\mathcal{P}ICC_{ij}| \\ (x_{ij}^l - y_{ij}^l)^2 \\ j:j=i \quad l=1 \end{array} \right), \end{aligned}$$

where $q_e[t]$ and $q_{ij}^l[t]$ are dual variables at the t -th time slot, whose values are feedback to s_i . The α_i are small constants and $\mathbf{y}_i = \{y_i^k, y_{ij}^l : \forall j, k, l\}$ are auxiliary variables of the proximal method in order to eliminate oscillation [63]. Periodically, \mathbf{y}_i is set to $\mathbf{x}_i[t]$ and the iteration continues using the new \mathbf{y}_i .

Scheduling Update The network selects the optimal scheduling policy for the t -th time slot by

$$\mathbf{r}[t] = \arg \max_{\mathbf{r} \in \mathcal{R}} \sum_{e \in E} q_e[t] r_e. \quad (3.5)$$

Queue-length Update Each link e updates its dual variable $q_e[t+1]$ according to the following equation.

$$q_e[t+1] = \left[q_e[t] + \beta_e \left(\sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e)x_i^k[t] + \sum_{(i,j):i=j} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} \frac{H_{ij}^l(e)x_{ij}^l[t]}{2} - r_e[t] \right) \right]^+,$$

where $[\cdot]^+ \triangleq \max(\cdot, 0)$ is the projection operator and β_e is a small step size for the sub-gradient method.

Balance Update Each destination d_i updates the dual variable $q_{ij}^l[t + 1]$ for all $j > i$. The dual variable q_{ij}^l accounts the difference between packet rates of sources i and j that use the same PICC.

$$q_{ij}^l[t + 1] = q_{ij}^l[t] + \beta_i (x_{ij}^l[t] - x_{ji}^l[t]), \forall j : j > i, \forall l, \quad (3.6)$$

where β_i is a small step size for the sub-gradient method.

Proximal Update Periodically, after every K time slots, set $\mathbf{y}_i \leftarrow \mathbf{x}_i[t]$. For notational simplicity, after the proximal update, we reset the timer value $t \leftarrow 0$.

The five different parts of Algorithm A are coupled implicitly via the queue lengths q_e and the balance information q_{ij}^l at the destinations. One important observation is that with PINC, only the rate and the balance updates, performed at the sources s_i and destinations d_i , differ from its non-coding counterpart (cf. [90]). The scheduling and queue-length updates remain identical. The impact of PINC on rate-control and scheduling is thus minimal and confined only in sources and destinations.

The complexity of Algorithm C depends mainly on the number of feedback messages $q_{ij}^l[t]$ that each source receives at each time slot t , which is proportional to the number of PICCs in the network. Therefore, the number of queue-length exchange messages is of the order $O(N \times |E| \times (\max_i \sum_{j:i=j} |\mathcal{PICC}_{ij}|))$. This is of similar complexity to that of traditional multipath routing with scheduling and congestion control [90] typically proposed for static mesh networks without mobility. Distributed methods that reduce the number of need-to-be-considered PICCs can be found in Section 3.6 and in [91], which mitigate the complexity of this algorithm and is executed only once in the initialization phase. Another approach for complexity reduction is to include the paths that form PICCs one by one in an adaptive way such that the number of control messages exceed a threshold as explained in [92]. For comparison, the number of multicast sessions used in the framework of [27] is $\binom{N}{2} \times |V| \times (|V| - 1)$, and $\binom{N}{2} \times |V| \times (|V| - 1)$, and the number of queue-length exchange messages in the corresponding back-pressure algorithm is thus $O(N^2 \times |V|^3 \times \overline{\text{nbs}})$, where $\overline{\text{nbs}}$ is the average number of neighboring nodes for nodes in V .

3.4.1 Convergence Analysis of Algorithm A

Proposition 3.4.1 *Consider a decreasing non-negative sequence $\{\beta_\tau\}$ such that $\sum_{\tau=1}^{\infty} \beta_\tau \rightarrow \infty$ and $\sum_{\tau=1}^{\infty} (\beta_\tau)^2 < \infty$. If in the beginning of each proximal iteration, we reset the step sizes $\beta_e = \beta_i = \beta_\tau$ with $\tau = 1$. As the inner iteration proceeds, we use β_τ , $\tau = 1, \dots, K$ as the step sizes in the K inner iterations. Then when the update period K of the proximal variable $\mathbf{y}_i \leftarrow \mathbf{x}_i[t]$ is sufficiently large, Algorithm A converges to the optimal solution of (3.1–3.3), the optimal rate assignment of PINC.*

A sketch of the proof is as follows. The boundedness of the rate region $\text{Co}(\mathcal{R})$ and rate-vector \mathbf{x} implies the boundedness of the sub-gradient of the dual problem of (3.1–3.3). Proposition 8.2.6 in [93] then guarantees the convergence. A detailed proof is relegated to [94]. The convergence with K bounded away from infinity and β_τ bounded away from zero is empirically verified during our simulations.

3.4.2 Stability of Algorithm A

Definition 3.4.1 *A system load $\{w_i : i = 1, \dots, N\}$ (we sometimes use $\{w_i\}_i$ as shorthand) can be stabilized by Algorithm A if there exists a non-negative vector $\mathbf{w} = \{w_i^k, w_{ij}^l : \forall i, j, k, l\}$ such that*

$$w_i = \sum_{k=1}^{|\mathcal{P}_i|} w_i^k + \sum_{j:j=i} \sum_{l=1}^{|\mathcal{P}ICC_{ij}|} w_{ij}^l, \forall i,$$

and $w_{ij}^l = w_{ji}^l, \forall (i, j) : i < j, \forall l.$ (3.7)

Moreover, if we replace the “rate update” in Algorithm A by a fixed rate assignment $\mathbf{x}[t] = \mathbf{w}$, then the dual variables $q_e[t]$ and $q_{ij}^l[t]$ must stay bounded away from infinity when t tends to infinity.

Let Λ denote a set of system loads $\Lambda = \{\{w_i\}_i\}$ such that for any $\{w_i\}_i \in \Lambda$, there exists a rate vector $\mathbf{r} \in \text{Co}(\mathcal{R})$, a non-negative vector $\mathbf{w} = \{w_i^k, w_{ij}^l : \forall i, j, k, l\}$ satisfying (3.7), and jointly \mathbf{w} and \mathbf{r} satisfy

$$\sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e) w_i^k + \sum_{(i,j):i=j} \sum_{l=1}^{|\mathcal{P}^{ICC}_{ij}|} \frac{H_{ij}^l(e) w_{ij}^l}{2} \leq r_e, \forall e \in E.$$

We then have the following stability result regarding the system load region Λ .

Proposition 3.4.2 *Any system load $\{w_i\}_i$ that is in the interior of Λ can be stabilized by the optimal rate-control/scheduling in Algorithm A.*

(Proposition 3.4.2 can be regarded as a corollary of Proposition 3.5.1 that will be introduced shortly after.)

3.5 Pairwise Intersession Network Coding with Imperfect Scheduling

In general, it is computationally expensive to find the optimal scheduling decision satisfying (3.5) in Algorithm A. Depending on different interference models, finding the optimal scheduling \mathbf{r} that maximizes $\sum_e q_e[t] r_e$ is NP-hard in many cases and generally requires centralized implementation. In practice, we would often have to resort to imperfect scheduling schemes that select the rate vector $\mathbf{r}[t]$ that achieves γ fraction of the maximum value. Namely, an imperfect scheduling policy choose $\mathbf{r}[t]$ satisfying

$$q_e[t] r_e[t] \geq \gamma \max_{\mathbf{r}} \sum_{e \in E} q_e[t] r_e, \quad (3.8)$$

where γ is a constant in $[0, 1]$. With imperfect scheduling ($\gamma < 1$), the tie between Algorithm A and the gradient method for the dual problem is severed and Algorithm A may not converge to any fixed-point solution. The following results show that even with imperfect scheduling, the proposed PINC scheme with cross-layer optimization Algorithm A still shows tractable performance in terms of the stability region.

Proposition 3.5.1 *Any system load $\{w_i\}_i$ that is in the interior of $\gamma\Lambda$ can be stabilized by Algorithm A with γ -imperfect scheduling.*

The sketch of the proof is provided in Appendix B.1, which covers Proposition 3.4.2 as a special case.

3.5.1 Networks with Dynamic Arrivals and Departures

In addition to networks with static arrivals and departures, we also consider the case of dynamic system loads with logarithmic utility functions. Consider N classes of users. For all i , users in class i have a common logarithmic utility function $U_i(x) = \kappa_i \log(x)$ where $\kappa_i > 0$ are predefined system parameters. All users in class i will send packets from s_i to d_i and will use the same routing paths in \mathcal{P}_i and the same PICCs in $\mathcal{P}ICC_{ij}$ for transmission. We also assume that users of class i arrive according to a Poisson process with rate λ_i and each user needs to send a file whose size is exponentially distributed with mean $\frac{1}{\mu_i}$. The system load of this network with dynamic arrivals is then defined as $\left\{ \left(\frac{\lambda_i}{\mu_i} \right) : \forall i \right\}$. The dynamic nature of this setting prompts a slightly different definition of stability.

Definition 3.5.1 *A system load $\left\{ \left(\frac{\lambda_i}{\mu_i} \right) : \forall i \right\}$ can be stabilized by Algorithm A if the dual variables $q_e[t]$ and $q_{ij}^l[t]$ are bounded away from infinity for each iteration with probability one.¹*

We then have the following stability result.

Proposition 3.5.2 (Stability for Dynamic Systems) *Consider logarithmic utility functions $U_i(x) = \kappa_i \log(x)$. With sufficiently small α_i , β_e , and β_i , any system load $\left\{ \left(\frac{\lambda_i}{\mu_i} \right) : \forall i \right\}$ that is in the interior of $\gamma\Lambda$ can be stabilized by Algorithm A with γ -imperfect scheduling.*

The proof of Proposition 3.5.2 is sketched in Appendix B.2. Proposition 3.5.2 implies that although the instantaneous system load imposed on the network may well exceed the network capacity, as long as the average system load is within γ times

¹In contrast with Definition 3.4.1 where the rate update rule is modified for a static system load, for a dynamic system load, the optimal rate update is kept unchanged. Only the scheduling update will be changed to incorporate imperfect scheduling as in (3.8).

the capacity, the queue lengths of the network are bounded away from infinity and the system is stable.

Proposition 3.5.2 shows that the graceful stability degradation that was previously known only for non-coding transmission (cf. [54]) also holds for PINC. Shifting from non-coding to network-coding solutions enhances the throughput without sacrificing the associated stability even with imperfect scheduling.

3.5.2 A Generalized Node Exclusive Interference Model With the WMA

The node exclusive model is a commonly used interference model for bluetooth or for FH-CDMA networks [95–97] that admits efficient and provably good approximation of the optimal scheduling policy. In the traditional node exclusive model (without taking advantages of the WMA), *the data rate of each link is fixed at c_e and each node can only send to or receive from one other node at any time*. The objective function of optimal scheduling is thus equivalent to

$$\max_{\mathbf{r}} \sum_{e \in E} q_e[t] r_e = \max_{\mathcal{M}} \sum_{e \in E} q_e[t] c_e 1_{\{e \in \mathcal{M}\}} = \max_{\mathcal{M}} \sum_{e \in \mathcal{M}} q_e[t] c_e, \quad (3.9)$$

where \mathcal{M} is a matching of the underlying graph G . Finding the optimal scheduling of (3.9) thus becomes a maximum weighted matching problem.

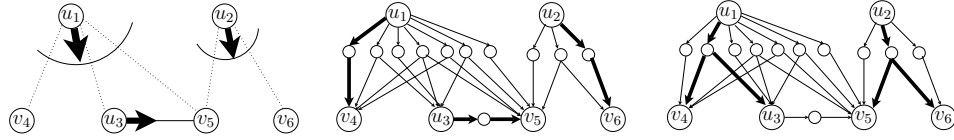
Nonetheless, when the WMA is taken into account, i.e. with the auxiliary nodes added for the broadcast nature of wireless transmission as discussed in Section 3.3.1, the objective function of scheduling becomes

$$\max_{\mathbf{r}} \sum_{e \in E} q_e[t] r_e = \max_{\mathcal{A}} \sum_{e \in E} q_e[t] c_e 1_{\{e \text{ is adjacent to some node in } \mathcal{A}\}}, \quad (3.10)$$

where \mathcal{A} is a set of *active auxiliary nodes*. Since each network node can only send to or receive from one auxiliary node (due to the node-exclusiveness assumption), we require that the node set \mathcal{A} satisfies that any node in \mathcal{A} does not share any common neighbor with any other node in \mathcal{A} .

Fig. 3.5(a) depicts a wireless network of six nodes. Nodes u_1 , u_2 , and u_3 would like to transmit and the transmission can be overheard by more than one receivers (see

Fig. 3.5(a)). Two possible scheduling policies (two different \mathcal{A} s) are illustrated. In Fig. 3.5(b), the wireline counterpart of Fig. 3.5(a), three auxiliary nodes are active and correspond to three *unicast transmissions* $((u_1, v_4), (u_2, v_6), \text{ and } (u_3, v_5))$ highlighted by thick edges. The other scheduling policy contains two active auxiliary nodes as in Fig. 3.5(c) that correspond to two *broadcast transmissions* $((u_1, \{v_4, u_3\}) \text{ and } (u_2, \{v_5, v_6\}))$.



(a) Consider the WMA of a wireless network. (b) Three active auxiliary nodes corresponding to three unicast transmissions. (c) Two active auxiliary nodes corresponding to two broadcast transmission.

Fig. 3.5. Illustration of the node exclusive model when the WMA is taken into consideration.

It can be shown that maximizing (3.10) is equivalent to solving a maximum weighted *hypergraph matching* (MWHM) problem. In [25] a greedy maximal hypergraph matching (GMHM) is proposed as an approximation of the MWHM. More explicitly, the network first selects an auxiliary node v_a that maximizes

$\sum_{e \text{ is adjacent to } v_a} q_e[t]c_e$ and includes v_a as part of the scheduling policy \mathcal{A} . Remove v_a and its neighbors and then restart this greedy selection of auxiliary node until a maximal scheduling policy \mathcal{A} is reached. In this way, GMHM guarantees to find a $\frac{1}{\max_{v_a} |\text{nbs}(v_a)|}$ -approximation of the MWHM, where $\text{nbs}(v_a)$ is the set of neighbors around v_a . We can further sharpen the approximation ratio as follows.

Proposition 3.5.3 *For any given network, the GMHM is a $\frac{1}{5}$ -approximation algorithm and can thus achieve at least $\frac{1}{5}$ of the stability region Λ when used as an imperfect scheduling policy for the node-exclusive model.*

Proof Since each PICC consists of six paths, any auxiliary node participating in a PICC has at most six outgoing branches plus one incoming edge. As a result, during the wireless to wireline conversion, there is no need to include auxiliary nodes of $> (6 + 1)$ neighbors as the neighbors of those nodes will not fully participate in a PICC. The above reasoning shows that the GMHM can be made a $\frac{1}{7}$ -approximation by eliminating the auxiliary nodes with > 7 neighbors. By further taking into account the edge-overlap conditions in Proposition 1.3.1, it can be shown that each auxiliary node needs to have at most four outgoing branches (two for the \mathcal{P} paths and two for the \mathcal{Q} paths). Therefore, the approximation ratio can be improved to $\frac{1}{5}$ by eliminating the auxiliary nodes with $> (4 + 1)$ neighbors. The proof is complete. ■

It is worth mentioning that the $\frac{1}{5}$ -approximation is a lower bound of the performance of the GMHM. In our numerical study, GMHM has almost identical performance to the optimal MWHM solution. Other studies on the performance of greedy maximal matching for non-coding networks can be found in [98, 99].

3.6 Distributed Code Design for PINC

The rate control and link scheduling algorithms described thus far allocate optimal rates at each link so that the utility function can be maximized. The next question is what is the network coding scheme that can achieve the optimal rate assignment? For a given PICC, we proposed a coding scheme based on identifying distributedly special edges in the PICC in [92]. Specifically, carefully chosen *decoding* operations are performed on special decoding edges, while random network coding is performed on all other edges for the sake of scalability and distributiveness. In this chapter we present a new, more efficient approach in which each edge e decides the coding operation based on the subset of the six paths of the given PICC that use e . Since each edge naturally knows whether itself participates in a given route/path or not (a byproduct of the initial path-search phase of Algorithm C), the corresponding coding operation can be decided locally without knowing the entire topology of the network.

Moreover, this scheme only uses binary XOR operations, which has computational advantages over schemes based on a large finite field $\text{GF}(2^8)$ or $\text{GF}(2^{16})$. The new binary scheme achieves the same optimal throughput as that of [91] while the later requires the use of a larger field.

Practical network coding [85] uses the concept of “generations” that synchronize the network operations as coding is performed only within the same generation. With appropriate route/path selection and a carefully-designed generation-flushing policy, packets seldom cycle in the network. For the following, we thus restrict our attention to acyclic networks. In [91], it is observed that some PICCs have negligible impact on scheduling/rate-control and can be absorbed by a pair of edge-disjoint paths or by other PICCs. Fig. 3.6(a) represents one such insignificant PICC. The \mathcal{P} and \mathcal{Q} paths in Figs. 3.6(b) and 3.6(c) verify that Fig. 3.6(a) is indeed a PICC. However, within Fig. 3.6(a), there exists a pair of edge-disjoint paths P_{s_1,d_1} and P_{s_2,d_2} as illustrated in Fig. 3.6(d). One can send symbols X and Y along the edge-disjoint paths without using up all available bandwidth in the given PICC. From the throughput/cost perspective, the pair of edge-disjoint paths (Fig. 3.6(d)) *dominates* the given PICC (Fig. 3.6(a)), the latter of which is thus insignificant in the rate-control/scheduling analysis and can be removed from consideration without affecting the optimality of the solution. Let (s_1, d_1) and (s_2, d_2) denote the pair of unicast sessions of interest.

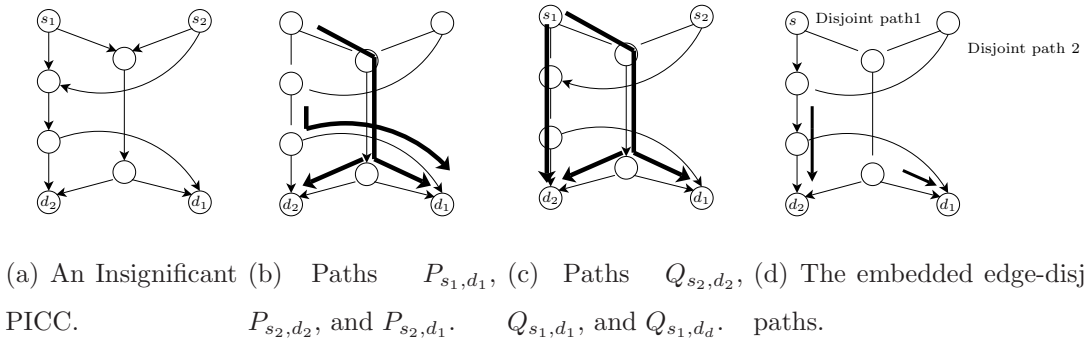


Fig. 3.6. Illustration of an insignificant PICC that contains a pair of edge-disjoint paths as a strict subgraph.

For acyclic networks, the following four rules² identify the insignificant PICCs (each consisting of three \mathcal{P} paths and three \mathcal{Q} paths).

- **Rule 1:** If P_{s_2,d_1} and Q_{s_1,d_2} meet at any edge, the PICC is insignificant.

Depending on whether P_{s_1,d_1} and Q_{s_1,d_1} are the same path (and symmetrically whether P_{s_2,d_2} and Q_{s_2,d_2} are the same path), we have the following three more rules.

- **Rule 2.1:** Suppose $P_{s_1,d_1} = Q_{s_1,d_1}$ and $P_{s_2,d_2} \neq Q_{s_2,d_2}$. If there exists an edge e shared by all three paths P_{s_1,d_1} , P_{s_2,d_2} , and Q_{s_2,d_2} , then the PICC is insignificant.
- **Rule 2.2:** Suppose $P_{s_1,d_1} \neq Q_{s_1,d_1}$ and $P_{s_2,d_2} = Q_{s_2,d_2}$. If there exists an edge e shared by all three paths Q_{s_2,d_2} , Q_{s_1,d_1} , and P_{s_1,d_1} , then the PICC is insignificant.
- **Rule 2.3:** Suppose $P_{s_1,d_1} \neq Q_{s_1,d_1}$ and $P_{s_2,d_2} \neq Q_{s_2,d_2}$. Declare the PICC as insignificant.

Rules 1 to 2.3 can be implemented distributedly in the initialization phase by sending tokens along the paths to explore whether the paths share a given edge. For example, the insignificant PICC in Fig. 3.6(a) can be identified by Rule 1 and removed from consideration. Our new XOR-based scheme is then performed on the remaining PICCs that are not removed by the above four rules. The detailed description of the code construction is as follows.

The source: Source s_1 sends its own symbol X along P_{s_1,d_1} , Q_{s_1,d_1} , and Q_{s_1,d_2} , and s_2 sends its own symbol Y along P_{s_2,d_2} , Q_{s_2,d_2} , and P_{s_2,d_1} .

Each intermediate edge: At each edge e that is the outgoing edge of an intermediate network node $v \neq s_1, s_2$. Consider the following cases.

- **Case 1:** $P_{s_1,d_1} = Q_{s_1,d_1}$ and $P_{s_2,d_2} \neq Q_{s_2,d_2}$. Consider four sub-cases. **Case 1.1:** If all incoming edges of v carry the same symbol, then forward that symbol. **Case 1.2:** If Case 1.1 is not satisfied and $e \notin P_{s_1,d_1}$, then send Y through e .

²A detailed proof of the correctness of these four rules and the corresponding distributed implementation can be found in [94].

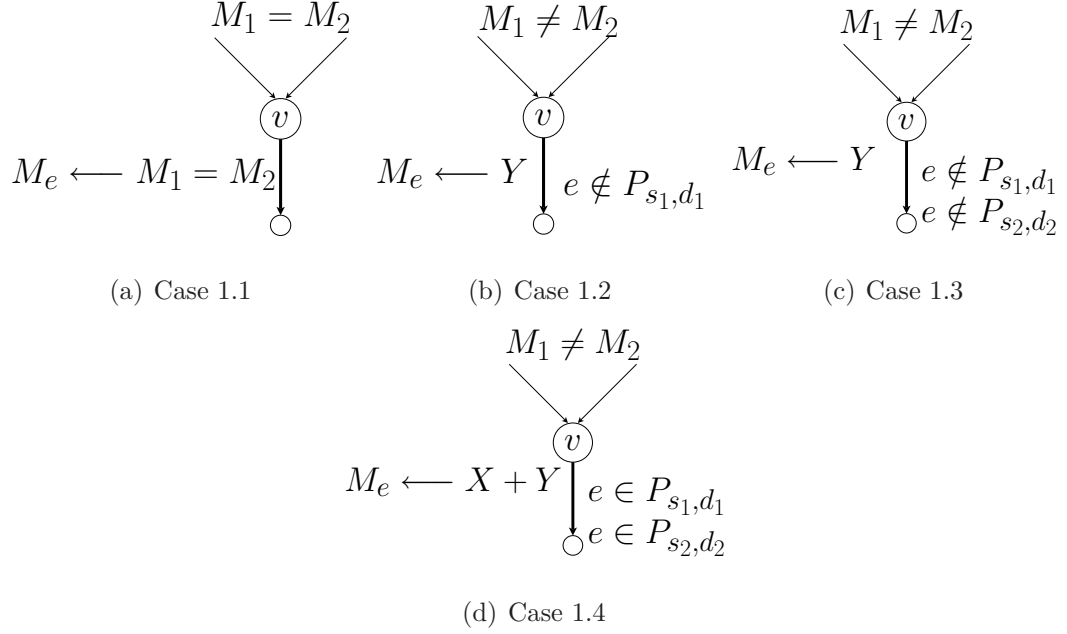


Fig. 3.7. Cases 1.1 to 1.4 of the new distributed code construction using only the binary XOR operation. M_1 and M_2 are the coded symbols along the two incoming edges of v . M_e is the outgoing coded symbol along edge e .

Case 1.3: If Case 1.1 is not satisfied, $e \in P_{s_1,d_1}$, and $e \notin P_{s_2,d_2}$, then send X through e . **case 1.4:** If Case 1.1 is not satisfied, $e \in P_{s_1,d_1}$, and $e \in P_{s_2,d_2}$, then send the binary XORed symbol $X + Y$ through e .

- **Case 2:** $P_{s_1,d_1} \neq Q_{s_1,d_1}$ and $P_{s_2,d_2} = Q_{s_2,d_2}$. This is a symmetric case of Case 1. We perform the symmetric operations of Case 1 by swapping the roles of the first unicast session (s_1, d_1) , X , and \mathcal{P} with the roles of the second session (s_2, d_2) , Q and Y .
- **Case 3:** $P_{s_1,d_1} = Q_{s_1,d_1}$ and $P_{s_2,d_2} = Q_{s_2,d_2}$. Perform the same operations as in Cases 1.1 to 1.4.

Fig. 4.1 illustrates Cases 1.1 to 1.4 for an outgoing edge e of an intermediate node v .

Proposition 3.6.1 *For an acyclic network, consider a PICC that is not removed by Rules 1 to 2.3. Then destination d_1 (resp. d_2) is able to recover the designated symbol X (resp. Y) using the above locally computed binary coding scheme.*

The proof is relegated to Appendix B.3.

3.7 Numerical Experiments

We perform simulations under the linear signal-to-noise-&-interference-ratio (SINR) model. The objective is to compare the non-coded and the network coding solutions with both perfect and imperfect scheduling. To implement imperfect scheduling, we maintain a smaller pool imperfect scheduling policy $\tilde{\Theta} \subsetneq \Theta$ and choose the imperfect scheduling from the smaller policy pool $\tilde{\Theta}$ in a similar way as in [54] according to the following. Every scheduling policy $\theta \in \tilde{\Theta}$ is associated with a rate vector r_e^θ . Further assume that every $\theta \in \tilde{\Theta}$ is associated with a set of queue lengths $\{q_e^\theta : \forall e\}$ such that the policy θ is a γ_θ -approximation policy satisfying $\sum_{e \in E} q_e^\theta r_e^\theta \geq \gamma_\theta \max_{\mathbf{r}} \sum_{e \in E} q_e^\theta r_e$. If the following condition holds in the t -th time slot

$$\max_{\substack{\theta \in \tilde{\Theta} \\ e \in E}} q_e[t] r_e^\theta \geq \gamma \min_{\substack{\theta \in \tilde{\Theta} \\ e \in E}} [q_e[t] - q_e^\theta]^+ r_e^{\max} + \frac{\sum_{e \in E} q_e^\theta r_e^\theta}{\gamma_\theta} ,$$

for some γ , where r_e^{\max} is the maximum possible rate along edge e , then policy θ_a^* that maximizes the left-hand side is a γ -approximation of the optimal scheduling policy with weights $q_e[t]$ on each edge. We can use such a scheduling policy θ_a^* in the reduced policy pool $\tilde{\Theta}$ without the computationally expensive step of computing the optimal scheduling policy θ^* in the right-hand side of (3.8). If no such θ_a^* exists, we compute directly one $\theta[t]$ satisfying (3.8) and store this new $\theta[t]$ and the associated $q_e[t]$ in the small pool $\tilde{\Theta}$.

We assume that the total power assigned to node u at any time slot is bounded by $P_{u,\max}$. To achieve the optimal throughput, in each time slot, each node u should either transmit at full power $P_{u,\max}$ or remain silent. For any unicast transmission

from u to v , the data rate $r_{u,v}$ is assumed to be proportional to the SINR level³ at the receiver v , which is formally expressed as.

$$r_{uv} = W \frac{G(u, v) 1_{\{(u,v) \text{ is activated}\}} P_{u,\max}}{N_0 + \sum_{w:w=u} G(w, v) 1_{\{\text{node } w \text{ is sending}\}} P_{w,\max}},$$

where N_0 is the background noise, W is the bandwidth of the system, and $G(u, v)$ is the path gain between nodes u and v which is set to $(\text{dist}(u, v))^{-4}$, where $\text{dist}(u, v)$ is the Euclidean distance between nodes u and v . With network coding, the data rate of the broadcast link with multiple receivers is proportional to the minimum of the SINR levels at those receivers. More precisely, if node u is broadcasting to nodes v_1, \dots, v_n , the data rate of this broadcast link, $r_{u,\{v_1, \dots, v_n\}}$, becomes

$$r_{u,\{v_1, \dots, v_n\}} = W \min_{\{i=1, \dots, n\}} \left\{ \frac{G(u, v_i) 1_{\{(u,\{v_1, \dots, v_n\}) \text{ is activated}\}} P_{u,\max}}{N_0 + \sum_{w:w=u} G(w, v_i) 1_{\{w \text{ is sending}\}} P_{w,\max}} \right\}.$$

We run the simulations on the topology in Fig. 3.8. The X- and Y-coordinates of the

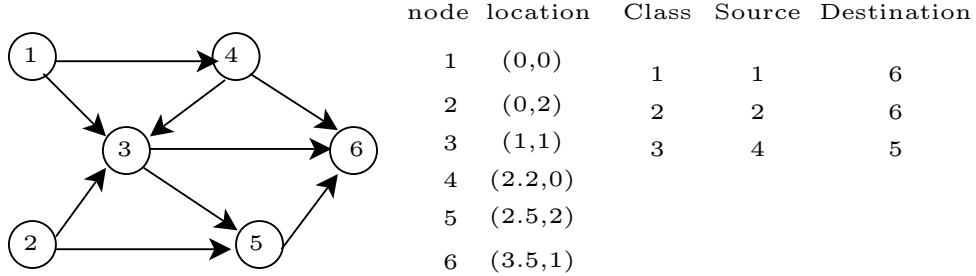


Fig. 3.8. The network topology, node locations, and three classes of unicast traffic used in the simulations.

six network nodes are specified in the figure. We simulate three classes of users and each class is allowed to use multi-path or multi-PICC communications. The source and destination pair of each class is also shown in the figure. A logarithmic utility function $U(\cdot) = \log(\cdot)$ is assumed for all classes. In our simulation, we use $W = 10$, $N_0 = 1$, $P_{u,\max} = 1$, $\forall u$, the proximal coefficient $\alpha_i = 0.01$, the step sizes $\beta_e = 0.01$, $\forall e$,

³The linear SINR model can be viewed as a first order approximation of the information-theoretic $W \log(1 + \text{SINR})$ model.

$\beta_i = 0.01, \forall i$, and 10 inner iterations within each proximal iteration $K = 10$. Fig. 3.9 represents the results for the case of deterministic arrival and departure. Using the non-coded solution with perfect scheduling the rates of classes 1 and 2 converge to about 0.377 and the rate of class 3 converges to 0.325. When imperfect scheduling is used without network coding and $\gamma = 0.6$ the rates of classes 1 and 2 remain the same as the perfect scheduling case and the rate of class 3 is reduced by only 0.02. The number of time slots in which new schedules need to be computed is 18 out of totally 5000 time slots (500 proximal iterations). When we further reduce γ to $\frac{1}{3}$, the rate of class 1 is 4.0, the rate of class 2 is 3.5, and the rate of class 3 is 3.2 which shows a deviation from the fairness point. The number of time slots in which new schedules need to be computed is 7 out of totally 5000 time slots.

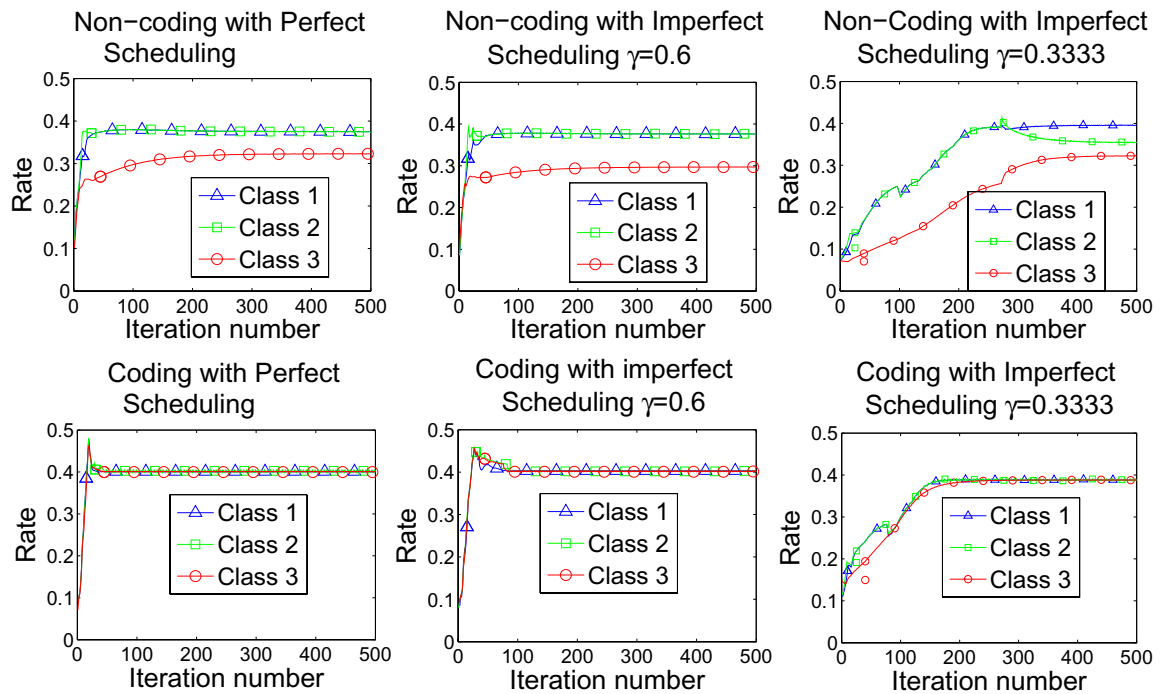


Fig. 3.9. The convergence results for the case of deterministic arrival with the linear SINR-based interference model.

The PINC solution with both the perfect and imperfect scheduling with $\gamma = 0.6, \frac{1}{3}$ achieves strict fairness as the data rates of all classes converge to 4 as shown in Fig. 3.9. Using imperfect scheduling from a reduced pool of scheduling policies, the new schedules need to be computed in only 21 time slots when $\gamma = 0.6$ and in only 13 time slots when $\gamma = \frac{1}{3}$. With network coding, the computationally efficient imperfect scheduling method outperforms the non-coding solution with optimal scheduling from both the throughput and fairness perspectives. To show that the performance gain of PINC is universal for other channel models, we have also simulated the same topology with a $W \log(1 + SINR)$ model. Similar performance gain is observed in Fig. 3.10.

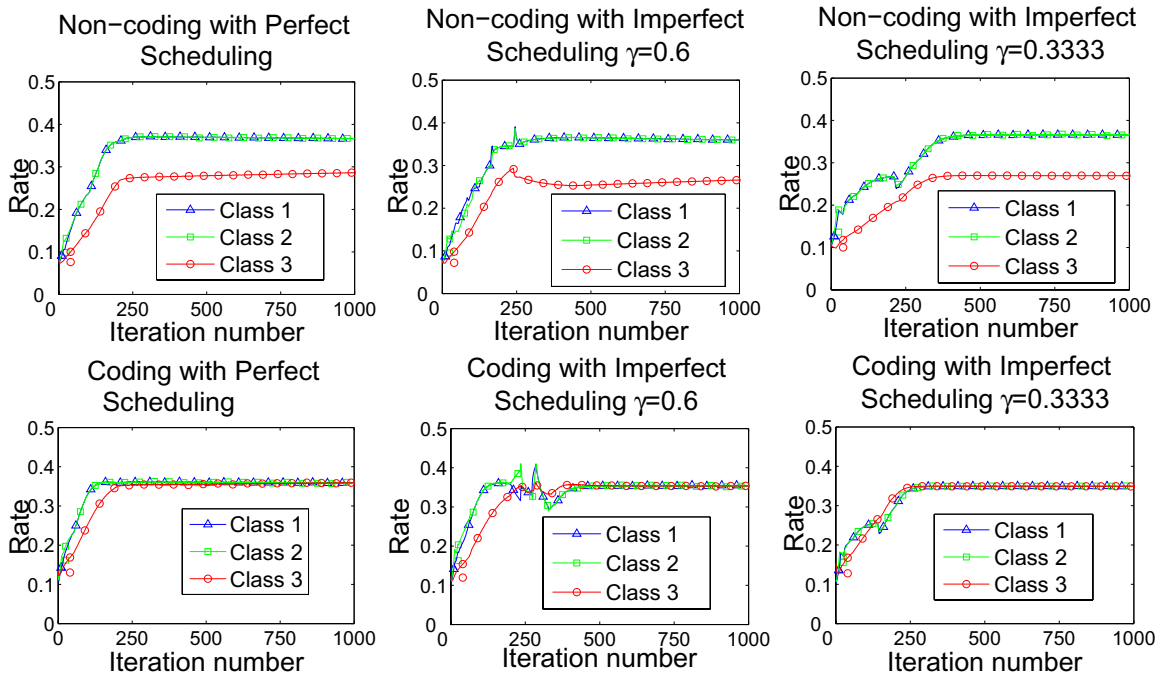


Fig. 3.10. The convergence results for the case of deterministic arrival with the $W \log(1 + SINR)$ -based interference model.

For the dynamic arrival and departure, we simulated the arrival of files for each class whose size is exponentially distributed with average size ($\frac{1}{\mu} = 100$). Each file

arrives according to a Poisson process with rate λ . We vary the rate λ and report in Fig. 3.11 the average number of users in the system with respect to the system load per user $\rho \triangleq \frac{\lambda}{\mu}$. As shown in the figure network coding with imperfect scheduling outperforms the non-coded solution with perfect scheduling by a significant 20%.

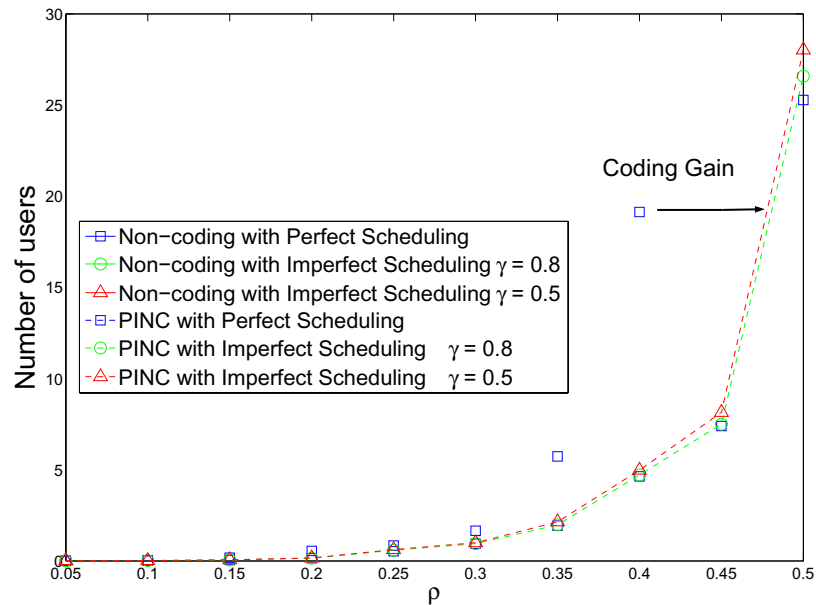


Fig. 3.11. The number of users in the system versus the system load for the case of dynamic arrival with the linear SINR-based interference model.

3.8 Conclusion

For intersession network coding, coding across many sessions requires greater transmission power to broadcast the coded symbol to many receiver, which results in higher interference in the wireless multi-hop network. In both empirical and analytical studies, it has been shown that for an interference/energy aware network coding scheme, most of the coding opportunities involve only two sessions, referred herein as pairwise intersession network coding (PINC).

In this chapter, we have proposed a jointly optimal coding, scheduling, and rate-control scheme for wireless multi-hop networks based on the recent theoretical finding of PINC. The corresponding coding, scheduling, and rate-control components are decoupled by the use of queue lengths and the introduction of rate-balance dual variables. Our results have proven that in a wireless multi-hop network, the throughput advantage of PINC can be achieved without sacrificing the stability conditions. Moreover, PINC has minimal impact on the optimal rate-control/scheduling as the only new component necessary for scheduling PINC traffic is the *balance update* performed at the receivers. Following this new formulation, we have also studied the impact of γ -imperfect scheduling on PINC-based rate-control algorithm and for the corresponding distributed greedy hypergraph matching algorithm.

Numerical experiments have also been conducted for the linear and the logarithmic signal-to-noise/interference-ratio (SINR) models, which shows that the achievable rates using PINC and efficient imperfect scheduling outperforms that of non-coding transmission with computationally expensive optimal scheduling.

The approach proposed in this chapter is a path-based approach which converges faster than the back-pressure approach. However, the path based approach in this chapter does not allow the decoded packet to be reencoded again. In the next chapter we study the back-pressure approach that provides a potentially bigger capacity region by allowing the decoded packet to be reencoded again. The back-pressure approach can be used for both rate control and energy minimization.

4. BACK-PRESSURE ALGORITHM FOR FULL UTILIZATION OF WIRELESS MULTIHOP NETWORKS WITH PAIRWISE INTERSESSION NETWORK CODING

4.1 Introduction and related work

In this chapter we develop a back-pressure algorithm for pairwise intersession network coding. In the back-pressure approach, paths are not fixed unlike the path-based approach in Chapter 3. This feature of the back-pressure approach allows finding all of the PINC opportunities, reencoding the decoded packets again, and reducing the complexity of implementing PINC in wireless networks. Our main contributions in this chapter that distinguish it from previous chapters are as follows:

- Development of an optimization framework that takes into account the transmission power, broadcast nature of wireless links, link scheduling, and the pairwise intersession network coding possibilities. Existing works either consider the non-network coding solutions [54, 60, 61, 100], use intersession network coding but limit the encoding and decoding nodes to be neighbors as in [26, 27, 76], consider specific structures for intersession network coding opportunities as the butterfly in [40, 41, 44], or do not allow decoded packets to be reencoded again as in Chapter 3. In contrast our framework considers all pairwise intersession network coding opportunities within κ -hops, where κ is an arbitrary integer balancing the tradeoff between complexity and the achievable capacity. Our approach also allows the decoded packet to be reencoded again.
- The framework allows us to develop cross-layer algorithms for both energy minimization and rate control and study their performance loss under suboptimal but distributed scheduling algorithms. Our performance loss results are tighter

than these in Chapter 3, because we carefully characterize the maximum broadcast link size with pairwise intersession network coding. Characterizing the maximum broadcast link size also simplifies the distributed implementation of the proposed algorithms and reduces their complexity.

The rest of the chapter is organized as follows. In Section 3.3 we review the characterization and implementation of pairwise intersession network coding in wireline as well as wireless networks. In Section 4.2 we provide a new result regarding the broadcast link size for pairwise intersession network coding in wireless networks. The formulation of our optimization framework is in Section 4.3 followed by distributed algorithms for energy minimization and rate control in Sections 4.4 and 4.5, respectively. Section 4.6 is devoted to practical implementation details of our algorithms. We present our simulation results in Section 4.7 and conclude the chapter in Section 4.8.

4.2 Broadcast Link size with PINC

To efficiently apply the WMA in Section 3.3.1 to PINC, we need to know the largest necessary broadcast link size that can provide *strict* throughput benefits over non-coded solutions when PINC is used. By excluding those broadcast links with size greater than the largest necessary, an efficient scheduling algorithm can be designed with provable approximation constant. Using the approach in Section 3.3.1, the number of added auxiliary nodes for node v grows exponentially in $\text{nb}(v)$, where $\text{nb}(v)$ is the number of neighbors for node v . By excluding the unnecessary broadcast links, the number of auxiliary nodes for node v grows in polynomial time with respect to $\text{nb}(v)$ (actually $(\text{nb}(v))^2$ as will be clear shortly after) which greatly reduces the complexity in dense mesh networks. Excluding unnecessary broadcast links also simplifies the formulation and the distributed algorithms in Sections 4.3, 4.4, and 4.5. The following proposition characterizes unnecessary broadcast links.

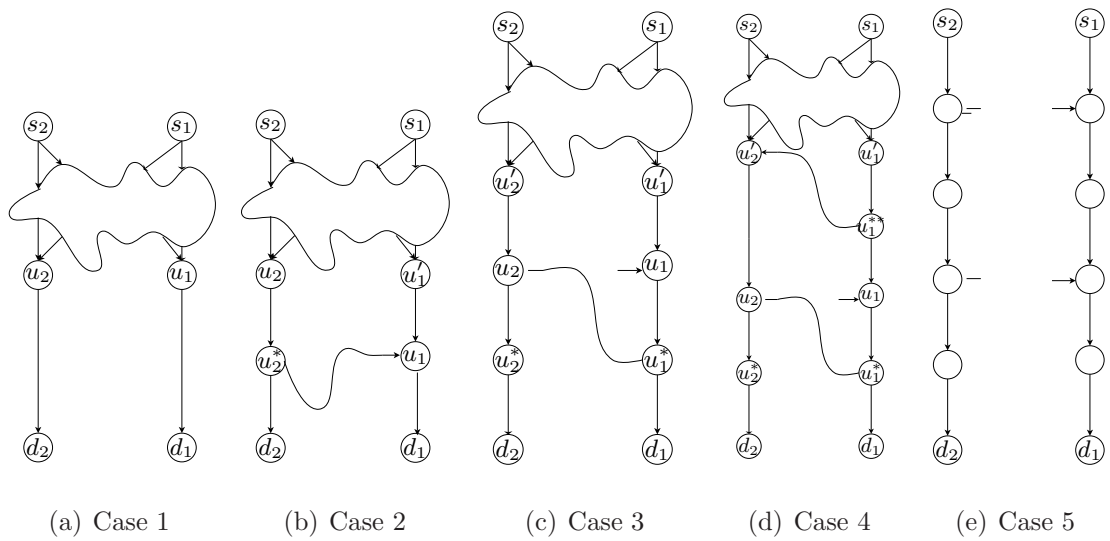


Fig. 4.1. Graphical representation of the different cases in the proof of Proposition 4.2.1

Proposition 4.2.1 *With the use of the WMA and PINC, ignoring broadcast links with more than two receivers does not affect the overall performance of any scheduling algorithm.*

We have the following corollary for the GMHM discussed in Chapter 3.

Corollary 4.2.1 *With PINC, GMHM guarantees to find a $\frac{1}{3}$ -approximation of the MWHM*

Proof This proposition follows for acyclic networks due to Theorem 3 in [78].

In [66] it has been shown that if a cyclic network contains no 2EDPs connecting (s_1, d_1) and (s_2, d_2) and satisfies the necessary and sufficient conditions for the existence of a PINC solution as in Theorem 1.3.1, then it must satisfy one of five cases. We will show that the Proposition holds in all of these cases. In all of these cases we will show that there exists two sets of path segments, and the path segments in each set are edge disjoint. Moreover, the sets of path segments satisfy that if we remove all edges in the network except those belonging to these two sets, condition 2 in Theorem 1.3.1 holds. Therefore, for any given edge e , at most two path segments use e (one from each set of path segments). Therefore, the number of links branching from a single edge or splitting after sharing an edge is at most two and there is thus no need to consider broadcast links with more than two receivers.

Before considering the cases, we use the following notations and definitions:

- We assume that the two unicast sessions are (s_1, d_1) and (s_2, d_2) .
- Critical 1-edge cut: For any node u in a cyclic network, the critical 1-edge cut e^* is a 1-edge cut separating $\{s_1, s_2\}$ and u that is the farthest from u . Namely, any path from $\{s_1, s_2\}$ to u will meet e^* before meeting any other 1-edge cut. If there exists 1-edge cut separating $\{s_1, s_2\}$ and u , then e^* always exists and is unique. A detailed discussion is in [66].
- We assume that $e_1 = (u_1, v_1)$ ($e_2 = (u_2, v_2)$) is the critical 1-edge cut separating $\{s_1, s_2\}$ and d_1 (d_2).

- $G \setminus e$: Denotes the subgraph induced by removing edge e from G .
- $P_{u_1, u_2} e_1 P_{v_1, v_2}$: is the path formed by the path segment P_{u_1, u_2} followed by edge e_1 and then followed by P_{v_1, v_2} .

Case 1: \exists 2EDP $P_{s_1, u_1}^{G \setminus e_2}, P_{s_2, u_1}^{G \setminus e_2}$ in $G \setminus e_2$ and \exists 2EDP $P_{s_1, u_2}^{G \setminus e_1}, P_{s_2, u_2}^{G \setminus e_1}$ in $G \setminus e_1$. We have 3 edge disjoint path segments $P_{s_1, u_1}^{G \setminus e_2}, P_{s_2, u_1}^{G \setminus e_2}, P_{u_1, d_1}^{G \setminus e_2}$ in $G \setminus e_2$ and other 3 edge disjoint path segments $P_{s_1, u_2}^{G \setminus e_1}, P_{s_2, u_2}^{G \setminus e_1}, P_{u_2, d_2}^{G \setminus e_1}$ in $G \setminus e_1$. We can construct the \mathcal{P} and \mathcal{Q} satisfying condition 2 of Theorem 1.3.1 by letting $P_{s_1, d_1} = Q_{s_1, d_1} = P_{s_1, u_1}^{G \setminus e_2} P_{u_1, d_1}^{G \setminus e_2}$, $P_{s_2, d_2} = Q_{s_2, d_2} = P_{s_2, u_2}^{G \setminus e_1} P_{u_2, d_2}^{G \setminus e_1}$, $P_{s_2, d_1} = P_{s_2, u_1}^{G \setminus e_2} P_{u_1, d_1}^{G \setminus e_2}$, and $Q_{s_1, d_2} = P_{s_1, u_2}^{G \setminus e_1} P_{u_2, d_2}^{G \setminus e_1}$. By the same reasons stated in the opening paragraph of the proof, Case 1 is proven. This case is shown in Figure 4.1(a).

Case 2: \exists 2EDP $P_{s_1, u_2}^{G \setminus e_1}, P_{s_2, u_2}^{G \setminus e_1}$ in $G \setminus e_1$, \nexists 2EDP $P_{s_1, u_1}^{G \setminus e_2}, P_{s_2, u_1}^{G \setminus e_2}$ in $G \setminus e_2$, and e_1 is reachable from both s_1 and s_2 in $G \setminus e_2$.

Let $e'_1 = (u'_1, v'_1)$ be the critical 1-edge cut for u_1 in $G \setminus e_2$. Therefore, we must have 2EDP $P_{s_1, u'_1}^{G \setminus e'_2}, P_{s_2, u'_1}^{G \setminus e'_2}$ in $G \setminus e'_2$. Also we must have path segments P_{u_2, u_1} and P_{u_1, d_1} . The first set of path segments contains the three edge disjoint path segments $P_{s_1, u_2}^{G \setminus e_1}, P_{s_2, u_2}^{G \setminus e_1}, P_{u_2, d_2}^{G \setminus e_1}$ in $G \setminus e_1$. Let u_2^* be the last vertex where P_{u_2, u_1} and $P_{u_2, d_2}^{G \setminus e_1}$ meets, the other set of path segments contains the following five path segments $P_{s_1, u'_1}^{G \setminus e'_2}, P_{s_2, u'_1}^{G \setminus e'_2}, P_{u'_1, u_1}^{G \setminus e_2}, P_{u_1, d_1}^{G \setminus e_2}$ and $P_{u_2^*, u_1}^{G \setminus e_2}$. $P_{u_1, d_1}^{G \setminus e_2}$ is edge disjoint from all the remaining four path segments because e_1 is the critical 1-edge cut for d_1 . $P_{u_2^*, u_1}^{G \setminus e_2}$ and $P_{u'_1, u_1}^{G \setminus e_2}$ are edge disjoint, because e'_1 is a critical 1-edge cut for u_1 in $G \setminus e_2$ and so the min-cut max flow value from $\{v_2, v'_1\}$ to u_1 is 2. $P_{u'_1, u_1}^{G \setminus e_2}$ is edge disjoint from both $P_{s_1, u'_1}^{G \setminus e'_2}$ and $P_{s_2, u'_1}^{G \setminus e'_2}$ because e'_1 is a critical 1-edge cut for u_1 in $G \setminus e_2$. If either of $P_{s'_1, u'_1}^{G \setminus e_2}$ or $P_{s_2, u'_1}^{G \setminus e'_2}$ meet with $P_{u_2^*, u_1}^{G \setminus e_2}$, then this means that $P_{u_2^*, u_1}^{G \setminus e_2}$ and $P_{u'_1, u_1}^{G \setminus e_2}$ meet because e'_1 is a critical 1-edge cut for u_1 in $G \setminus e_2$ which is impossible. We can construct the \mathcal{P} and \mathcal{Q} satisfying condition 2 of Theorem 1.3.1 by letting $P_{s_1, d_1} = P_{s_1, u'_1}^{G \setminus e'_2} P_{u'_1, u_1}^{G \setminus e_2} P_{u_1, d_1}^{G \setminus e_2}$, $Q_{s_1, d_1} = P_{s_1, u_2}^{G \setminus e_1} P_{u_2, u_2^*}^{G \setminus e_1} P_{u_2^*, u_1}^{G \setminus e_2} P_{u_1, d_1}^{G \setminus e_2}$, $P_{s_2, d_2} = Q_{s_2, d_2} = P_{s_2, u_2}^{G \setminus e_1} P_{u_2, d_2}^{G \setminus e_1}$, $P_{s_2, d_1} = P_{s_2, u'_1}^{G \setminus e'_2} P_{u'_1, u_1}^{G \setminus e_2} P_{u_1, d_1}^{G \setminus e_2}$, and $Q_{s_1, d_2} = P_{s_1, u_2}^{G \setminus e_1} P_{u_2, d_2}^{G \setminus e_1}$.

Case 3: \nexists 2EDP $P_{s_1, u_2}^{G \setminus e_1}, P_{s_2, u_2}^{G \setminus e_1}$ in $G \setminus e_1$, \nexists 2EDP $P_{s_1, u_1}^{G \setminus e_2}, P_{s_2, u_1}^{G \setminus e_2}$ in $G \setminus e_2$, and \exists 2EDP $P_{s_1, u'_1}^{G \setminus e'_2}, P_{s_2, u'_1}^{G \setminus e'_2}$ in $G \setminus e'_2$, \exists 2EDP $P_{s_1, u'_2}^{G \setminus e'_1}, P_{s_2, u'_2}^{G \setminus e'_1}$ in $G \setminus e'_1$, where $e'_1 = (u'_1, v'_1)$ ($e'_2 = (u'_2, v'_2)$) is the critical 1-edge cut for u_1 (u_2).

The first set contains the following five path segments $P_{s_1, u'_1}^{G \setminus e'_2}, P_{s_2, u'_1}^{G \setminus e'_2}, P_{u'_1, u_1}^{G \setminus e'_2}, P_{u_1, d_1}^{G \setminus e'_2}$ and $P_{u_2^*, u_1}$ and the second set contains the following five path segments $P_{s_2, u'_2}^{G \setminus e'_1}, P_{s_1, u'_2}^{G \setminus e'_1}, P_{u'_2, u_2}^{G \setminus e'_1}, P_{u_2, d_2}^{G \setminus e'_1}$ and $P_{u_1^*, u_2}$. Paths in every set are edge disjoint by similar statement to the above. We can construct the \mathcal{P} and \mathcal{Q} satisfying condition 2 of Theorem 1.3.1 by letting $P_{s_1, d_1} = P_{s_1, u'_1}^{G \setminus e'_1} P_{u'_1, u_2^*}^{G \setminus e'_1} P_{u_2^*, u_1}^{G \setminus e_1} P_{u_1, d_1}^{G \setminus e_2}$, $P_{s_2, d_2} = P_{s_2, u'_2}^{G \setminus e'_1} P_{u'_2, u_2}^{G \setminus e_1} P_{u_2, d_2}^{G \setminus e_1}$, $P_{s_2, d_1} = P_{s_2, u'_1}^{G \setminus e'_2} P_{u'_1, u_1}^{G \setminus e_2} P_{u_1, d_1}^{G \setminus e_2}$, $Q_{s_1, d_1} = P_{s_1, u'_1}^{G \setminus e'_2} P_{u'_1, u_1}^{G \setminus e_2} P_{u_1, d_1}^{G \setminus e_2}$, $Q_{s_2, d_2} = P_{s_1, u'_1}^{G \setminus e'_2} P_{u'_1, u_1^*}^{G \setminus e_2} P_{u_1^*, u_2}^{G \setminus e_1} P_{u_2, d_2}^{G \setminus e_1}$, $Q_{s_1, d_2} = P_{s_1, u'_2}^{G \setminus e'_1} P_{u'_2, u_2}^{G \setminus e_1} P_{u_2, d_2}^{G \setminus e_1}$.

Case 4 \nexists 2EDP $P_{s_1, u_2}^{G \setminus e_1}, P_{s_2, u_2}^{G \setminus e_1}$ in $G \setminus e_1$, \nexists 2EDP $P_{s_1, u_1}^{G \setminus e_2}, P_{s_2, u_1}^{G \setminus e_2}$ in $G \setminus e_2$, and \exists 2EDP $P_{s_1, u'_1}^{G \setminus e'_2}, P_{s_2, u'_1}^{G \setminus e'_2}$ in $G \setminus e'_2$, \nexists 2EDP $P_{s_1, u'_2}^{G \setminus e'_1}, P_{s_2, u'_2}^{G \setminus e'_1}$ in $G \setminus e'_1$, and u_2 is reachable from both s_1 and s_2 .

Let $e''_2 = (u''_2, v''_2)$ be the critical 1-edge cut for e'_2 . The first set contains the following five paths segments $P_{s_1, u'_1}^{G \setminus e'_2}, P_{s_2, u'_1}^{G \setminus e'_2}, P_{u'_1, u_1}^{G \setminus e'_2}, P_{u_1, d_1}^{G \setminus e'_2}, P_{u_2^*, u_1}$ that are disjoint by the same argument in Case 3. The other set contains the following seven path segments $P_{s_2, u''_2}^{G \setminus e''_1}, P_{s_1, u''_2}^{G \setminus e''_1}, P_{u''_2, u_2}, P_{u_1^{**}, u_2}, P_{u_1^*, u_2}, P_{u_2, u_1}$, and P_{u_1, d_1} , such that u_1^{**} is the last vertex where $P_{u_1^*, u_2}$ and P_{u_1, d_1} meet. These seven path segments are disjoint by similar argument to case 2. We can construct the \mathcal{P} and \mathcal{Q} satisfying condition 2 of Theorem 1.3.1 by letting $P_{s_1, d_1} = P_{s_1, u'_1}^{G \setminus e'_2} P_{u'_1, u_1}^{G \setminus e_2} P_{u_1, d_1}^{G \setminus e_2}$, $P_{s_2, d_2} = P_{s_2, u'_1}^{G \setminus e'_2} P_{u'_1, u_1^{**}}^{G \setminus e_2} P_{u_1^{**}, u_2}^{G \setminus e_1} P_{u_2, u_2}^{G \setminus e_1}$, $P_{s_2, d_1} = P_{s_2, u'_1}^{G \setminus e'_2} P_{u'_1, u_2^*}^{G \setminus e_2} P_{u_2^*, u_1}^{G \setminus e_1} P_{u_1, d_1}^{G \setminus e_2}$, $Q_{s_1, d_1} = P_{s_1, u'_1}^{G \setminus e'_2} P_{u'_1, u_1}^{G \setminus e_2} P_{u_1, d_1}^{G \setminus e_2}$, $Q_{s_2, d_2} = P_{s_2, u'_1}^{G \setminus e'_2} P_{u'_1, u_1^*}^{G \setminus e_2} P_{u_1^*, u_2}^{G \setminus e_1} P_{u_2, d_2}^{G \setminus e_1}$, $Q_{s_1, d_2} = P_{s_1, u'_2}^{G \setminus e'_1} P_{u'_2, u_2}^{G \setminus e_1} P_{u_2, d_2}^{G \setminus e_1}$.

Case 5: The Network contains the network in Figure 4.1(d). In this case, if we remove all edges except those for the network in Figure 4.1(d), it is obvious that the proposition follows without considering the two sets. ■

4.3 Problem Formulation

For the following, we model the wireless multi-hop network by a directed hypergraph $G = (V, E)$, which is equivalent to the auxiliary node-based framework in Section 3.3.1¹, where V is the set of vertices that models the nodes and E is the set of hyperedges that models the links and broadcast links (as in Section 3.3.1). For the rest of the chapter we will use the term link to refer to both broadcast and non-broadcast links unless otherwise specified. We use $I(v)$ and $O(v)$ to represent the sets of incoming and outgoing links of node v , respectively. Their union is denoted as $N(v)$.

We use X_v^d to represent the long-term average data rate supported from node v to node d . Nodes v and d here form a unicast session where a long term average data rate X_v^d is supported. The “type” of this session is defined by the destination node d . For any two pairs of nodes (u_1, v_1) and (u_2, v_2) , as in Figure 4.2, a PICC can be defined as the six paths that satisfy the necessary and sufficient conditions for PINC in Theorem 1.3.1 that can be used to transmit packets of unicast session d_1 from u_1 to v_1 and packets of unicast session d_2 from u_2 to v_2 . In Figure 4.2 we term u_1 and v_1 the entry and exit nodes of the PICC for session d_1 , respectively. Similarly, u_2 and v_2 are the entry and exit nodes of the PICC for session d_2 . In this chapter, each PICC is thus described and differentiated by two parameters: the subgraph structure and the index pairs of the two sessions that are mixed together. For example, there are four distinct PICCs in Figure 4.3. All of the PICCs have the same subgraph structure but have different index pairs: (d_1, d_2) , (d_1, d_4) , (d_3, d_2) , (d_3, d_4) , where the first coordinate corresponds to the session using (u_1, v_1) as the entry and exit nodes. Similarly, the second coordinate corresponds to the second session using (u_2, v_2) as the entry and exit nodes.

¹The auxiliary node-model allows us to directly apply Theorem 1.3.1 to the wireline counterpart of a wireless network. On the other hand the hypergraph framework facilitates the use of one hyperedge for one physical transmission.

We define \mathcal{PICC} as the collection of all distinct PICCs, and $\mathcal{PICC}(d)$ as those PICCs with one of the session indices being d . For example: $\mathcal{PICC}(d_1) \cap \mathcal{PICC}(d_2)$ denote those PICCs with the two session indices being d_1 and d_2 . Suppose the k -th PICC is mixing two sessions d_1 and d_2 , we use $\Gamma_+(k, d_1)$ to denote the entry node of the k -th PICC for session d_1 . Similarly, $\Gamma_-(k, d_2)$ denotes the exit node of the k -th PICC for session d_2 . In this chapter, the alphabet k is reserved as the index of the PICCs in \mathcal{PICC} .

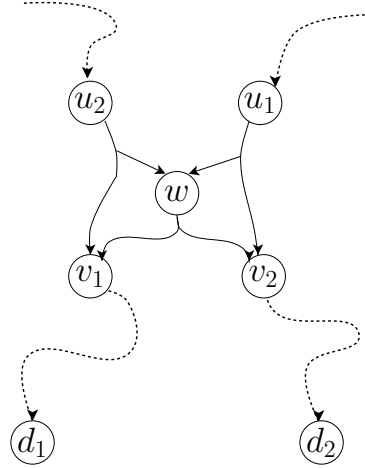


Fig. 4.2. The X-topology identified by the solid links with entry and exit nodes for session d_1 (d_2) being u_1, v_1 (u_2, v_2). The dotted lines represent the logical paths the information is flowing through. The logical paths might be non-coded paths or through PICCs

For the resource allocation problem of interest, each edge e has two types of rate assignments: x_e^d represents the rate allocated for transmitting non-coded packets, and x_e^k represents the rate allocated for transmitting coded packets over the k -th PICC in \mathcal{PICC} (totally there are $|\{d\}| + |\mathcal{PICC}|$ such x 's). We also use \vec{x} to denote the vector containing all of x_e^d and x_e^k , and use \vec{x}_e to represent the vector containing all of

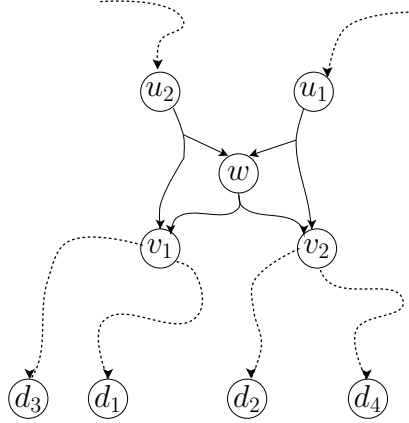


Fig. 4.3. Four PICCs are using the X-topology. The four sets of coexisting sessions for these PICCs are (d_1, d_2) , (d_1, d_4) , (d_3, d_2) , (d_3, d_4) . The dotted lines represent the logical paths the information is flowing through. The logical paths might be non-coded paths or through PICCs

the x variables for link e . The following constraints represent a new PINC capacity region.

$$\begin{aligned}
 & x_e^d + \sum_{k \in \mathcal{P}ICC(d)} \frac{x_e^k}{|O_k(v)|} \mathbf{1}_{\{e \in O_k(v), v = \Gamma_+(k, d)\}} \\
 & - \sum_{e \in I(v)} x_e^d + \sum_{k \in \mathcal{P}ICC(d)} \frac{x_e^k}{|I_k(v)|} \mathbf{1}_{\{e \in I_k(v), v = \Gamma_-(k, d)\}} - X_v^d \geq 0, \forall v, \forall d, \quad (4.1)
 \end{aligned}$$

$$x_{e_1}^k - x_{e_2}^k \geq 0, \forall k, \forall e_1 \in E, e_2 \in I_k(e_1), \quad (4.2)$$

$$x_{e_1}^k = x_{e_2}^k, \forall k, \forall v \in V, \forall e_1, e_2 \in I_k(v), \quad (4.3)$$

$$x_{e_3}^k = x_{e_4}^k, \forall k, \forall v \in V, \forall e_3, e_4 \in O_k(v), \quad (4.4)$$

$$\text{The scheduling constraints are satisfied.} \quad (4.5)$$

Here, $I_k(v)$ ($O_k(v)$) represents the set of incoming (outgoing) links to (from) node v that the k -th PICC uses. Similarly, $O_k(e)$ ($I_k(e)$) represents the set of next (previous) hop links from (to) link e in the k -th PICC.

Equations (4.1)–(4.5) represent the constraints for the routing and data link layers. Constraint (4.1) regards the PICC as a black box. It requires that at every node

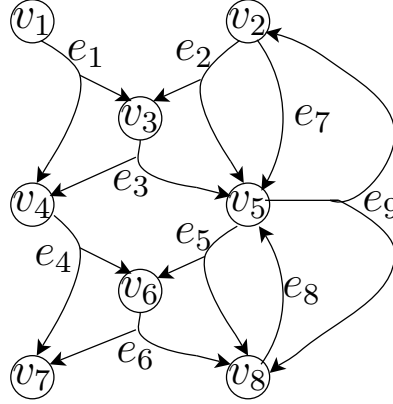


Fig. 4.4. Topology used to represent the linear formulation

v , the total outgoing rate for non-coded traffic of type d plus the rate of the coded traffic through all PICCs (where node v is the entry node for session d) should be able to support the locally generated traffic X_v^d , the total incoming rate for non-coded traffic of session d , and the rate through all PICCs where node v is the exit node for session d . The division by $|I_k(v)|$ and $|O_k(v)|$ in (4.1) becomes clear when we discuss the constraints in (4.3) and (4.4). The constraint in (4.2) handles the traffic inside the PICC. It requires that for a node that is neither an entry nor an exit node in a PICC the outgoing flow for that PICC from that node through any link should be able to support the incoming flows for that PICC through any link to that node. The constraint in (4.3) ((4.4)) requires that if node v has more than one incoming (outgoing) link in a PICC, the data rate for that PICC through these links should be the same. For a PICC with $|O_k(v)| > 1$ it means that each of the $|O_k(v)|$ links is supporting data rate x_e^k . Therefore, the total occupied capacity is $\sum_{k \in \mathcal{P}ICC(d)} x_e^k \mathbf{1}_{\{e \in O_k(v), v = \Gamma_+(k, d)\}}$ while the net information rate is $\sum_{k \in \mathcal{P}ICC(d)} \frac{x_e^k}{|O_k(v)|} \mathbf{1}_{\{e \in O_k(v), v = \Gamma_+(k, d)\}}$, which explains the division by $|I_k(v)|$ and $|O_k(v)|$ in (4.1).

In Figure 4.4, node v_5 is the entry node for the bottom X-topology PICC formed by links e_4, e_5, e_6 for session d , the exit node in the top X-topology PICC formed by links e_1, e_2, e_3 for session d , and an intermediate node for the third PICC (involving

v_2 , v_5 , and v_8) that exchange packets among nodes v_2 and v_8 and uses links e_7, e_8, e_9 . Constraint 4.1 requires that the total outgoing non-coded traffic of type d from node v_5 and the total outgoing traffic of type d using the bottom X-topology should be no less than the total generated traffic of type d at node v_5 , the incoming non-coded traffic of type d and the decoded traffic of type d from the top X-topology at node v_5 . Constraint 4.2 requires that the outgoing rate at node v_5 for the third PICC which is through the broadcast link e_9 should be no less than the input rates for that PICC at that node which are from links e_7 and e_8 . Constraint 4.3 ensures that the coded data rates for the third PICC through links e_7 and e_8 are the same.

Consider the wireless network in Fig. 4.5, where each source s_i wants to send one packet to its corresponding sink t_i . Using the approach in [26] the total number of transmissions cannot be less than twelve. The approach in [101] reduces the total number of transmissions to ten, but more than two hundred PICCs need to be examined. Our approach in this chapter reduces the number of transmissions to nine and the number of examined PICCs to 36. The reduction in the number of transmissions is due to allowing decoded packets to be reencoded again and the reduction in the number of examined PICCs is due to that the selected paths for forming PICCs in our approach have limited number of hops, while the approach in [101] uses all the end to end paths from the sources to the sinks which are so many. Another example is in Fig. 4.6 where there are three sessions. The sources and sinks for these three sessions are in the same figure. To send one packet through each session, we need eight transmissions using the approach in [101] and seven transmissions using both our approach and the approach in [26]. The above two examples show the superiority of our approach in terms of both throughput and complexity. In the following we will use the above constraints to formulate both the energy minimization and the rate control problems.

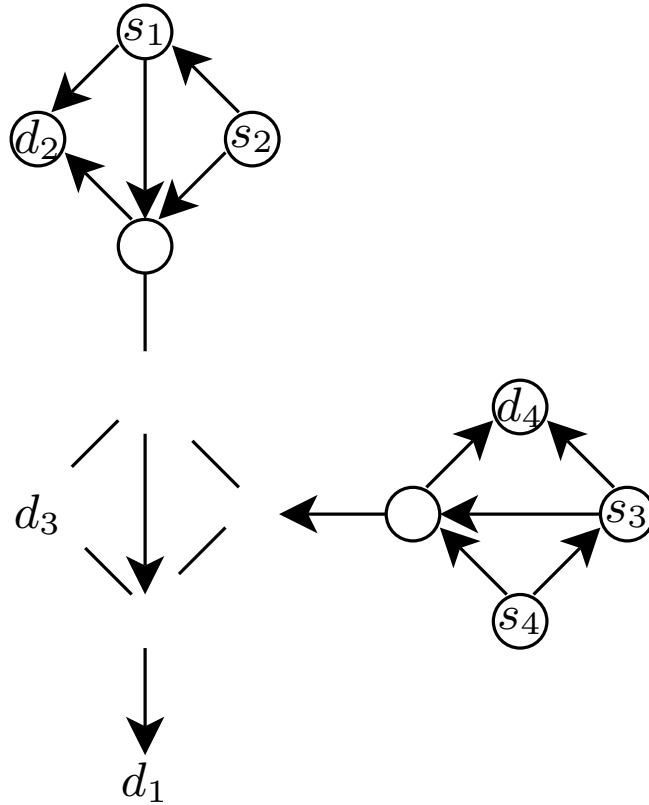


Fig. 4.5. A topology used to compare different capacity regions

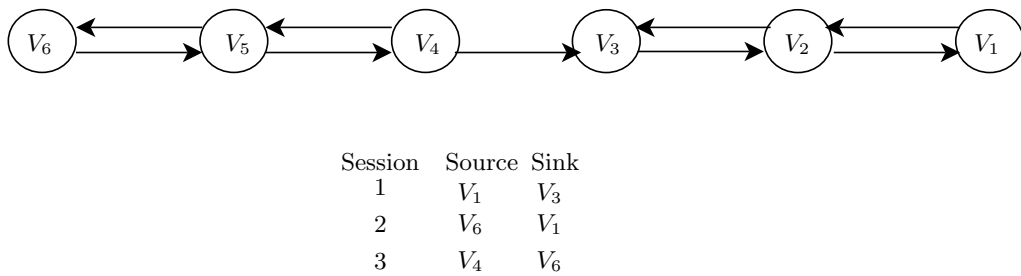


Fig. 4.6. A topology used to compare different capacity regions

4.4 Energy minimization

4.4.1 Distributed Algorithm

We assume that the power assignment at link e during time slot t is $h(R_e[t])$, where $R_e[t]$ is the data rate at link e during time slot t and $h(\cdot)$ is a non-decreasing convex

function satisfying $h(0) = 0$. For example $h(\cdot)$ can be the power-rate relationship in an Additive White Gaussian Noise (AWGN) channel. Let f_e represent the fraction of time slots for which link e is activated, we have,

$$f_e = \frac{\sum_d x_e^d + \sum_{k \in \mathcal{P}ICC} x_e^k}{R_e}, \forall e \in E,$$

which is due to the following lemma from [60].

Lemma 4.4.1 *For all time-slots t when link e is activated, the instantaneous data rate $R_e[t]$ is independent of t in any power-optimal scheme.*

Using a similar approach to that in [61], the optimization problem becomes:

$$\min_{\vec{x}, \vec{f}} \varphi_e(\vec{x}_e, f_e), \quad (4.6)$$

subject to: (4.1)-(4.5) and

$$(\vec{x}, \vec{f}) \in \mathcal{X},$$

where

$$\varphi_e(\vec{x}_e, f_e) = \begin{cases} 0, & f_e = 0 \\ f_e h\left(\frac{\sum_d x_e^d + \sum_{k \in \mathcal{P}ICC} x_e^k}{f_e}\right) & f_e > 0, \end{cases} \quad (4.7)$$

and

$$\mathcal{X} = \left\{ (\vec{x}, \vec{f}) : 0 \leq f_e \leq 1, \forall e, 0 \leq x_e^d \leq x_{\max}, \forall e, \forall d, \quad 0 \leq x_e^k \leq x_{\max}, \forall e, \forall k, \right. \\ \left. \sum_d x_e^d + \sum_{k \in \mathcal{P}ICC} x_e^k \leq R_{\max} f_e, \forall e \right\},$$

where x_{\max} and R_{\max} are the allowable maximum rates of a practical system. The constraint $\sum_d x_e^d + \sum_{k \in \mathcal{P}ICC} x_e^k \leq R_{\max} f_e, \forall e$ is to ensure convergence, otherwise \vec{x} can be set to a large number and \vec{f} to zero. The objective function (4.6) is convex over the set \mathcal{X} , (see [61] for details).

What remains to be specified is the scheduling constraint (4.5). If we use a layered approach, then (4.5) can be replaced by

$$f_e < \vartheta, \quad \forall v \in V, \quad (4.8)$$

$$e \in N(v)$$

where ϑ is a constant $\in [0, 1]$. Using Proposition 4.2.1, it can be shown that with the node exclusive model and PINC a hypergraph maximal matching scheduling algorithm can stabilize the system with any value of $\vartheta < \frac{1}{3}$.

If we assume the class of K -hop interference model as defined in [102], one can also use the cross-layer approach. In this interference model if a link is scheduled all other links that are within K -hop distance can not be scheduled simultaneously. When $K = 1$, the model reduces to the node exclusive model and when $K = 2$ it reduces to the 802.11 interference model. Let Ψ be the set of schedules and ψ is an instance of Ψ . We define the following

$$f_e^\psi = \begin{cases} 1 & e \in \psi \\ 0 & \text{Otherwise,} \end{cases}$$

and

$$\Pi = \left\{ \vec{f} \mid \vec{f} = \sum_{\psi \in \Psi} c_\psi \vec{f}^\psi, c_\psi \geq 0, c_\psi \leq 1 \right\},$$

where \vec{f}^ψ is a vector containing all variables of the form $f_e^\psi, \forall e$ and c_ψ is the time sharing coefficient. Therefore, (4.5) can be replaced by

$$\vec{f} \in \Pi \tag{4.9}$$

We refer to (4.6), (4.1)–(4.4), and (4.8) as the layered energy minimization formulation and to (4.6), (4.1)–(4.4), and (4.9) as the cross-layer energy minimization formulation. In either of the two settings, we have a convex optimization problem. In the following we will work on the dual problem of the layered formulation.

Associate Lagrange multipliers q_v^d with each constraint in (4.1), q_{e_1, e_2}^k with each constraint in (4.2), $\lambda_{v, \text{in}}^k$ with each constraint in (4.3), $\lambda_{v, \text{out}}^k$ with each constraint in (4.4), and q_v with each constraint in (4.8). We assume that $e = (u, \mathbf{v})$, where \mathbf{v} is the set of receiving nodes at link e . If e is a non-broadcast link, then \mathbf{v} contains one node. On the other hand, if e is a broadcast link, \mathbf{v} contains two nodes. Note that by

Proposition 4.2.1, a broadcast link with $|\mathbf{v}| > 2$ can be ignored without affecting the performance. The Lagrange function is separable and can be written as the following.

$$L(\vec{x}, \vec{f}, \vec{\lambda}, \vec{q}) = \sum_{e \in E} B_e(\vec{x}_e, f_e, \vec{q}_e, \vec{\lambda}_e) + \sum_{v \in V} \sum_{d \in \mathbf{v}} q_v^d X_v^d - \sum_{v \in V} q_v \vartheta,$$

where,

$$B_e(\vec{x}_e, f_e, \vec{q}_e, \vec{\lambda}_e) = \varphi_e(\vec{x}_e, f_e) - \sum_{d \in \mathbf{v}} BL_r(d, e) x_e^d - \sum_{k \in \mathcal{P}ICC} BL_c(k, e) x_e^k + \sum_{v \in \mathbf{v}} (q_v + q_u) f_e, \quad (4.10)$$

where $\vec{\lambda}_e$ is a vector containing the lagrange multipliers $\lambda_{u,\text{out}}^k$ and $\lambda_{v,\text{in}}^k \forall v \in \mathbf{v}$ and \vec{q}_e is a vector containing the lagrange multipliers $q_u^d, \forall d, q_v^d, \forall d, \forall v \in \mathbf{v}, q_{e_1,e}^k, \forall k, \forall e_1 \in O_k(e), q_{e,e_2}^k, \forall k, \forall e_2 \in I_k(e), q_u, q_v, \forall v \in \mathbf{v}$. Here, the non-coded backlog

$$BL_r(d, e) = \sum_{v \in \mathbf{v}} (q_u^d - q_v^d),$$

and the coded backlog

$$\begin{aligned} BL_c(k, e) = & \left(\frac{1}{|O_k(u)|} \left(q_u^{d_1} \mathbf{1}_{\{e \in O_k(u), u = \Gamma_+(k, d_1)\}} + q_u^{d_2} \mathbf{1}_{\{e \in O_k(u), u = \Gamma_+(k, d_2)\}} \right) \right. \\ & - \sum_{v \in \mathbf{v}} \frac{1}{|I_k(v)|} \left(q_v^{d_1} \mathbf{1}_{\{e \in I_k(v), v = \Gamma_+(k, d_1)\}} + q_v^{d_2} \mathbf{1}_{\{e \in O_k(u), u = \Gamma_+(k, d_2)\}} \right) \\ & - \sum_{e_1 \in O_k(e)} q_{e_1,e}^k + \sum_{e_2 \in I_k(e)} q_{e,e_2}^k - \lambda_{u,\text{out}}^k \mathbf{1}_{\{e = O_k(u,1)\}} + \lambda_{u,\text{out}}^k \mathbf{1}_{\{e = O_k(u,2)\}} \\ & \left. - \sum_{v \in \mathbf{v}} \left(\lambda_{v,\text{in}}^k \mathbf{1}_{\{e = I_k(v,1)\}} - \lambda_{v,\text{in}}^k \mathbf{1}_{\{e = I_k(v,2)\}} \right) \right) \Bigg|_{\substack{(d_1, d_2): \\ d_1 > d_2 \\ k \in \mathcal{P}ICC(d_1) \\ k \in \mathcal{P}ICC(d_2)}} \quad (4.11) \end{aligned}$$

As implied by the proof of Proposition 4.2.1, the maximum number of outgoing or incoming edges for any node in a PICC is at most 2. Therefore, the output (input) links of node u corresponding to the k -th PICC are numbered and we use $O_k(u, 1)$ ($I_k(u, 1)$) to refer to the first link and $O_k(u, 2)$ ($I_k(u, 2)$) for the second link. If there is only one outgoing edge from v for the k -th PICC, then $\lambda_{v,\text{out}}^k$ is set to zero.

Figure 4.7 represents the queuing process used for PICCs. In the figure two symbols X and Y in the input links of v are to be coded together and the resulting symbol Z is to be sent through the output link of u . We maintain three queues at u . The first two queues are for the received packets at the input links, we call them the I queues. I queues grow up when new packets arrive in the input link they are assigned to. They shrink when the coded packet is sent through the output link. The third queue is for the coded symbol Z in the output link, we call it O queue. O queues grow up when two packets from both input links are received and coded together. They shrink when this coded packet is sent through the output link. The term $\sum_{e_2 \in I_k(e)} q_{e,e_2}^k$ in (4.11) can be interpreted as the average O queue lengths at node u , we will refer to this term by \hat{q}_u^k . The term $\sum_{e_1 \in O_k(e)} q_{e_1,e}^k$ can be interpreted as the average I queue lengths at node v we will refer to this term by \hat{q}_v^k . We have

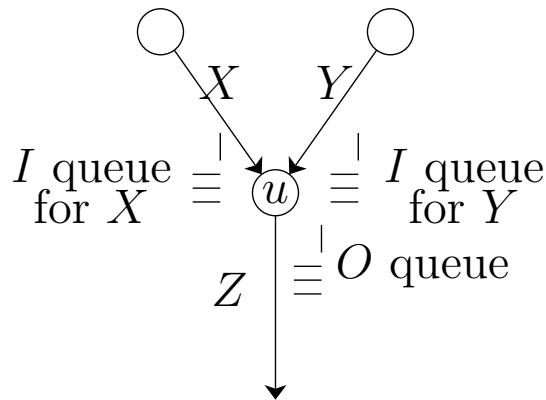


Fig. 4.7. Two I queues at node u for the received symbols X and Y in the input links and one O queue for the symbol Z sent through the output link. They grow up at possibly different times, but they shrink at the same time when a packet of the coded symbol Z is sent through the output link of u .

the following proposition.

Proposition 4.4.1 *There is no duality gap of the energy minimization problem. The optimal solution to the dual problem gives the optimal solution the primal problem.*

Proof The formulation satisfies Assumption 6.4.1 in [64]. Therefore by Proposition 6.4.2 in [64] the statement is correct. \blacksquare

The Lagrange function is separable and each link can maximize its own share of the Lagrange function. The dual objective function becomes

$$D(\vec{q}, \vec{\lambda}) = \min_{(\vec{x}, \vec{f}) \in \mathcal{X}} L(\vec{x}, \vec{f}, \vec{q}, \vec{\lambda}),$$

subject to $\vec{q} \geq \mathbf{0}$. The dual problem becomes:

$$\max_{\vec{q} \geq \mathbf{0}, \vec{\lambda}} D(\vec{q}, \vec{\lambda}),$$

We can rewrite $x_e^d[t] = f_e[t]R_e^d[t]$ and $x_e^k[t] = f_e[t]R_e^k[t]$, where $R_e^d[t]$ is the instantaneous rate for the non-coded traffic of session d through edge e at time slot t and $R_e^k[t]$ is the instantaneous rate for the k -th PICC through link e at time t . We have $B_e(\vec{x}_e[t], f_e[t]) = f_e[t]C_e(\vec{R}_e[t])$, where,

$$\begin{aligned} C_e(\vec{R}_e[t]) = & h \left(\sum_d R_e^d[t] + \sum_k R_e^k[t] \right) - \sum_d BL_r(d, e)R_e^d[t] \\ & + \sum_{k \in \text{PICC}} BL_c(k, e)R_e^k[t] + \left(\sum_{v \in \mathbf{v}} q_v[t] + q_u[t] \right). \end{aligned}$$

We then have the following Layered Energy Efficient Algorithm:

Algorithm LEEA

(1) **Initialization phase:** Each node finds all the possible PICCs in which it participates. The detailed PICC search algorithm will be explained in Section 4.6.

(2) Iterative phase

- **Finding the maximum backlog:** Each link minimizes C_e . The minimal value C_e^* is attained if the total energy is assigned to one of the R_e^d , or one of the R_e^k .

Each link finds

$$d^*(e) = \arg \max_d BL_r(d, e), \text{ and}$$

$$k^*(e) = \arg \max_k BL_c(k, e).$$

- **Selecting the power level:** Choose $R_e^{d^*(e)}$ and $R_e^{k^*(e)}$ that minimizes C_e , and denote the minimized value as C_e^* .

- **Considering the link for scheduling:** Set:

$$\hat{f}_e[t] = \begin{cases} 1 & \text{if } C_e^* < 0 \\ 0 & \text{otherwise,} \end{cases} \quad (4.12)$$

- **Link scheduling:** Perform a maximal hypergraph matching e.g., the GMHM algorithm on the links which have their $\hat{f}_e[t] = 1$. Note that $\hat{f}_e[t] = 1$ means that the link is a candidate to be scheduled. It does not mean that the link will be scheduled. It is shown in [61] that by choosing ϑ carefully, $\frac{1}{3}$ in our case, the number of time slots that link e has to wait until being scheduled is bounded. It is also shown that this will not affect the optimal solution.

- **Selecting the packet to send through the link:** Given that link e is scheduled, if

$BL_r(d^*(e), e) > BL_c(k^*(e), e)$ un-coded packets of type $d^*(e)$ is sent through e , otherwise perform pairwise intersession network coding on the $k^*(e)$ -th PICC through edge e . The distributed XOR-based coding scheme in [101] is used for coding.

Dual updates: Here, for simplicity we assume that the step size $\beta[t]$ is a constant β .

- Each node v update $q_v^d, \forall d$ according to the following,

$$q_v^d[t+1] = q_v^d[t] - \beta \left(\sum_{e \in O(v)} x_e^d[t] + \sum_{k \in \text{PICC}(d)} \frac{x_e^k[t]}{|O_k(v)|} 1_{\{e \in O_k(v), v = \Gamma_+(k, d)\}} \right) - \sum_{e \in I(v)} x_e^d[t] - \sum_{k \in \text{PICC}(d)} \frac{x_e^k[t]}{|I_k(v)|} 1_{\{e \in I_k(v), v = \Gamma_-(k, d)\}} - X_v^d \quad ,$$

where $()^+$ is a projection on $[0, \infty)$.

- At each node v and every two links e_1 and e_2 s.t $e_2 \in I_k(v)$ and $e_1 \in O_k(v)$, update q_{e_1, e_2}^k according to the following,

$$q_{e_1, e_2}^k[t+1] = q_{e_1, e_2}^k[t] - \beta \left(x_{e_1}^k[t] - x_{e_2}^k[t] \right) \quad .$$

- Each node v updates $\lambda_{v,\text{out}}^k, \forall k$ according to the following,

$$\lambda_{v,\text{out}}^k[t+1] = \lambda_{v,\text{out}}^k[t] + \beta \left(x_{e_1}^k[t] 1_{\{e_1=O_k(v,1)\}} - x_{e_2}^k[t] 1_{\{e_2=O_k(v,2)\}} \right) .$$

- Each node v updates $\lambda_{v,\text{in}}^k, \forall k$ according to the following,

$$\lambda_{v,\text{in}}^k[t+1] = \lambda_{v,\text{in}}^k[t] + \beta \left(x_{e_1}^k[t] 1_{\{e_1=I_k(v,1)\}} - x_{e_2}^k[t] 1_{\{e_2=I_k(v,2)\}} \right) .$$

- Each node v updates its own q_v according to the following,

$$q_v[t+1] = q_v[t] + \beta \sum_{e \in N(v)} f_e[t] - \vartheta .$$

The Cross-layer Energy Efficient Algorithm CEEA can be obtained by removing the term $(\sum_{v \in \mathbf{v}} q_v + q_u)$ from $C(\vec{R}_e)$ and performing the scheduling with the objective function being

$$\max_{\mathcal{M}} \sum_{e \in \mathcal{M}} -C_e^*[t],$$

where \mathcal{M} is a set of valid schedules.

Since $f_e[t]$ is set to either 0 or 1 under the K -hop interference model, there might be no single time slot where the algorithms converge to the optimal solution. The following proposition shows that the average consumed power over many time slots indeed converges to the optimal solution.

Proposition 4.4.2 (a) *Let Φ be the set of all $(\vec{q}, \vec{\lambda})$ that maximizes $D(\vec{q}, \vec{\lambda})$ for the layered energy minimization formulation. Then for any $\epsilon > 0, \exists \beta_0 > 0$, such that for any LEEA with a constant $\beta < \beta_0$ and ϑ small enough to guarantee stability we have*

a1) \exists a time T_0 such that $\forall t > T_0$ we have:

$$\min_{(\vec{q}^*, \vec{\lambda}^*) \in \Phi} \|(\vec{q}[t], \vec{\lambda}[t]) - (\vec{q}^*, \vec{\lambda}^*)\| \leq \epsilon.$$

a2)

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{e \in E} \wp_e(\vec{x}_e[t], f_e[t]) < \sum_{e \in E} \wp_e(\vec{x}_e^*, f_e^*) + \epsilon,$$

where (\vec{x}_e^*, f_e^*) is the optimal solution to the Layered energy formulation.

(b) Let Φ be the set of all $(\vec{q}, \vec{\lambda})$ that maximizes $D(\vec{q}, \vec{\lambda})$ for the cross layer energy minimization formulation. Then for any $\epsilon > 0$, $\exists \beta_0 > 0$, such that for any CEEA with perfect scheduling and a constant $\beta < \beta_0$ we have

b1) \exists a time T_1 such that $\forall t > T_1$, we have:

$$\min_{(\vec{q}^*, \vec{\lambda}^*) \in \Phi} \|\vec{q}[t], \vec{\lambda}[t] - (\vec{q}^*, \vec{\lambda}^*)\| \leq \epsilon.$$

b2)

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{e \in E} \wp_e(\vec{x}_e[t], f_e[t]) < \sum_{e \in E} \wp_e(\vec{x}_e^*, f_e^*) + \epsilon,$$

where (\vec{x}_e^*, f_e^*) is the optimal solution to the Cross-layer energy formulation.

Proof Due to the constraints $x_e^d \leq x_{\max}$, $x_e^k \leq x_{\max}$, and $f_e \leq 1$, the subgradient of the dual problem is bounded. Note that because $\wp_e(\vec{x}_e, f_e)$ is nondecreasing in \vec{x}_e , the optimal solution to the problem with $x_{\max} = |V|^2 \max_{v,d} X_v^d$ is the same as that without the constraints $x_e^d \leq x_{\max}$, $x_e^k \leq x_{\max}$.

Since the gradient of the dual problem is bounded a1 is guaranteed by Proposition 8.2.2 in [64] and the fact the the objective function in (4.6) is continuous. Note that statement a1 implies that the sequence $(\vec{q}[t], \vec{\lambda}[t])$ is bounded for all t which will be used to prove statement a2. Statement b1 follows by similar argument.

To show statement a2, we use the following Lyapunov function:

$$V[t] = \frac{1}{2} \sum_{v,d} (q_v^d[t])^2 + \sum_{v,k} \left\{ \sum_{\substack{e_1 \in I_k(v) \\ e_2 \in O_k(v)}} (q_{e_1, e_2}^k[t])^2 \right. \\ \left. + (\lambda_{v, \text{in}}^k[t])^2 + (\lambda_{v, \text{out}}^k[t])^2 \right\} + \sum_v (q_v[t])^2$$

The subgradients of $D(\vec{q}, \vec{\lambda})$ at time slot t can be written as:

$$\begin{aligned} \delta q_v^d[t] &\triangleq \frac{\partial}{\partial q_v^d} D(\vec{q}[t], \vec{\lambda}[t]) \\ &= - \sum_{e \in O(v)} x_e^d[t] + \sum_{k \in \text{PICC}(d)} \frac{x_e^k[t]}{|O_k(v)|} \mathbf{1}_{\{e \in O_k(v), v = \Gamma_+(k, d)\}} \\ &\quad - \sum_{e \in I(v)} x_e^d[t] + \sum_{k \in \text{PICC}(d)} \frac{x_e^k[t]}{|I_k(v)|} \mathbf{1}_{\{e \in I_k(v), v = \Gamma_-(k, d)\}} - X_v^d \ . \\ \delta q_{e_1, e_2}^k[t] &\triangleq \frac{\partial}{\partial q_{e_1, e_2}^k} D(\vec{q}[t], \vec{\lambda}[t]) = - x_{e_1}^k[t] - x_{e_2}^k[t] \ . \\ \delta \lambda_{v, \text{out}}^k[t] &\triangleq \frac{\partial}{\partial \lambda_{v, \text{out}}^k} D(\vec{q}[t], \vec{\lambda}[t]) = x_{e_1}^k[t] \mathbf{1}_{\{e_1 = O_k(v, 1)\}} - x_{e_2}^k[t] \mathbf{1}_{\{e_2 = O_k(v, 2)\}} \ . \\ \delta \lambda_{v, \text{in}}^k[t] &\triangleq \frac{\partial}{\partial \lambda_{v, \text{in}}^k} D(\vec{q}[t], \vec{\lambda}[t]) = x_{e_1}^k[t] \mathbf{1}_{\{e_1 = I_k(v, 1)\}} - x_{e_2}^k[t] \mathbf{1}_{\{e_2 = I_k(v, 2)\}} \ . \\ \delta q_v[t] &\triangleq \frac{\partial}{\partial q_v} D(\vec{q}[t], \vec{\lambda}[t]) = \sum_{e \in N(v)} f_e[t] - \vartheta \ . \end{aligned}$$

We have,

$$\begin{aligned}
& V[t+1] - V[t] \\
&= \frac{1}{2} \sum_{v,d} \left\{ (q_v^d[t] + \beta\delta q_v^d[t])^+ \right. \\
&\quad \left. - (q_v^d[t])^2 \right. \\
&+ \sum_{\substack{v,k \\ e_1 \in I_v(k) \\ e_2 \in O_k(v)}} (q_{e_1,e_2}^k[t] + \beta\delta q_{e_1,e_2}^k[t])^+ \\
&\quad \left. - (q_{e_1,e_2}^k[t])^2 \right. \\
&+ (\lambda_{v,\text{in}}^k[t] + \beta\delta \lambda_{v,\text{in}}^k[t])^+ \\
&\quad \left. - (\lambda_{v,\text{in}}^k[t])^2 \right. \\
&+ (\lambda_{v,\text{out}}^k[t] + \beta\delta \lambda_{v,\text{out}}^k[t])^+ \\
&\quad \left. - (\lambda_{v,\text{out}}^k[t])^2 \right. \\
&+ \sum_v (q_v[t] + \beta\delta q_v[t])^+ \\
&\quad \left. - (q_v[t])^2 \right.
\end{aligned}$$

$$\begin{aligned}
&\leq \frac{1}{2} \sum_{v,d} (q_v^d[t] + \beta \delta q_v^d[t])^2 - (q_v^d[t])^2 \\
&+ \sum_{v,k} \sum_{\substack{e_1 \in I_v(k) \\ e_2 \in O_k(v)}} (q_{e_1, e_2}^k[t] + \beta \delta q_{e_1, e_2}^k[t])^2 - (q_{e_1, e_2}^k[t])^2 \\
&+ \sum_v (\lambda_{v, \text{in}}^k[t] + \beta \delta \lambda_{v, \text{in}}^k[t])^2 - (\lambda_{v, \text{in}}^k[t])^2 \\
&+ \sum_v (\lambda_{v, \text{out}}^k[t] + \beta \delta \lambda_{v, \text{out}}^k[t])^2 - (\lambda_{v, \text{out}}^k[t])^2 \\
&+ \sum_v (q_v[t] + \beta \delta q_v[t])^2 - (q_v[t])^2 \\
&\leq \beta \sum_{v,d} (q_v^d[t] \delta q_v^d[t]) \\
&+ \sum_{v,k} \sum_{\substack{e_1 \in I_v(k) \\ e_2 \in O_k(v)}} (q_{e_1, e_2}^k[t] \delta q_{e_1, e_2}^k[t]) \\
&+ (\lambda_{v, \text{in}}^k[t] \delta \lambda_{v, \text{in}}^k[t]) + (\lambda_{v, \text{out}}^k[t] \delta \lambda_{v, \text{out}}^k[t]) \\
&+ \sum_v q_v[t] \delta q_v[t] + \beta^2 J
\end{aligned}$$

where J is a constant large enough to guarantee the inequality. Such a constant exists because both $(\vec{q}, \vec{\lambda})$ and the subgradient of the dual problem are bounded. Therefore, we have,

$$\begin{aligned}
&V[t+1] - V[t] + \beta \sum_{e \in E} \wp_e(\vec{x}_e[t], f_e[t]) \\
&\leq \beta D(\vec{q}[t], \vec{\lambda}[t]) + \beta^2 J \\
&\leq \beta \max_{\vec{q} \geq \mathbf{0}, \vec{\lambda}} (D(\vec{q}[t], \vec{\lambda}[t],)) + \beta^2 J \\
&= \beta \sum_{e \in E} \wp_e(\vec{x}_e^*, f_e^*) + \beta^2 J.
\end{aligned}$$

The last inequality is due to the strong duality theorem. Summing over $t = 1, \dots, T$ and dividing by T , we have,

$$\begin{aligned} & \frac{V(T+1) - V(1)}{T} + \frac{\beta}{T} \sum_{t=1}^{\infty} \sum_{e \in E} \wp_e(\vec{x}_e, f_e) \\ & \leq \beta \sum_{e \in E} \wp_e(\vec{x}_e^*, f_e^*) + \beta^2 J \end{aligned}$$

For a given ϵ , $\exists \beta_2$ s.t. $\beta J \leq \frac{\epsilon}{2}$, $\forall \beta \leq \beta_2$. Also, since the sequence $\{\vec{x}_e^t[t]\}$ is bounded, for any ϵ , $\exists T$ large enough s.t. $\frac{V(T+1) - V(1)}{\beta T} \leq \frac{\epsilon}{2}$. This implies that a2 follows. Statement b2 follows by similar argument. ■

4.4.2 Power Efficiency Ratio

Let Υ_e be the interference set of link e , i.e., the set of links that interfere with link e . Assume that the capacity of link e is r_e . It is shown in [55] that if

$$\sum_{e \in \Upsilon_e} \frac{\sum_d x_e^d + \sum_k x_e^k}{r_e} \leq 1, \forall e, \quad (4.13)$$

a maximal matching algorithm stabilizes the system. With PINC using Proposition 4.2.1, the maximum number of links in the interference set that can be scheduled simultaneously is 3. Hence, if $\sum_{e \in N(v)} f_e \leq \frac{1}{3} - \delta$, $\forall v$, $\forall \delta > 0$, then the condition in (4.13) is satisfied and the system is stable.

As discussed in Section 4.3 setting $\vartheta = 1$ gives a necessary conditions for scheduling. On the other hand if $\vartheta = \frac{1}{3} - \delta$, using a maximal hypergraph matching algorithm guarantees stability. Assume that (\vec{x}^*, \vec{f}^*) results in the optimal power assignment with perfect scheduling, it is obvious that $(\vec{x}^*, \frac{\vec{f}^*}{3+\epsilon})$ is a feasible solution to our problem. Using the first order approximation in [60] the consumed power in the additive white gaussian noise channel is proportional to $\sum_e \frac{1}{f_e}$. Therefore, the power consumption of our algorithm is at most $(3 + \epsilon)$ times the optimal power assignment for any $\epsilon > 0$.

4.5 Rate Control

Instead of the energy minimization problem considered in previous section, we can also consider the utility maximization problem. For this case, we can further extend the K -hop interference model to a general interference model. Let Ω be the set of feasible power assignments, we assume that $\vec{x} = u(\vec{p})$, i.e., the data rates are completely defined by the global power assignment. Let $\mathcal{R} = \{u(\vec{p}), \vec{p} \in \Omega\}$. We assume that $Co(\mathcal{R})$, the convex hull of \mathcal{R} is close and bounded. The rate control formulation becomes:

$$\max_{0 \leq \vec{X} \leq X_{\max}} U(X_v^d) \quad (4.14)$$

subject to: (4.1)-(4.4), and $\vec{x} \in Co(\mathcal{R})$. Here, U is a concave non-decreasing function and $\vec{X} = \{X_v^d : \forall v, d, v \neq d\}$.

Using the dual approach as in the energy minimization problem we have the following Cross-layer Rate Control Algorithm CRCA:

Source Algorithm. The source node v that wants to send packets to d solves the following problem.

$$X_v^d[t] = \arg \max_{0 \leq X_v^d \leq X_{\max}} U(X_v^d) - q_v^d[t] X_v^d \quad (4.15)$$

Link Algorithm. Link e chooses $d^*(e)$ and $k^*(e)$ as in the energy minimization problem and sets

$$w_e[t] = \max(BL_r(d^*(e), e), BL_c(k^*(e), e)).$$

Link scheduling. Perform link scheduling with the objective function being $\max_{\{\vec{R} \in u(\vec{p}), \vec{p} \in \Omega\}} \sum_e R_e w_e[t]$.

Selecting the packets to send through the link If

$BL_r(d^*(e), e) < BL_c(k^*(e), e)$, coded packets from the $k^*(e)$ -th PICC are sent through link e at rate $R_e[t]$. Otherwise un-coded packets of type $d^*(e)$ at rate $R_e[t]$ are sent.

Proposition 4.5.1 For CRCA with perfect scheduling, let Φ be the set of $(\vec{q}, \vec{\lambda})$ that minimizes $D(\vec{q}, \vec{\lambda})$, the dual objective function of the cross-layer rate control

formulation, and let \vec{X}^* be the unique optimal solution. If

$\vec{\beta}[t] \rightarrow 0$ as $t \rightarrow \infty$ and $\sum_t \beta[t] = \infty$ then,

$\min_{(\vec{q}^*, \vec{\lambda}^*) \in \Phi} (\vec{q}[t], \vec{\lambda}[t]) - (\vec{q}^*, \vec{\lambda}^*) \rightarrow 0$ and
 $\vec{X}[t] \rightarrow \vec{X}^*$ as $t \rightarrow \infty$

Proof The rate control formulation satisfies Assumption 6.4.1 in [64]. Therefore, there is no duality gap by Proposition 6.4.2 in [64]. By Proposition 8.2.5 in [64] we have

$$\min_{(\vec{q}^*, \vec{\lambda}^*) \in \Phi} (\vec{q}[t], \vec{\lambda}[t]) - (\vec{q}^*, \vec{\lambda}^*) \rightarrow 0$$

as $t \rightarrow \infty$. Also because there is no duality gap the primal variables converge to the optimal solution. What remains to be proven is that there is a single maximizer \vec{X}^* .

Let

$$\begin{aligned} g_v^d(\vec{X}, \vec{x}) = & - \sum_{e \in O(v)} x_e^d + \sum_{k \in \text{PICC}(d)} \frac{x_e^k}{|O_k(v)|} 1_{\{e \in O_k(v), v = \Gamma_+(k, d)\}} \\ & - \sum_{e \in I(v)} x_e^d + \sum_{k \in \text{PICC}(d)} \frac{x_e^k}{|I_k(v)|} 1_{\{e \in I_k(v), v = \Gamma_-(k, d)\}} - X_v^d, \\ g_{e_1, e_2}^k(\vec{x}) = & - x_{e_1}^k - x_{e_2}^k. \end{aligned}$$

$$h_{\text{out}, e_1, e_2}^k(\vec{x}) = x_{e_1}^k 1_{\{e_1 = O_k(v, 1)\}} - x_{e_2}^k 1_{\{e_2 = O_k(v, 2)\}}.$$

$$h_{\text{in}, e_1, e_2}^k(\vec{x}) = x_{e_1}^k 1_{\{e_1 = I_k(v, 1)\}} - x_{e_2}^k 1_{\{e_2 = I_k(v, 2)\}}.$$

Also let \vec{g} be a vector containing all g functions and \vec{h} be a vector containing all h functions. Therefore for any $(\vec{q}, \vec{\lambda}) \in \Phi$, we have:

$$\begin{aligned} U(X_v^{d*}) = \max_{X_v^d \leq X_{\max}^d} & U(X_v^d) - \vec{q} \vec{g}(\vec{X}, \vec{x}) - \vec{\lambda} \vec{h}(\vec{X}, \vec{x}) \\ \geq & U(X_v^{d*}) - \vec{q} \vec{g}(\vec{X}^*, \vec{x}^*) - \vec{\lambda} \vec{h}(\vec{X}^*, \vec{x}^*). \end{aligned}$$

Since $\vec{h}(\vec{X}^*, \vec{x}^*) = 0$, we should have $\vec{q}\vec{g}(\vec{X}^*, \vec{x}^*) \geq 0$. But $\vec{q}\vec{g}(\vec{X}^*, \vec{x}^*) \leq 0$, because \vec{X}^* is a maximizer. Therefore, we have $\vec{q}\vec{g}(\vec{X}^*, \vec{x}^*) = 0$ and the following equality follows,

$$\begin{aligned} & \max_{X_v^d \leq X_{\max}} \underset{v=d}{U(X_v^d) - \vec{q}\vec{g}(\vec{X}, \vec{x}) - \vec{\lambda}\vec{h}(\vec{X}, \vec{x})} \\ &= \underset{v=d}{U(X_v^{d*}) - \vec{q}\vec{g}(\vec{X}^*, \vec{x}^*) - \vec{\lambda}\vec{h}(\vec{X}^*, \vec{x}^*)}. \end{aligned}$$

However, given $(\vec{q}, \vec{\lambda})$, the point (\vec{X}, \vec{x}) that maximizes $L(\vec{X}, \vec{x}, \vec{q}, \vec{\lambda})$ should be the optimal solution to (4.15). Since $U(\cdot)$ is strictly concave, the optimal solution is unique. ■

4.6 Finding PICCs

For the developed algorithms to run properly, every node should know all PICCs in which it itself participates. In this section we provide a way to find all the PICCs in the network such that every one of the six paths of the PICC uses at most κ hops. Here, κ is an arbitrary integer. As κ increases, more PICCs are considered which results in a better solution. On the other hand increasing κ increases the complexity of the algorithm. The following steps describe how to find the PICCs distributively.

Phase 1: Finding the sets of candidate entry and exit nodes to form PICCs

1. Every node u finds all of the paths P_{uv} that use at most κ hops and start at u .
2. The destinations of these paths then form possible exit nodes for the κ -hop PICCs with node u being one of the entry nodes. Node u finds all possible pairs of exit nodes.
3. For every found path P_{uv} , that uses more than one hop, node u sends a control message containing all pairs of exit nodes found in the previous step such that v is an element in all of these pairs².

²Nodes u and v here are candidates for the entry and exit nodes for the same session in a PICC. It is inefficient for neighboring nodes to be the entry and exit nodes for the same session, because non-coded packets can be sent in one time slot. Therefore, we exclude one-hop paths in this step.

4. If two paths with different sources and different destinations that carry control messages meet at an intermediate node, the intermediate node searches for a match in the exit nodes pairs that the control messages carry. If a match is found a feedback message is sent back to the sources of both paths to set up PICCs with the entry nodes being the sources of the paths and the exit nodes being the nodes in the matched pair.

Phase 2: Checking the edge overlap conditions Assume that two node pairs (u_1, v_1) (u_2, v_2) are selected in Phase 1 as the entry and exit node pairs. The corresponding PICCs are formed according to the following steps.

1. Every node adds auxiliary nodes as in Section 3.3.1 to consider broadcast links of size 2.
2. Node u_1 (u_2) forms two sets of paths such that the first one contains all paths of the form P_{u_1, v_1} (P_{u_2, v_2}) and the second one contains all paths of the form P_{u_1, v_2} (P_{u_2, v_1}). Here, one path from the first set of u_1 , one path from the first set of u_2 , and one path from the second set of u_1 form the set \mathcal{P} in Theorem 1.3.1. Similarly, one path from the first set of u_1 , one path from the first set of u_2 , and one path from the second set of u_2 form the set \mathcal{Q} in Theorem 1.3.1.
3. Nodes u_1 and u_2 announce the sizes of the sets formed in the previous step to all of the nodes in the paths on these four sets. Every possible PICC will have a unique ID number. Since intermediate nodes know the sizes of the four sets in step 2 of this phase, the ID number can be computed locally.
4. Every link e knows all the paths that it is involved in. For every possible PICC at link e if the the controlled-edge overlap condition in Theorem 1.3.1 is not satisfied, a feedback message is sent back to both u_1 , u_2 to exclude that PICC³.

The energy minimization and the rate-control algorithms in Sections 4.4 and 4.5 can then be performed on the constructed PICCs.

³Our results consider both cyclic and acyclic PICCs which is guaranteed by Theorem 1.3.1. If we restrict our assumption to acyclic PICCs, redundant PICCs can be excluded in a similar way to [91].

4.7 Simulation Results

The purpose of our simulations is to verify the convergence of the algorithms and to show the benefit of the new approach over the non-coded solutions.

We first run simulations for both our CEEA and the non-coded version of the CEEA. We assume the node-exclusive model and use the following power rate function, $R_e = \log(1 + \frac{GP_e}{N_0W})$, where the link gain $G = 1.6 \times 10^{-13}$, the background noise $N_0 = 1.6 \times 10^{-18}$ mW/Hz, and the bandwidth $W = 1 \times 10^6$ MHz. For scheduling we use the Greedy Maximal Matching GMM algorithm for the non-coded algorithm and GMHM for the network coding one. For the network coding algorithm, we use $\kappa = 3$. We run simulations on the 3×3 grid topology in Figure 4.8 with four sessions shown in the same figure. Each sessions demand is $500kb/sec$. The convergence results for both the network coding and non-coded versions of the CEEA algorithm are in Figure 4.9. The non-coded solution consumes 26% more power than the network coding one.

For rate control we assume the linear signal-to-noise-&- interference-ratio (SINR) model. We assume that the total power assigned to node u at any time slot is bounded by $P_{u,max}$. In each time slot, each node u should either transmit at full power $P_{u,max}$ or remain silent in order to achieve the optimal scheduling solution. Without WMA, the data rate R_e at link $e = (u, v)$ is assumed to be proportional to the SINR level at the receiver v , which is formally expressed as.

$$R_e = W \frac{G(u, v) 1_{\{e \text{ is activated}\}} P_{u,max}}{N_0 + \sum_{w:w=u} G(w, v) 1_{\{\text{node } w \text{ is sending}\}} P_{w,max}},$$

where N_0 is the background noise, W is the bandwidth of the system, and $G(u, v)$ is the path gain between nodes u and v which is set to $(\text{dist}(u, v))^{-4}$, where $\text{dist}(u, v)$ is the Euclidean distance between nodes u and v . With PINC and WMA, the data rate of the broadcast link with two receivers is proportional to the minimum of the

SINR levels at those receivers. More precisely, for $e = (u, \{v_1, v_2\})$, the data rate of this broadcast link, R_e , becomes,

$$R_e = W \min \frac{G(u, v_1) 1_{\{e \text{ is activated}\}} P_{u, \max}}{N_0 + \sum_{w:w=u} G(w, v_1) 1_{\{w \text{ is sending}\}} P_{w, \max}},$$

$$\frac{G(u, v_2) 1_{\{e \text{ is activated}\}} P_{u, \max}}{N_0 + \sum_{w:w=u} G(w, v_2) 1_{\{w \text{ is sending}\}} P_{w, \max}}.$$

We use the topology in Figure 4.10 with two sessions (s_1, d_1) and (s_2, d_2) and $N_0 = 1$, $W = 10$, $P_{u, \max} = 10$. For the session between nodes v and d we use the utility function $U(X_v^d) = \ln(X_v^d)$. The locations of the nodes are shown in Figure 4.10. Figure 4.11 shows the rate assigned to the sessions at each iteration for both the coded and non-coded versions of the CRCA algorithm. An improvement of 35% is gained by using network coding.

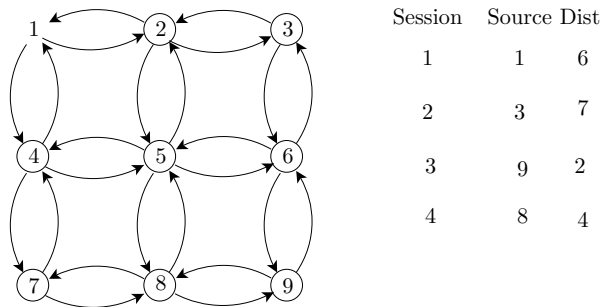


Fig. 4.8. A 3×3 grid topology used in the simulations.

4.8 Conclusion

In this chapter we present a framework for using pairwise intersession network coding in wireless networks. The framework is used to develop distributed algorithms for both the energy minimization and rate control problems. As is the case in the non-coded case, the most complex part of our algorithms is the scheduling part. While our formulation is suitable for a general class of interference models, we have used a

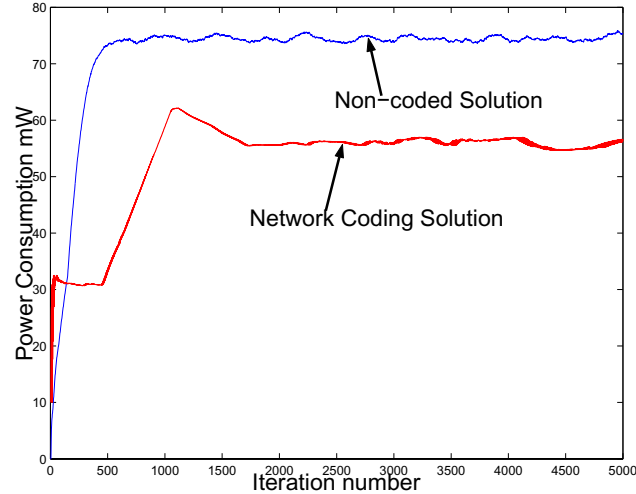


Fig. 4.9. Convergence for both the coded and non-coded versions of the CEEA Algorithm on a 3×3 grid topology.

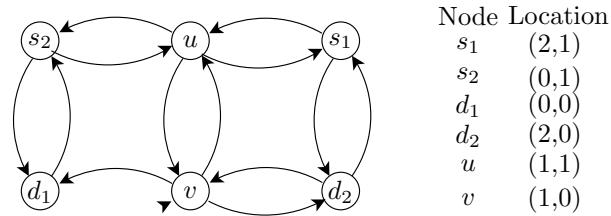


Fig. 4.10. The wireless butterfly topology used in the simulations

distributed algorithms for scheduling in the node exclusive interference model. The approach used in this chapter is a back-pressure one which allows decoded packets to be reencoded again. Since our approach uses local paths to form PICCs its complexity is less than that provided in Chapter 3.

The assumption made in this chapter and the previous one is that of opportunistic scheduling that assumes that the channel conditions are known prior to transmission, which might not be the case in most of practical wireless networking scenarios. In

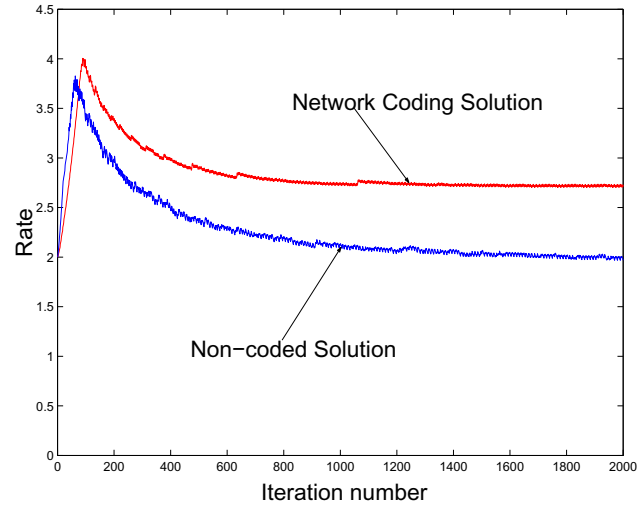


Fig. 4.11. Convergence for both the coded and non-coded versions of the CRCA Algorithm.

the next chapter we remove this assumption and develop the corresponding optimal coding schemes.

5. THE CAPACITY REGION OF 2-HOP RELAY NETWORK WITHOUT OPPORTUNISTIC SCHEDULING

5.1 Main Contribution

In wireless mesh networks it is hard to perform opportunistic scheduling which is to estimate the channel conditions prior to transmission. Therefore, the feedback that is obtained after the transmission plays a major role in designing optimal transmission schemes. Without intersession network coding and under the assumption of no opportunistic scheduling, the optimal transmission policy can be obtained using opportunistic routing [75], or intrasession network coding [28] when complete or limited feedback is allowed. To further increase the achievable rate, intersession network coding should be used. This chapter represents the first work that characterizes intersession network coding capacity when the channel conditions are unknown prior to transmission. Our main contributions in this chapter are as follows.

- It is hard to characterize the general capacity region of intersession network coding. Therefore, when we limit the encoding and decoding nodes to be neighbors also called as 2-hop relay network and under the no opportunistic scheduling assumption, we characterize the capacity region for two flows using linear programming. We also provide a coding scheme that uses random network coding and achieves the characterized capacity.
- Under the no opportunistic routing case and for 2-hop relay network, when only XOR operations are allowed, we characterize the capacity region for arbitrary number of sessions. The capacity region is represented by a linear program.

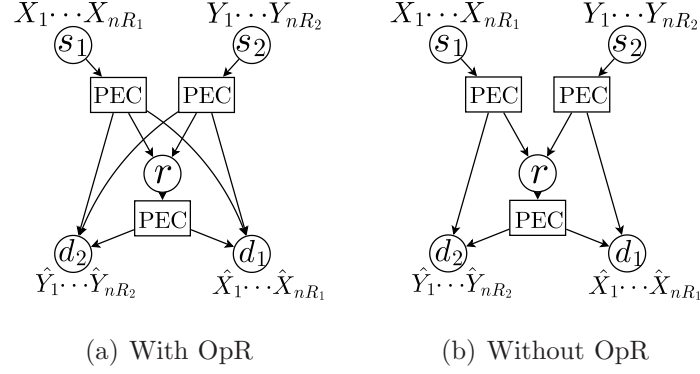


Fig. 5.1. Illustration of 2-session relay networks.

- We compare the improvement achieved by using three different techniques to maximize the achievable rate of wireless network. These are, Intersession Network Coding, Opportunistic Routing, and Cross-Layer Design.

5.2 The Capacity region of two hop coding with two flows

5.2.1 The Setting

We use X (and sometimes Y) to denote a symbol in $\mathbf{GF}(q)$. A 1-to- K packet erasure channel (PEC) takes an input $X \in \mathbf{GF}(q)$ and outputs a K -dimensional vector in $(\{X\} \cup \{*\})^K$, where the k -th coordinate being “*” denotes that the input symbol X is erased for the k -th receiver. We consider only i.i.d. PECs of which the erasure pattern is independent for each channel usage. For example, a 1-to-2 PEC can be described by four parameters $p_{s;12}$, $p_{s;12^c}$, $p_{s;1^c2}$, and $p_{s;1^c2^c}$, which denote the probabilities that X is received by both receivers (rx) 1 and 2, by only rx 1, by only rx 2, and by neither rx 1 nor 2. This notation can be easily extended to the marginal success probability $p_{s;1} \triangleq p_{s;12} + p_{s;12^c}$ and to the union $p_{s;1 \cup 2} \triangleq p_{s;12} + p_{s;12^c} + p_{s;1^c2}$.

A 2-session relay network is described in Fig. 5.1(a), in which source s_1 would like to send nR_1 packets X_1, \dots, X_{nR_1} to destination d_1 , and s_2 would like to send nR_2 packets Y_1, \dots, Y_{nR_2} to d_2 . r is a relay node. Each of s_1 , s_2 , and r can use

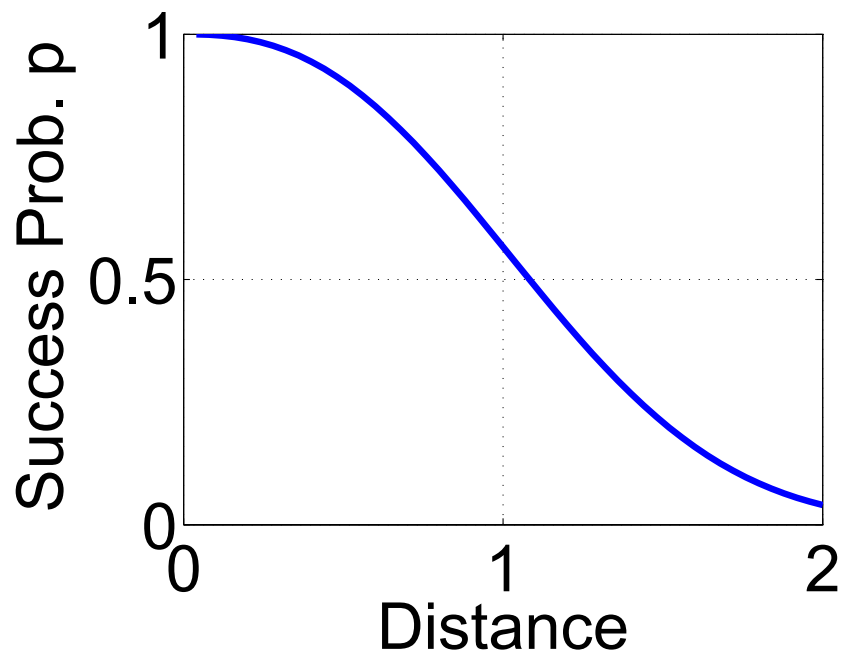


Fig. 5.2. The overhearing probability versus the distance.

the corresponding PEC n times, respectively, and we are interested in the largest achievable rate pair (R_1, R_2) that guarantees decodability at d_1 and d_2 with close-to-1 probability for sufficiently large n and the finite field size q .

To model the “reception report” suggested in COPE, we enforce the following sequential, round-based feedback schedule: Each of s_1 and s_2 transmits n symbols, respectively. After the transmission of $2n$ symbols, two reception reports are sent from d_1 and d_2 , respectively, back to the relay r so that r knows which packets have successfully arrived which destinations. After the reception reports, no further feedback is allowed and the relay r has to make its own decision how to use the available n PEC usages to guarantee decodability at d_1 and d_2 . In our setting, we also assume that the success probability parameters of all PECs and all the coding operations are known to all nodes. The only unknown part is the values of the X and Y symbols. For the purpose of illustration, a simplified network setting is also depicted in Fig. 5.1(b), in which the packets sent by s_i will not be overheard by the 2-hop-away destination d_i . In this simplified setting, the question thus becomes given the following parameters: $p_{s_1;rd_2}$, $p_{s_1;rd_2^c}$, $p_{s_1;rcd_2}$, and $p_{s_1;rcd_2^c}$; $p_{s_2;rd_1}$, $p_{s_2;rd_1^c}$, $p_{s_2;rcd_1}$, and $p_{s_2;rcd_1^c}$; $p_{r;d_1d_2}$, $p_{r;d_1d_2^c}$, $p_{r;d_1^cd_2}$, and $p_{r;d_1^cd_2^c}$, what is the maximum achievable rate under the round-based feedback model. For future reference, we say Fig. 5.1(a) admits OpR as the packets can be overheard by the two-hop destinations while Fig. 5.1(b) does not admit OpR. Since the settings are symmetric, we sometimes assume that $p_{r;d_1} \geq p_{r;d_2}$ which can be achieved by relabelling the sessions.

5.2.2 The Capacity Results

Consider Fig. 5.1(b) and the scenario in which s_1 and s_2 have finished transmission and the reception reports have been sent to r . The question now becomes a broadcast PEC problem with side information (SI) as depicted in Fig. 5.3. That is, the packets $X_1^{[2]}, \dots, X_{nR_{1;2}}^{[2]}$ have been overheard by d_2 and the packets $X_1^{[2^c]}, \dots, X_{nR_{1;2^c}}^{[2^c]}$ have not. Similarly, $Y_1^{[1]}, \dots, Y_{nR_{2;1}}^{[1]}$ have been overheard by d_1 and $Y_1^{[1^c]}, \dots, Y_{nR_{2;1^c}}^{[1^c]}$ have

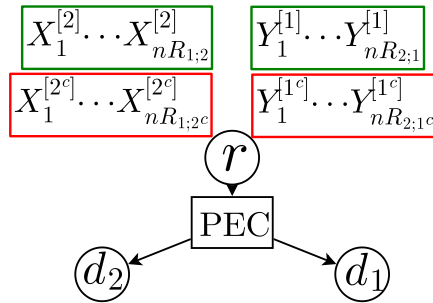


Fig. 5.3. A broadcast PEC problem with side information.

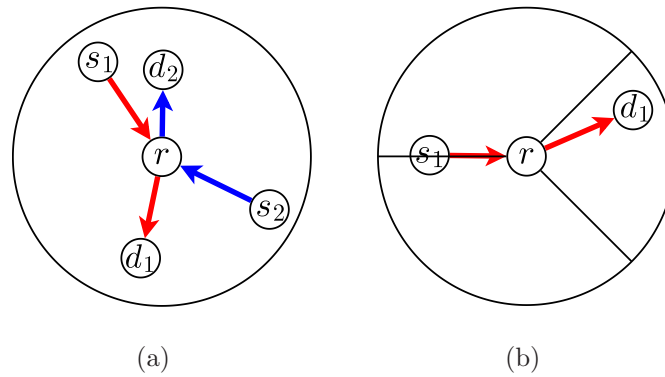


Fig. 5.4. (a) An example of random node placement for $M = 2$ sessions. (b) The constraint on the topological relationship between s_i and d_i .

not. $X^{[2]}$ and $Y^{[1]}$ packets can later serve as SI when decoding at d_2 and d_1 , respectively. The total rates for each session are $R_1 = R_{1;2} + R_{1;2^c}$ and $R_2 = R_{2;1} + R_{2;1^c}$. We then have the following proposition:

Proposition 5.2.1 *For any rate vectors $(R_{1;2}, R_{1;2^c}, R_{2;1}, R_{2;1^c})$ for the broadcast PEC with SI in Fig. 5.3 and assuming that $p_{r;d_1} \geq p_{r;d_2}$, one can communicate the values of all X and Y packets to d_1 and d_2 , respectively, within n channel usage if and only if $(R_{1;2}, R_{1;2^c}, R_{2;1}, R_{2;1^c})$ satisfies*

$$R_{1;2} + R_{1;2^c} + R_{2;1^c} \leq p_{r;d_1} \quad (5.1)$$

$$R_{2;1} + R_{2;1^c} + \frac{p_{r;d_2}}{p_{r;d_1}} R_{1;2^c} \leq p_{r;d_2}. \quad (5.2)$$

Remark: The broadcast capacity of *Gaussian channel* with SI has been considered in many papers (see [103] as a representative work). In addition to the difference of the settings of PECs and Gaussian channels, this chapter will also consider the best coding strategy at sources s_1 and s_2 and characterize the capacity as a linear programming problem. It is worth noting that for 3-session Gaussian broadcast channels with general SI, the capacity remains an open problem. On the other hand, in [104] we show that the capacity of 3-session broadcast PEC can be computed in a similar way as outlined in this paper. Recently, the capacity of 2-session broadcast PECs with instant packet-by-packet feedback (but without SI) is studied in [105].

Sketch of the proof of Proposition 5.2.1:

Achievability: For any vector $(R_{1;2}, R_{1;2^c}, R_{2;1}, R_{2;1^c})$ satisfying (5.1) and (5.2), perform the following 2-staged coding scheme sequentially.

Stage 1: Whenever we can use the broadcast PEC, we mix the packets of the three groups: $X_1^{[2]}$ to $X_{nR_{1;2}}^{[2]}$, $Y_1^{[1]}$ to $Y_{nR_{2;1}}^{[1]}$, and $Y_1^{[1^c]}$ to $Y_{nR_{2;1^c}}^{[1^c]}$ by random linear network coding (RLNC) [106] and generate one outgoing symbol. Repeat this random packet generation until we have sent out the following amount of randomly generated packets:

$$\max \left(\frac{nR_{1;2} + nR_{2;1^c}}{p_{r;d_1}}, \frac{nR_{2;1} + nR_{2;1^c}}{p_{r;d_2}} \right). \quad (5.3)$$

Stage 2: Whenever we can use the broadcast PEC, we mix the packets of $X_1^{[2^c]}$ to $X_{nR_{1;2^c}}^{[2^c]}$ by RLNC and generate one outgoing symbol. Repeat this random packet generation until we have sent out the following amount of packets:

$$\frac{nR_{1;2^c}}{p_{r;d_1}}. \quad (5.4)$$

The decodability proof is straightforward as from rx 1's perspective, d_1 has received a sufficiently large amount of coded packets to decode the information packets in the three groups: $X^{[2]}$, $X^{[2^c]}$, and $Y^{[1^c]}$ (since d_1 has already heard the $Y^{[1]}$ packets). Similarly, d_2 simply ignores the packets transmitted in Stage 2 and use the packets in Stage 1 to decode the desired two groups of packets $Y^{[1^c]}$ and $Y^{[1]}$. It remains to show that one can finish Stages 1 and 2 within the allowed n channel usages, which is equivalent to proving that (5.3) plus (5.4) is no larger than n . By noticing that (5.3)+(5.4) $\leq n$ is satisfied for any $(R_{1;2}, R_{1;2^c}, R_{2;1}, R_{2;1^c})$ vector satisfying (5.1) and (5.2), the proof of the achievability is thus complete.

The converse part can be proven by deriving the lower bound on the amount of interference that session 1 will impose on destination d_2 . (Stage 2 of the achievability scheme carefully limits the amount of interference to the minimal possible amount, $\frac{p_{r;d_2}}{p_{r;d_1}}R_{1;2^c}$, one can hope to achieve.)

Proposition 5.2.1 can be used to derive the 2-session relay network capacity by noting that if we let each of s_1 and s_2 send n packets by RLNC, we can maximize the overheard SI at d_2 and d_1 , respectively. Based on this observation, the capacity of a 2-session relay network becomes:

Proposition 5.2.2 *Assuming that $p_{r;d_1} \geq p_{r;d_2}$, for a 2-session relay network without OpR (Fig. 5.1(b)), the rate pair (R_1, R_2) is achievable if and only if they satisfy*

$$\begin{aligned} R_1 &\leq \min(p_{s_1;r}, p_{r;d_1} - (R_2 - p_{s_2;d_1})^+) \\ R_2 &\leq \min\left(p_{s_2;r}, p_{r;d_2} - \frac{p_{r;d_2}}{p_{r;d_1}}(R_1 - p_{s_1;d_2})^+\right), \end{aligned}$$

where $(\cdot)^+$ is the projection to a non-negative number.

In the following, we generalize Proposition 5.2.2 for OpR (Fig. 5.1(a)) and for the case in which s_1 , s_2 , and r have different amount of available time slots: nt_{s_1} , nt_{s_2} , and nt_r , respectively.

Proposition 5.2.3 *Assuming that $p_{r;d_1} \geq p_{r;d_2}$ and variable transmission time nt_{s_1} , nt_{s_2} , and nt_r , for a 2-session relay network with OpR (Fig. 5.1(b)), the rate pair (R_1, R_2) is achievable if and only if they satisfy*

$$R_1 \leq t_{s_1} p_{s_1;d_1} + \min(t_{s_1} p_{s_1;r d_1^c}, t_r p_{r;d_1} - (R_2 - t_{s_2} p_{s_2;d_1 \cup d_2})^+) \quad (5.5)$$

$$R_2 \leq t_{s_2} p_{s_2;d_2} + \min(t_{s_2} p_{s_2;r d_2^c}, t_r p_{r;d_2} - \frac{p_{r;d_2}}{p_{r;d_1}} (R_1 - t_{s_1} p_{s_1;d_1 \cup d_2})^+). \quad (5.6)$$

5.3 The Capacity of two hop coding with only XOR operations

5.3.1 The Settings

We consider N -sessions 2 hop relay network with each channel represented by a PEC. Each source s_i intend to send symbols to d_i at rate R_i through the relay node. Direct communication between s_i and d_i is impossible. Relay node r is limited to perform XOR operations. After each transmission by the relay node r , it gets a feedback from all d_i nodes. We use t_r^A to represent the frequency that the relay node sends XORed packets formed by the packets of the sessions in the set A . We also use x_i^A to represent the achievable rate for session i from the auxiliary session formed by XORing packets from the sessions in the set A . Symbol x_i^{AB} represents the achievable rate for session i from the auxiliary session formed by XORing packets from the sessions in the set A with the constraint that session i packets that are used in XORing are received by exactly all the nodes in $r \cup (\bigcup_{j \in B} d_j)$ before being XORed. We use $R_{i,A}$ to represent the rate at which packets sent by s_i are overheard by all the nodes d_j , $j \in A, i \neq j$ at the time they are transmitted by the relay node.

5.3.2 The Characterization

The capacity region can be represented by the following four equations.

$$R_i \leq \sum_{A:i \in A} x_i^A, \forall i \quad (5.7)$$

$$x_i^A \leq t_r^A p_{r,d_i}, \forall A, i \in A \quad (5.8)$$

$$x_i^A = \sum_{B:(A \setminus i) \subseteq B} x_i^{AB} \forall A, i \in A \quad (5.9)$$

$$x_i^{AB} = R_{i,A}, \forall B, i \notin B \quad (5.10)$$

Necessity

Using XOR coding, any coded packet is formed by xoring packets of session i , $\forall i \in A$, where A is a set of sessions belonging to the power set of all sessions. Constraint (5.7) states that the total rate of session i is the sum of the achievable rate for session i from all of the auxiliary sessions A , where $i \in A$.

Since t_r^A is the frequency of sending XORed packets by the relay node formed by packets of the sessions in the set A , node d_i will receive XORed packets for the auxiliary session A from the relay node at rate $t_r^A p_{r,d_i}$. Therefore, constraint (5.8) should be satisfied for any achievable XOR based code.

Note also that (5.8) does not require the coded packet for the auxiliary session A to be received by all of d_i , $i \in A$, every time it is sent, any one of the d_i that receive this packet can decode it and it will count as a decodable packet.

For any auxiliary session A and $i \in A$, the set of the packets for session i that are used in this auxiliary session should be received from s_i by all the nodes in the set $r \cup (\bigcup_{j \in A, j \neq i} d_j)$, because r should be able to relay the XORed packets formed in part by these packets, and also because all d_j should have enough remedy to remove the components corresponding to these packets from the XORed packets and recover their respective packets. Also the set of packets for session i that are received from s_i

by any super set of $r \cup (\bigcup_{j \in A, j \neq i} d_j)$ can be used in the XORed auxiliary session A , because this will guarantee that all the nodes in the set $r \cup (\bigcup_{j \in A, j \neq i} d_j)$ have received these packets. This explains the constraint (5.9).

The right hand side of (5.10) $R_{i,A}$ represents the rate of session i packets received by exactly all of the nodes in the set $r \cup (\bigcup_{j \in A, j \neq i} d_j)$ after being sent by s_i . These packets can be used by any auxiliary session A such that $(A \setminus i) \subseteq B$. This is because this guarantees that all of the nodes $d_j, j \in A, i \neq j$ will have enough remedy packets to remove session i components in the XORed packets. Therefore, we have constraint (5.10). We postpone calculating a closed form expression for $R_{i,A}$ to the end of this section.

Note that the packets sent by s_i can be divided among all of the auxiliary sessions $A, i \in A$. This is due to the following.

- Because the right hand side of (5.10) represents the rate at which an exactly specific set of nodes are receiving the packets from s_i . Therefore every triple (i, A, B) can be assigned an exclusive share of these packets.
- Because each x_i^{AB} appears only once in (5.9), the packets of session i that are used in in the auxiliary session A will be $\bigcup_{B:(A \setminus i) \subseteq B} Y_i^{AB}$, where Y_i^{AB} are the set of set of packets assigned for the triple (i, A, B) .

An achievable coding scheme

-Node $s_i, \forall i$ keeps trying to send its NR_i packets one-by-one until all of them are received by the relay nodes.

-Feedback messages from all $d_j, j \neq i$ to s_i are sent to the relay node r .

-For every set A , the relay node chooses the corresponding feasible $x_i^A, \forall i$ from the linear program depending on the objective function. It also assign Nx_i^A packets for every A and i , such that these packets are received by r and all $j \in A, j \neq i$. As explained above, we can assign unique packets for every A .

-For every A the relay node XORs one packet from each Nx_i^A packets for all $i \in A$ and sends this packet. If this packet is received by d_j for $j \in A$, this means that the packets belonging to session j in the XORed packets can be recovered by d_j . Therefore, we remove this packet from the set of packets assigned to j and A at the relay node. The relay node keep performing the XORing and sending untill all of the packets assigned for the set A at the relay node are empty.

Computing $R_{i,A}$

In this section, we provide a closed form expression for $R_{i,A}$. Since we keep sending a packet until it is received by the relay node, we have.

$$\begin{aligned}
R_{i,A} &= t_{s_i} \times (\text{delivery rate from } s_i \text{ to } r) \\
&\quad \times (\text{probability that } r \text{ receives a} \\
&\quad \text{symbol and by that time the symbol is received by} \\
&\quad \text{exactly the nodes in } d_j, j \in A, j \neq i) \\
&= t_{s_i} p_{s_i,r} \left[\sum_{n=1}^{\infty} p_{s_i,r} (1 - p_{s_i,r})^{n-1} (\prod_{j \notin A} (1 - p_{s_i,d_j})^n) \prod_{j \in A} (1 - (1 - p_{s_i,d_j})^n) \right] \\
&= t_{s_i} p_{s_i,r}^2 \times \\
&\quad \sum_{n=1}^{\infty} \prod_{j \notin A} (1 - p_{s_i,d_j}) [(1 - p_{s_i,r}) \prod_{j \notin A} (1 - p_{s_i,d_j})]^{n-1} [\prod_{j \in A} (1 - (1 - p_{s_i,d_j})^n)] \\
&= t_{s_i} p_{s_i,r}^2 \prod_{j \notin A} (1 - p_{s_i,d_j}) \times \\
&\quad \sum_{n=0}^{\infty} [(1 - p_{s_i,r}) \prod_{j \notin A} (1 - p_{s_i,d_j})]^n [\prod_{j \in A} (1 - (1 - p_{s_i,d_j})^{n+1})] \\
&= t_{s_i} p_{s_i,r}^2 \prod_{j \notin A} (1 - p_{s_i,d_j}) \times \\
&\quad \sum_{n=0}^{\infty} \left[[(1 - p_{s_i,r}) \prod_{j \notin A} (1 - p_{s_i,d_j})]^n \left[\prod_{D: D \subseteq A} \prod_{k \in D} (1 - p_{s_i,d_k})^{n+1} \right] \right].
\end{aligned}$$

By Fubini's theorem we have.

$$\begin{aligned}
R_{i,A} &= t_{s_i} p_{s_i,r}^2 \prod_{j \notin A} (1 - p_{s_i,d_j}) \times \\
&\quad \prod_{k \in D} (1 - p_{s_i,d_k}) \left[\sum_{n=0}^{\infty} [(1 - p_{s_i,r}) \prod_{j \notin A} (1 - p_{s_i,d_j}) \prod_{k \in D} (1 - p_{s_i,d_k})]^n \right] \\
&= t_{s_i} p_{s_i,r}^2 \prod_{j \notin A} (1 - p_{s_i,d_j}) \times \\
&\quad \prod_{k \in D} (1 - p_{s_i,d_k}) \left[\frac{1}{1 - [(1 - p_{s_i,r}) \prod_{j \notin A} (1 - p_{s_i,d_j}) \prod_{k \in D} (1 - p_{s_i,d_k})]} \right]
\end{aligned}$$

5.4 Two-hop Relay Network Simulations

5.4.1 The Variants of Optimal Intersession Network Coding (INC)

The optimal capacity of INC in Proposition 5.2.3 is cast as a linear-programming problem and contains three orthogonal components: (i) INC via the new coding-based argument in Proposition 5.2.1, (ii) Opportunistic Routing (OpR) via modelling the overhearing opportunities as the 1-to-3 PECs of s_1 and s_2 in Fig. 5.1(a), and (iii) Cross-Layer (CL) design via variable transmission time t_{s_1} , t_{s_2} , and t_r . One can thus quantify the improvement of each component. For example, with CL we are allowed to optimally choose t_{s_1} to t_r with $t_{s_1} + t_{s_2} + t_r = 1$ assuming that the node exclusive model is used and s_1 , s_2 , and r cannot transmit simultaneously. If without CL, we simply set $t_{s_1} = t_{s_2} = t_r = 1/3$ for fair time-sharing between the sources and the relay. If OpR is not allowed, we simply use the extended version of Proposition 5.2.2 with added time variables t 's. We denote the four variants of the INC schemes, depending on whether we use OpR or CL or not, by (INC, OpR, CL), (INC, OpR, \times CL), (INC, \times OpR, CL), (INC, \times OpR, \times CL).

5.4.2 The Baseline Non-Coding Scheme and Its Variants

If we use a single-hop routing scheme without OpR, then the capacity region can be described as

$$R_i \leq \min(t_{s_i} p_{s_i;r}, t_r^{[i]} p_{r;d_i}), \forall i \in \{1, 2\}, \quad (5.11)$$

where $t_r^{[i]}$ is the amount of time that r sends the session- i packets. If we allow OpR, for which the packets can directly reach its 2-hop-away destinations, the capacity region becomes

$$R_i \leq t_{s_i} p_{s_i;d_i} + \min(t_{s_i} p_{s_i;r d_i^c}, t_r^{[i]} p_{r;d_i}), \forall i \in \{1, 2\}. \quad (5.12)$$

Whether to use CL or not will impose different time sharing constraints on t_{s_i} and $t_r^{[i]}$ in a similar way as discussed in Section 5.4.1. (5.11) and (5.12) can thus be used to generate four variants of non-coding schemes, depending on whether we use OpR or CL or not. We denote the four variants by $(\times\text{INC}, \text{OpR}, \text{CL})$, $(\times\text{INC}, \text{OpR}, \times\text{CL})$, $(\times\text{INC}, \times\text{OpR}, \text{CL})$, $(\times\text{INC}, \times\text{OpR}, \times\text{CL})$. The simplest scheme $(\times\text{INC}, \times\text{OpR}, \times\text{CL})$ will be used as the baseline of all our numerical comparison.

5.4.3 Schemes for More Than 2 Sessions

The $\times\text{INC}$ schemes described in (5.11) to (5.12) can be easily generalized to the case when there are $M > 2$ sessions, for which (5.11) and (5.12) hold for $i = 1$ to M ; and the time sharing constraints without CL become

$$\forall i \in \{1, \dots, M\} \quad t_{s_i} = \frac{1}{M+1}, \text{ and } \sum_{i=1}^M t_r^{[i]} = \frac{1}{M+1},$$

or when CL is used: $\sum_{i=1}^M t_{s_i} + \sum_{i=1}^M t_r^{[i]} = 1.$

Proposition 5.2.3 describes the INC-based schemes for $M = 2$. In [104], Proposition 5.2.3 has been generalized for $M = 3$ and new upper and lower bounds are

provided for general $M > 3$, which are empirically tight for most PEC channel parameters when $M \leq 5$. In our simulation results, we also include the curves for $M = 3$ to 5 for better comparison.

5.4.4 Practical, Low-Complexity Schemes for $M > 2$

Optimally performing INC over multiple sessions, say $M = 5$, is an exponentially complicated process as it requires carefully taking advantages of the various overheard portions of the data stream. In this paper, we thus consider some suboptimal but more practical schemes as follows.

Multipath Routing

Unlike the single-path routing, each source s_i can choose whether to directly send the packets to the 2-hop-away destination at the cost of higher drop rate (since $p_{s_i;d_i}$ is generally less than $p_{s_i;r}$), or to send the packets through the relay r . The corresponding capacity is thus characterized by the following linear program:

$$R_i \leq t_{s_i}^{[\text{direct}]} p_{s_i;d_i} + \min(t_{s_i}^{[\text{relay}]} p_{s_i;r}, t_r^{[i]} p_{r;d_i}).$$

With CL, we enforce that $\sum_{i=1}^M t_{s_i}^{[\text{direct}]} + t_{s_i}^{[\text{relay}]} + t_r^{[i]} \leq 1$.

2-INC and 3-INC

To reduce complexity of INC over multiple sessions, we can limit the number of sessions to be coded together and use linear programming to optimally identify which sessions to encode. For example, 2-INC denotes a scheme that allows INC only over two sessions while optimally allocating the corresponding time-sharing percentage. The following linear constraints describe the capacity region of 2-INC without OpR.

$$\begin{aligned}
R_i &\leq \sum_{j:j=i} R_i^{\{i,j\}} \quad \forall i \in \{1, \dots, M\} \\
\forall i, j, i \neq j, \quad R_i^{\{i,j\}} &\leq \min \left(t_{s_i}^{\{i,j\}} p_{s_i;r}, \right. \\
&\quad \left. t_r^{\{i,j\}} p_{r;d_i} - \min \left(1, \frac{p_{r;d_i}}{p_{r;d_j}} \right) \left(R_j^{\{i,j\}} - t_{s_j}^{\{i,j\}} p_{s_j;d_i} \right)^+ \right),
\end{aligned}$$

where $\{i, j\}$ is an unordered pair of distinct indices with $1 \leq i \neq j \leq M$. For example, both notations $t_{s_1}^{\{1,2\}}$ and $t_{s_1}^{\{2,1\}}$ refer to the same variable. There are $\binom{M}{2}$ different unordered $\{i, j\}$ pairs. Therefore, there are totally $2 \cdot \binom{M}{2}$ variables of $R_i^{\{i,j\}}$, $2 \cdot \binom{M}{2}$ variables of $t_{s_i}^{\{i,j\}}$, $\binom{M}{2}$ variables of $t_r^{\{i,j\}}$, and M variables of R_i . With CL, we require the sum of all $t_{s_i}^{\{i,j\}}$ and $t_r^{\{i,j\}}$ to be ≤ 1 . Similar formulation can be made by combining the 3-session INC capacity in [104] to derive a 3-INC scheme for general $M \geq 3$.

A multicast-based scheme in [27]

In Section 4 of [27] an INC scheme, denoted as the CCH scheme, is proposed based on using several multicast sessions (totally $2^M - 1$ multicast sessions) to serve the need of M unicast sessions. Each multicast session is indexed by $C \in 2^{\{1, \dots, M\}}$, which contains all sessions participate in this multicast session. The corresponding capacity region is described by the following linear constraints:

$$\begin{aligned}
R_i &\leq \sum_{C:i \in C} R_i^C, \quad \forall i \in \{1, \dots, M\} \\
R_i^C &\leq t_{s_i}^C p_{s_i;r}, \quad \forall i \in \{1, \dots, M\}, \forall C \in 2^{\{1, \dots, M\}} \\
R_i^C + \sum_{j:j \in A \setminus i} R_j^C &\leq t_r^C p_{r;d_i} + \sum_{j:j \in A \setminus i} t_{s_j}^C p_{s_j;d_i}, \\
&\quad \forall i \in \{1, \dots, M\}, \forall C \in 2^{\{1, \dots, M\}}, \forall A \subseteq C,
\end{aligned}$$

where R_i^C , $t_{s_i}^C$, and t_r^C are the rate of session i , and the time allocations of s_i and r for the multicast session C , respectively. With CL, we require the sum of all $t_{s_i}^C$ and t_r^C to be ≤ 1 .

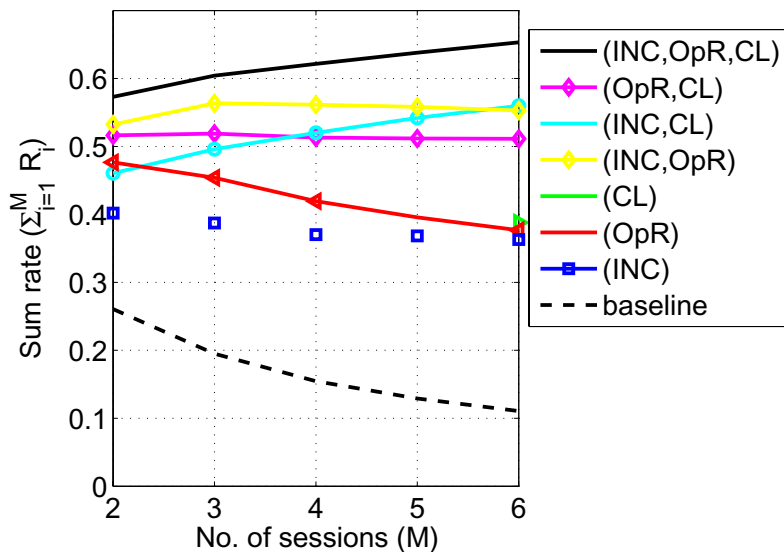


Fig. 5.5. The average sum rates for different schemes combining INC, OpR, and CL.

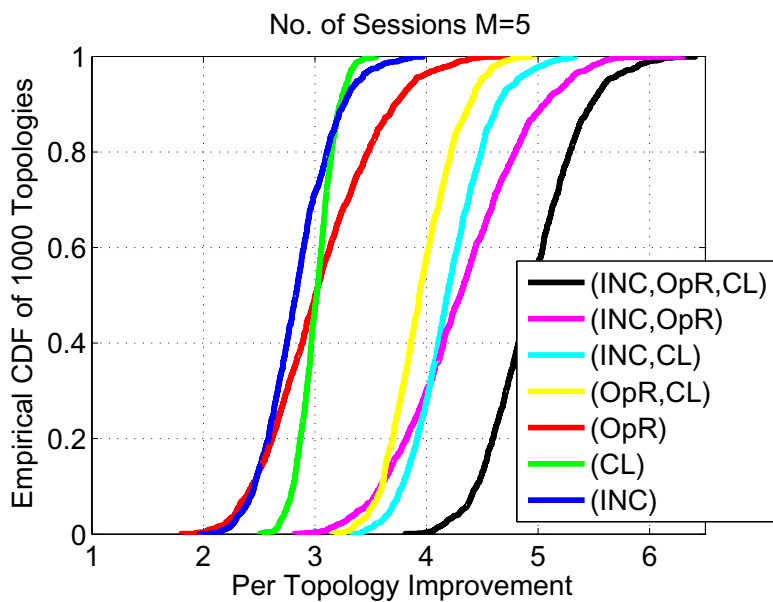


Fig. 5.6. The cumulative distribution function (CDF) of per topology improvement (PTI). The schemes in the legend corresponds to the curves from the rightmost to the leftmost when focusing on the intersection of CDF=0.6.

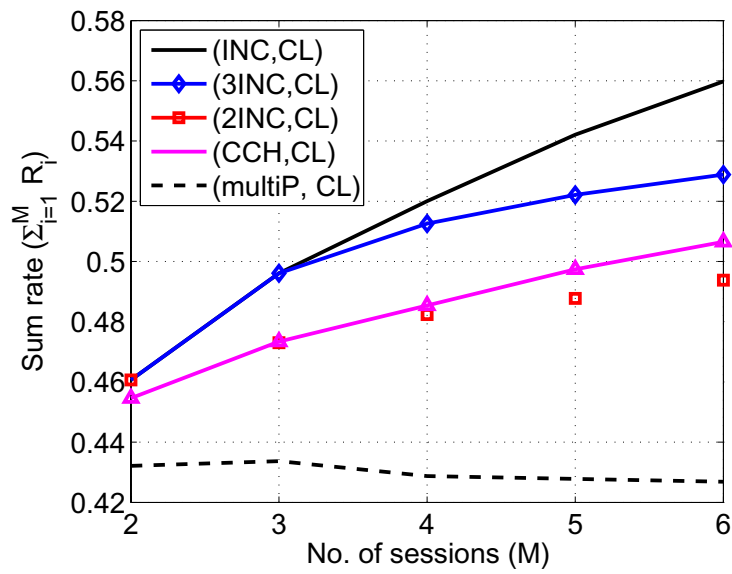


Fig. 5.7. The average sum rates for different practical schemes without OpR.

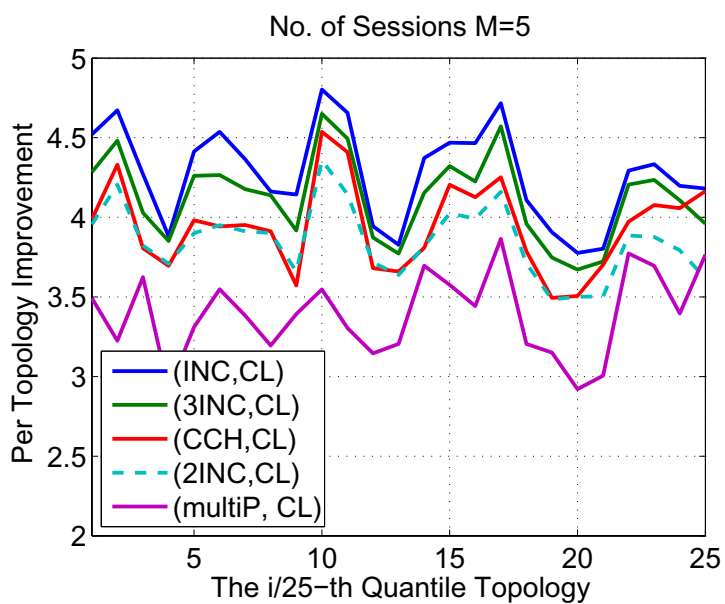


Fig. 5.8. The PTI for 25 representative topologies. These 25 topologies have their sum rates being the 25-th quantile points when sorting the sum rates in the ascending order. The list of schemes in the legend corresponds to the curves from the topmost to the bottommost when focusing on the performance of of the 15/25-th quantile point.

5.4.5 Numerical Experiments

We compare the throughput of various schemes on randomly constructed 2-hop relay networks. Our construction starts from a unit circle with the relay r placed at the center. We then uniformly randomly place M source nodes s_i and M destination nodes d_i in the circle (see Fig. 5.4(a)). The only condition we impose is that for each (s_i, d_i) pair, d_i must be in the 90-degree pie area opposite to s_i (see Fig. 5.4(b)). For each randomly constructed network, we use the Euclidean distance between each node to determine the overhearing probability. More explicitly, for any two nodes separated by distance D we use the Rayleigh fading model to decide the overhearing probability: $p = \int_{T^*}^{\infty} \frac{2x}{\sigma^2} e^{-\frac{x^2}{\sigma^2}} dx$, where we choose $\sigma^2 \triangleq \frac{1}{(4\pi)^2 D^\alpha}$, the path loss order $\alpha = 2.5$, and the decodable SNR threshold $T^* = 0.06$. Fig. 5.2 plots the overhearing probability p versus distance D . We assume that the overhearing event is independent among different receivers.

For each randomly generated network, we compute the overhearing probabilities and use the corresponding linear constraints on the time-sharing variables t 's and the rate variables R 's to compute the achievable rate of each scheme. A common linear objective function is used for all different schemes, which is described in the following:

$$\max_{t's, R's, \beta} \sum_{i=1}^M R_i \quad (5.13)$$

$$\text{subject to } R_i = \beta \min(p_{s_i;r \cup d_i}, p_{s_i;d_i} + p_{r;d_i}), \quad \forall i, \quad (5.14)$$

linear ineq. for the scheme of interest.

Namely, we are interested in maximizing the sum rate while (5.14) enforces that the rates are proportional to the unicast capacity with fair scheduling (assuming all other sessions $j \neq i$ remain silent). Constraint (5.14) thus ensures fairness proportional to the inherent interference-free capacity.

Given a randomly generated network, the achievable sum rates are computed for all the schemes. We then repeat this computation for 1000 randomly generated networks. Let $\zeta_{\text{scheme},k}^*$ denote the achievable sum rate of the given scheme for

the k -th randomly chosen topology. We are interested in the following two performance metrics. The average sum rate over 1000 topologies $\frac{1}{1000} \sum_{k=1}^{1000} \zeta_{\text{scheme},k}^*$ and per topology improvement $\triangleq \frac{\zeta_{\text{scheme},k}^*}{\zeta_{\text{baseline},k}^*}$.

A common baseline scheme, the vanilla (\times INC, \times OpR, \times CL) scheme as described in Section 5.4.2, is used when computing the Per Topology Improvement (PTI). The above procedure is then repeated for different M values.

Fig. 5.4.4 plots the sum rates of different schemes with respect to (w.r.t.) the number of sessions. To satisfy the proportional fairness constraint in (5.14), the sum rate of the baseline scheme decreases w.r.t. M due to the increased congestion at relay. On the other hand, schemes using CL and INCs successfully mitigate the congestion as their sum rates remain flat (even after satisfying (5.14)) for large M . When we jointly incorporate all three techniques: INC, OpR, and CL, not only the congestion is resolved but sum rate also increases w.r.t. M by taking full advantage of the spatial diversity, which amounts to x6.5 improvement over the baseline single-hop routing scheme when $M = 6$. We also plot the empirical cumulative distribution function (CDF) of PTI for the 1000 random topologies with $M = 5$. The (INC,OpR,CL) scheme achieves x3.8 to x6.5 throughput improvement when compared to the baseline scheme.

We also compare the practical schemes discussed in Section 5.4.4. Fig. 5.4.4 plots the average sum rates versus M . Among all practical schemes, 3-INC realizes the largest percentage of the gains of the optimal INC scheme. 2-INC schemes has similar performance to the CCH scheme for $M = 2-5$. The 2-INC scheme has lower complexity as it considers only $\binom{M}{2}$ pairs of sessions, while the CCH scheme jointly considers all $2^M - 1$ different multicast sessions. All INC schemes do not allow direct s_i to d_i communication (i.e., without OpR) but can still outperform the multipath-routing schemes with direct (s_i, d_i) links by an additional 10–20% for $M = 5$. Fig. 5.4.4 plots the PTI for 25 representative topologies whose sum rates are the $i/25$ -th quantile points when sorting the 1000 topologies according to the ascending order of the sum rates of the baseline scheme. As shown in Fig. 5.4.4, the practical INC schemes and

the optimal INC scheme demonstrate uniform PTI (3.5–4.7) for all topologies. In general, the more congested the original topology is, the higher PTI we have when using INC schemes (other than the multipath-routing scheme) to resolve congestion, which is illustrated by the negative trend of the PTI when moving from the 1st to the 25-th quantile point topology.

5.5 Conclusion

In this chapter, we have derived the capacity region for 2-hop relay network with two sessions under the PEC channel model and no opportunistic scheduling assumption. We have also characterized the capacity region of the same network with arbitrary number of sessions when only XOR operations are allowed. Both characterizations are represented by a linear program. We also provide capacity approaching codes for the characterized problems. Additionally we have constructed and compared various inter-session coding schemes for practical 2-hop relay networks both theoretically and numerically, which have demonstrated significant throughput benefits (x3.8–x6.5) when compared to the baseline single-path routing solution. In the next chapter we use our results in this chapter to provide a coding scheme that enhances the achievable rate region of a general multihop lossy wireless network.

6. INTERSESSION NETWORK CODING FOR LOSSY WIRELESS MULTIHOP NETWORKS

6.1 Main Contribution

In this Chapter we consider a general lossy wireless multihop network. For such networks the use of opportunistic scheduling is impractical. Therefore, two techniques have been used to maximize the capacity. These are opportunistic routing as in [27, 75, 107] and intrasession network coding as in [28, 108, 109]. These techniques do not allow intersession network coding which limits the achievable capacity. In this chapter we build on the results from the previous chapter to study the throughput benefits of using intersession network coding in such networks. We provide a linear programming formulation of the achievable rate region and compare it to the other previously used techniques.

6.2 Settings

We consider a general multihop wireless network represented by a graph $G = (V, E)$, where V is the set of vertices representing the nodes and E is the set of edges representing the links between the nodes. Transmission by a node can be overheard by multiple nodes and we model this by a hyperarc (i, J) , where i is the transmitter and J is a subset of the set of direct receivers. Every broadcast channel with K receivers is modelled by a 1- K PEC. There are N sessions in the network. For every session c , the source node $SRC(c)$ wants to send packets at rate f_c to the session's destination node $DST(c)$ over possibly multiple intermediate nodes. We use $\mathbb{P}(c)$ to refer to the path used for session c . For every node i on path $\mathbb{P}(c)$, $V_1(i, c)$ represents the next hop node on that path and $V_2(i, c)$ represents the next hop node of $V_1(i, c)$

on $\mathbb{P}(c)$. We assume that every sent packet is either a packet formed by intrasession coding packets of one session or intersession coding different sessions' packets. This include the case of sending non-coded packets as a special case. We use y_{ij}^c to represent the rate of linearly independent intrasession coded packets for session c that is sent by i and can be decoded by j , if j is $DST(c)$ or can be forwarded by j otherwise. Symbol y_{ij}^{cX} represent the same as y_{ij}^c but for intersession coded packets. The fraction of time node i is scheduled for sending session c intrasession coded packets is represented by α_i^c and α_i^X represents the fraction of time node i is scheduled to send intersession coded packets. Symbol α_i represents the fraction of time node i is scheduled.

6.3 Formulation without Intersession Network Coding

Without using intersession network coding and if we do not have specified paths, the linear constraints that specify the capacity region are as follows.

$$y_{j:i=j}^c - y_{j:i=j}^c \leq \begin{cases} -f_c & i = SRC(c) \\ 0 & \text{Else,} \end{cases} \quad \forall c, i \in E \setminus DST(c) \quad (6.1)$$

$$y_{j:j \in J}^c \leq \alpha_i^c p_{iJ}, \quad \forall (i, J), c. \quad (6.2)$$

where p_{iJ} is the probability that any node in J receive the packet.

The constraints in (6.1) represent balance equations such that the total received linearly independent packets and the total generated packets at a node should be at most equal to the totally sent linearly independent ones. Constraint (6.2) states that for any set of nodes that can receive the sent packets by a specific node, the total number of linearly independent packets that these nodes can forward equal to the probability that anyone of these nodes received the packet which is p_{iJ} .

If the paths are not specified, the solution of ((6.1)-(6.2)) will result in a back-pressure algorithm which has bad delay performance and might not converge to the optimal solution as noted in [72, 74]. Therefore, in the following we study the case of specified paths. The formulation becomes.

$$y_{j_1 i}^c + y_{j_2 i}^c - y_{i V_1(i,c)}^c - y_{i V_2(i,c)}^c \leq \begin{cases} -f_c & i = SRC(c) \\ 0 & \text{Else,} \end{cases} \quad \forall c, i \in E \setminus DST(c) \quad (6.3)$$

$$\text{Where } i = V_1(j_1, c), i = V_2(j_2, c) \quad (6.4)$$

$$y_{ij}^c \leq \alpha_i^c p_{ij}, \quad \forall c, \quad i, j \in \mathbb{P}(c) \quad (6.5)$$

$$y_{i V_1(i,c)}^c + y_{i V_2(i,c)}^c \leq \alpha_i^c (p_{i;V_1(i,c)} + p_{i;V_2(i,c)} - p_{i;V_1(i,c)} p_{i;V_2(i,c)}) \quad \forall i \in \mathbb{P}(c). \quad (6.6)$$

The above formulation can be obtained by noting that the only hyperarc for a node with session c is the one with the receivers being $V_1(i, c)$ and $V_2(i, c)$. This modelling agrees with practical implementations of intrasession network coding as in [28, 108] which state that overhearing of a node transmission over a path happens only for one and two hop away nodes.

6.4 Extension to Intersession Network Coding

6.4.1 Single hop Coding Restrictions for Intersession Network Coding

It is hard to characterize the general capacity region of lossy wireless multihop networks. Therefore, in this section we provide a restriction under which we can characterize the capacity region by linear constraints. The restriction is that intersession network coding is limited to be in the form of 2-hop relay network, where the coding node is the relay node and the decoding nodes are subset of the next hop nodes of the relay node. Therefore, the general lossy multihop network can be decomposed into a superposition of non-intersession coding traffic and intersession network coding traffic in 2-hop relay networks. The assumption we have has the following implications on the capacity region.

- In Fig. 6.1 if there are two sessions one of them goes through the path $v_1 w_2 v_2 r v_3$ and the other one goes through $u_1 u_2 r u_3$. Intersession network coding can happen at node r . Due to our restriction, we assume there is no side information from w_2 to u_3 nor from u_1 to v_3 . If such side information exist, we ignore them.

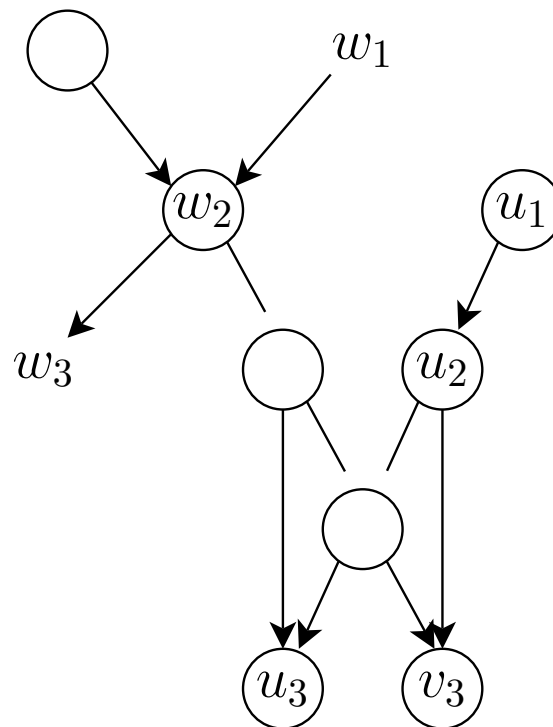


Fig. 6.1. An example of network with two flows to explain the single hop coding restrictions.

- If in the same figure there is another session that goes through $w_1w_2w_3$, and node w_2 is acting as a relay node for performing intersession network coding between this session and the session that goes through $v_1w_2v_2rv_3$, and the intersession coded packets are overheard by r , node r will deal with these packets as useless and drop them.

6.4.2 The Capacity of 2-hop Relay Network

In Chapter 5 we have characterized the capacity region of 2-hop relay network with 2 sessions. In order to use these results to enhance the achievable rate for multihop networks, we need to extend the characterization to the case of arbitrary number of sessions. Based on our results in Chapter 5, the work in [104] does the

extension. In the extension, both an upper and lower bounds were provided. For the case of less than four sessions the two bounds are the same which represents the capacity region. For four sessions or more the two bounds agrees 99% of the time. Both the upper and lower bounds can be solved by linear programs. Under the assumption that the channels are independent, let $\boldsymbol{\alpha} \triangleq [\alpha_r, \alpha_{s_1}, \dots, \alpha_{s_N}]$, $\mathbf{p}_{s_1\mathbf{d}} \triangleq [p_{s_1;d_1}, \dots, p_{s_1;d_{l-1}}, p_{s_1;d_{l+1}}, \dots, p_{s_1;d_N}]$, $\mathbf{p}_{s\mathbf{d}} \triangleq [\mathbf{p}_{s_1\mathbf{d}}, \dots, \mathbf{p}_{s_N\mathbf{d}}]$, $\mathbf{p}_s \triangleq [p_{s_1;r}, \dots, p_{s_N;r}]$, and $\mathbf{p}_d \triangleq [p_{r;d_1}, \dots, p_{r;d_N}]$. In the case when d_l cannot overhear s_l , we use

$$\text{CapO}(\boldsymbol{\alpha}, \mathbf{p}_{s\mathbf{d}}, \mathbf{p}_s, \mathbf{p}_d) = \{(R_1, \dots, R_N) : \text{The rates } R_1, \dots, R_N \text{ satisfy the upper bound constraints.}\}$$

and we use

$$\text{CapI}(\boldsymbol{\alpha}, \mathbf{p}_{s\mathbf{d}}, \mathbf{p}_s, \mathbf{p}_d) = \{(R_1, \dots, R_N) : \text{The rates } R_1, \dots, R_N \text{ satisfy the lower bound constraints.}\}$$

For example when $N = 2$, the rates that satisfy

$$\begin{aligned} R_1 &\leq \min(\alpha_{s_1} p_{s_1;r}, \alpha_r p_{r;d_1} - (R_2 - \alpha_{s_2} p_{s_2;d_1})^+) \\ R_2 &\leq \min(\alpha_{s_2} p_{s_2;r}, \alpha_r p_{r;d_2} - (R_1 - \alpha_{s_1} p_{s_1;d_2})^+ \frac{p_{r;d_2}}{p_{r;d_1}}) \end{aligned}$$

belong to both $\text{CapO}(\boldsymbol{\alpha}, \mathbf{p}_{s\mathbf{d}}, \mathbf{p}_s, \mathbf{p}_d)$ and $\text{CapI}(\boldsymbol{\alpha}, \mathbf{p}_{s\mathbf{d}}, \mathbf{p}_s, \mathbf{p}_d)$, because both the upper and lower bounds are the same when $N = 2$. When d_l can overhear s_l , and if d_l forwards $\gamma_l p_{s_l;d_l}$ linearly independent symbols of the overheard packets or decode them if it is the last destination of the packets, intersession network coding should happen for the symbols in the complementary spaces of the forwarded or decoded symbols. Let $\boldsymbol{\gamma} \triangleq [\gamma_1, \dots, \gamma_N]$ and $\mathbf{p}'_{s\mathbf{d}} \triangleq [p_{s_1;d_1}, \dots, p_{s_N;d_N}]$ in this case we use

$$\text{Cap}'\text{O}(\boldsymbol{\gamma}, \boldsymbol{\alpha}, \mathbf{p}'_{s\mathbf{d}}, \mathbf{p}_{s\mathbf{d}}, \mathbf{p}_s, \mathbf{p}_d) = \{(R_1 - \gamma_1 p_{s_1;d_1}, \dots, R_N - \gamma_N p_{s_N;d_N}) :$$

The rates R_1, \dots, R_N satisfy the upper bound constraints when d_l can overhear s_l .}

and we use

$$\text{Cap}'\text{I}(\boldsymbol{\gamma}, \boldsymbol{\alpha}, \mathbf{p}'_{s\mathbf{d}}, \mathbf{p}_{s\mathbf{d}}, \mathbf{p}_s, \mathbf{p}_d) = \{(R_1 - \gamma_1 p_{s_1;d_1}, \dots, R_N - \gamma_N p_{s_N;d_N}) :$$

The rates R_1, \dots, R_N satisfy the inner bound constraints when d_l can overhear s_l .}

For example when $N=2$ we any (R'_1, R'_2) that satisfy the following constraints belong to both $\text{Cap}'\text{O}(\gamma, \alpha, \mathbf{p}'_{\text{sd}}, \mathbf{p}_{\text{sd}}, \mathbf{p}_{\text{s}}, \mathbf{p}_{\text{d}})$ and $\text{Cap}'\text{I}(\gamma, \alpha, \mathbf{p}'_{\text{sd}}, \mathbf{p}_{\text{sd}}, \mathbf{p}_{\text{s}}, \mathbf{p}_{\text{d}})$, because both the upper and lower bounds agree when $N = 2$.

$$\begin{aligned} R'_1 &\leq \min(Y_1, \alpha_r p_{r;d_1} - (R_2 - Z_1)^+) \\ R'_2 &\leq \min(Y_2, \alpha_r p_{r;d_2} - (R_1 - Z_2)^+) \frac{p_{r;d_2}}{p_{r;d_1}}, \end{aligned}$$

where Y_1, Y_2, Z_1, Z_2 satisfy the following.

$$\begin{aligned} Y_1 &\leq \alpha_{s_1}(p_{s_1;r} + p_{s_1;d_1} - p_{s_1;d_1}p_{s_1;r}) - \gamma_{s_1;d_1}p_{s_1;d_1} \\ Y_2 &\leq \alpha_{s_1}p_{s_1;r} \\ Y_2 &\leq \alpha_{s_2}(p_{s_2;r} + p_{s_2;d_2} - p_{s_2;d_2}p_{s_2;r}) - \gamma_{s_2;d_2}p_{s_2;d_2} \\ Y_1 &\leq \alpha_{s_2}p_{s_2;r} \\ Z_1 &\leq \alpha_{s_2}(p_{s_2;d_1} + p_{s_2;d_2} - p_{s_2;d_1}p_{s_2;d_2}) \\ Z_2 &\leq \alpha_{s_1}(p_{s_1;d_2} + p_{s_1;d_1} - p_{s_1;d_2}p_{s_1;d_1}) \end{aligned}$$

Using random network coding and when considering the symbols directly received from s_i by any d_m or r , any two symbols related to two different received packets are linearly independent. Therefore, using the feedback the relay will be able to know the coefficients related to the received packets by its next hop nodes to generate packets with coefficients in their complementary space.

6.4.3 Formulation with Intersession Network Coding

The following linear equations represent an upper bound on the capacity region.

$$y_{j_1 i}^c + y_{j_2 i}^c + y_{j_1 i}^{cX} - y_{iV_1(i,c)}^c - y_{iV_2(i,c)}^c - y_{iV_1(i,c)}^{cX} \leq \begin{cases} -f_c & i = SRC(c) \\ 0 & \text{Else,} \end{cases} \quad \forall c, i \in E \setminus DST(c)$$

$$\text{Where } i = V_1(j_1, c), i = V_2(j_2, c) \quad (6.7)$$

$$y_{ij}^c = \gamma_{ij}^c p_{ij} \leq \alpha_i^c p_{i,j}, \quad \forall c, \quad i, j \in \mathbb{P}(c) \quad (6.8)$$

$$y_{iV_1(i,c)}^c + y_{iV_2(i,c)}^c \leq \alpha_i^c (p_{i;V_1(i,c)} + p_{i;V_2(i,c)} - p_{i;V_1(i,c)} p_{i;V_2(i,c)}) \quad \forall i \in \mathbb{P}(c). \quad (6.9)$$

$$(y_{iV_1(i,c_1(i))}^{c_1(i)X}, \dots, y_{iV_1(i,c_k(i))}^{c_k(i)X}) \in \text{Cap}'\text{O}(\boldsymbol{\gamma}^i, \boldsymbol{\alpha}^i, \mathbf{P}_{\text{sd}}^i, \mathbf{P}_{\text{sd}}^i, \mathbf{P}_{\text{s}}^i, \mathbf{P}_{\text{d}}^i). \quad (6.10)$$

Here, k is the number of sessions intersecting at node i . These sessions are $c_1(i), \dots, c_k(i)$ and we use $j_1(i, c_l(i))$ such that $i = V_1(j_1(i, c_l(i)), c_l(i))$. Also, we have $\boldsymbol{\gamma}^i \triangleq [\gamma_{j_1(i,c_1(i))V_1(i,c_1(i))}^{c_1(i)}, \dots, \gamma_{j_1(i,c_k(i))V_1(i,c_k(i))}^{c_k(i)}]$, $\boldsymbol{\alpha}^i \triangleq [\alpha_i^X, \alpha_{j_1(i,c_1(i))}^{c_1(i)}, \dots, \alpha_{j_1(i,c_k(i))}^{c_k(i)}]$, $\mathbf{P}_{\text{sd}}^i \triangleq [p_{j_1(i,c_1(i));V_1(i,c_1(i))}, \dots, p_{j_1(i,c_k(i));V_1(i,c_k(i))}]$, $\mathbf{P}_{\text{sd}}^i \triangleq [p_{j_1(i,c_l(i));V_1(i,c_l(i))}, \dots, p_{j_1(i,c_l(i));V_1(i,c_{l-1}(i))}, p_{j_1(i,c_l(i));V_1(i,c_{l+1}(i))}, \dots, p_{j_1(i,c_l(i));V_1(i,c_k(i))}]$, $\mathbf{P}_{\text{sd}}^i \triangleq [\mathbf{P}_{\text{sd}}^i, \dots, \mathbf{P}_{\text{sd}}^i]$, $\mathbf{P}_{\text{s}}^i \triangleq [p_{j_1(i,c_1(i));i}, \dots, p_{j_1(i,c_k(i));i}]$, and $\mathbf{P}_{\text{d}}^i \triangleq [p_{i;V_1(i,c_1(i))}, \dots, p_{i;V_1(i,c_k(i))}]$. For session c , any node i has three different kinds of incoming packets and three different kinds of outgoing packets. The incoming packets types are intrasession received from previous hop with rate $y_{j_1 i}^c$, intrasession packets overheard from two-hop away node with rate $y_{j_2 i}^c$, and intersession coded packets received from previous hop with rate $y_{j_1 i}^{cX}$. Note that due to the restriction we have, intersession coded packets overheard from two hop away nodes are dropped. The outgoing packets can also be classified as intersession with rate $y_{iV_1(i,c)}^{cX}$, intrasession that is received and used by the next hop with rate $y_{iV_1(i,c)}^c$ and intrasession that is overheard and used by the next two hop away nodes with rate $y_{iV_2(i,c)}^c$.

The constraints in (6.7) state that at every node and for every session the total incoming traffic at a node should be equal to the total outgoing traffic. The constraints (6.8)-(6.9) are for intrasession network coding and are the same as in the previous section. Constraints (6.10) specify the intersession coding rate at node i by

treating it as a relay node in a 2-hop relay network. Due to the restriction we have, at node i only the incoming intrasession coded traffic from previous hop can be used for intersession network coding at node i . This is reflected in the formulation by using α^i as the second arguments of Cap'O which only contains the intrasession scheduling frequency of the previous hop nodes of node i . Since $\gamma_{j_1(i, c_l(i))V_1(i, c_l(i))}^{c_1(i)} \mathcal{P}_{j_1(i, c_l(i)); V_1(i, c_l(i))}$ is the rate of the intrasession coded packets for session $c_l(i)$ that are sent by node $j_1(i, c_l(i))$ and overheard by the the node $V_1(i, c_l(i))$ and used by that node, the first argument in Cap'O states that intersession network coding is performed in the complementary space of the symbols related to these packets.

6.5 Numerical Results

To show the benefits of intersession network coding, we use randomly generated topologies of fifteen nodes located in a 6×6 unit square area. We compute the delivery rate between the nodes the same way as in previous chapter. We choose $\alpha = 2.5$ and $T^* = 0.1$. For simplicity, we assume that the channels are orthogonal and every node can be scheduled in every time slots. This can be achieved by equipping the nodes with multiple-input and multiple-output (MIMO). For each possible source and sink pair we find the path that minimize the ETX metric as defined in [110]. For each source, we mark the longest path among the found paths that minimize the ETX metric. We randomly select K paths from the marked ones to perform the simulation. For each value of K we simulated 200 different topologies. Fig 6.2 represents the average gain of intersession network coding ((6.7)-(6.10)) over the intrasession network coding solution ((6.4)-(6.6)) when the objective is to maximize the total throughput. We report the network coding gain when the lower and upper bounds agree. As the number of the sessions increases, the gain of intersessions network coding increases due to the fact that we have more sessions going through a node which increases the intersession coding opportunities. Figure 6.3 shows that intersession network coding

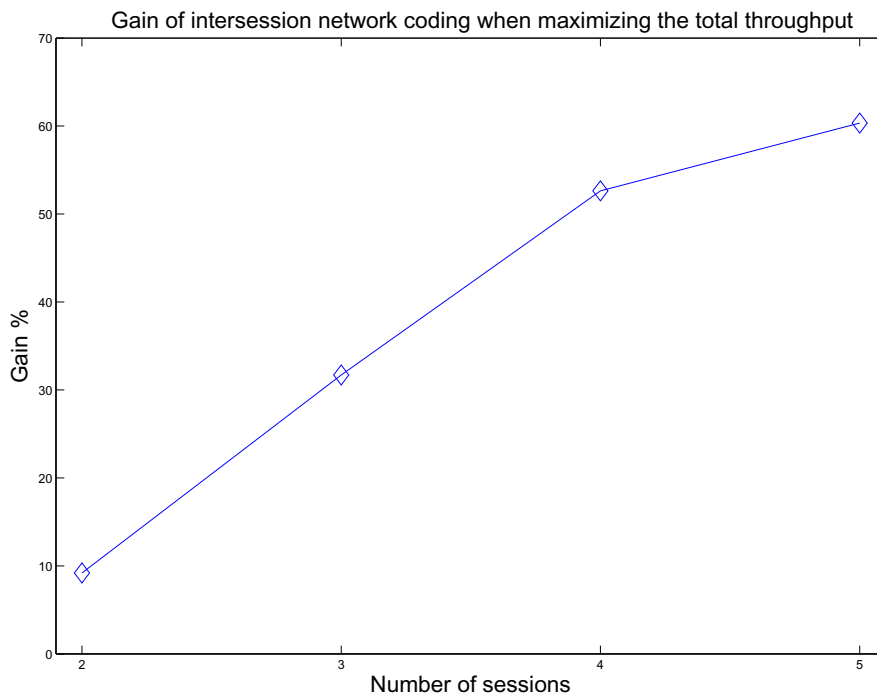


Fig. 6.2. Average Gain of Intersession network coding for randomly generated topologies with respect to the number of sessions in the network. The Objective is to maximize the total throughput of the network.

gain is even more when the objective function is to achieve strict fairness among the flows.

6.6 Conclusions

In this chapter we provided a formulation for using intersession network coding in lossy multihop wireless network. The formulation is in a form of linear program. To achieve the throughput provided by the formulation, random linear coding is required. The capacity achieved by the linear constraints is not in general the capacity region of multihop wireless networks, but our simulations show large gain over the traditional

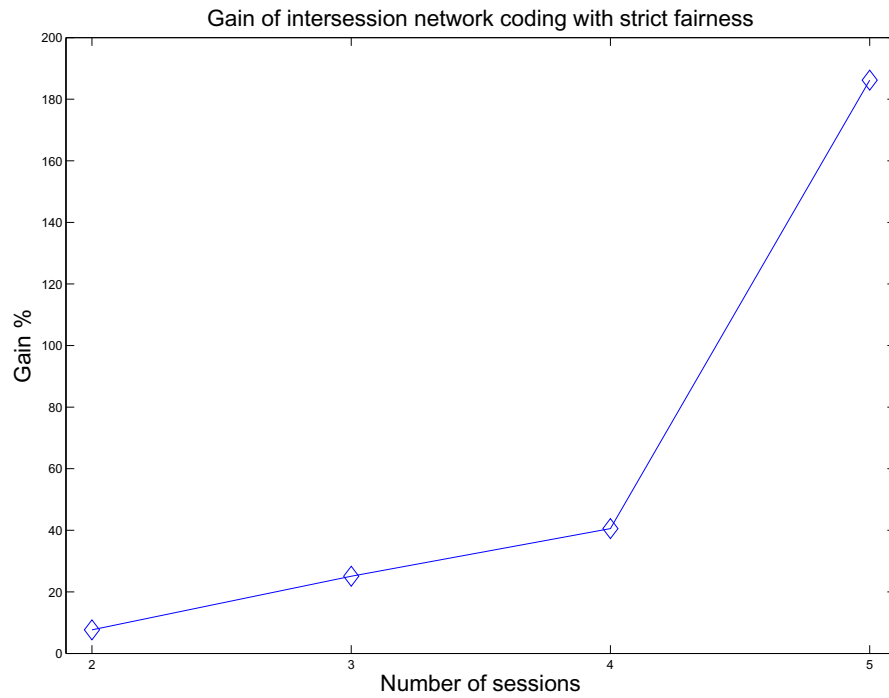


Fig. 6.3. Average Gain of Intersession network coding for randomly generated topologies with respect to the number of sessions in the network. The Objective is to achieve strict fairness among the flows.

intrasession network coding schemes. The gain is also increasing with respect to the number of sessions in the network.

7. CONCLUSION AND FUTURE WORK

7.1 Summary of Contribution

Intersession network coding is a very rich research area. The major open problem in this area is to explicitly characterize the capacity region of the network with many sources sending at different rates to different sets of sinks and to design the corresponding coding scheme that can achieve the capacity. Since such problem is shown to be NP-hard and linear network coding is not sufficient for such a problem, in this dissertation we put some restrictions on the number of sessions to be mixed together, or the structure of the topology such that optimal solutions can be implemented in a distributed and fast way using linear coding.

For wireline networks, we restricted any coded packet to be a result of coding at most two original packets. Based on the results in [66, 67], we have designed both optimal distributed rate control algorithm and optimal distributed coding scheme for the problem of multiple-unicast-sessions. The decision of assigning rates at different links in the rate control algorithm is controlled by the queue lengths at different links and done in a distributed way. In the coding scheme few links are selected in a distributed way; random network coding is performed in all of the remaining links and decoding operations are performed on the selected links.

The results obtained in wireline networks were extended to wireless networks with opportunistic scheduling capability. This means that the channel conditions are known prior to transmission. We developed cross-layer rate control frameworks for wireless multihop networks. The solutions require minimal interactions between the layers where every layer has its own distributed algorithm. Two approaches have been used to develop the algorithms. The first approach is a path-based one which converges quickly, and makes it easier to study the impact of imperfect scheduling under

both the static and stochastic arrival models. The second approach is a back-pressure approach where the framework can be extended to include energy minimization. The back-pressure approach also allows the decoded packets to be reencoded again which results in an improved capacity region. The back-pressure algorithm searches for local coding opportunities which results in a reduced complexity algorithm. The coding scheme developed for wireline networks can be used in wireless networks, but it requires large computation processing, which is not suitable for most wireless networks. Therefore, we proposed an XOR based coding scheme that achieves the same performance as the coding scheme designed for wireline networks with a lower processing overhead.

We studied another wireless networks settings by removing the assumption of opportunistic scheduling to consider more realistic conditions. For this channel model, we proposed a coding scheme for single-hop intersession network coding that outperforms existing schemes. The relay node uses its knowledge about the erasure probabilities of the links to decide whether to perform intersession or intrasession network coding. We characterized the capacity region obtained by this scheme using a set of linear equations. When the number of sessions in the network is two, we show that the scheme achieves the capacity region of the network.

In the following we discuss different future research directions. Some of these directions are extensions of the work in this dissertation and hence can be conducted under the same settings considered here i.e., either (i) pairwise intersession network coding for both wireline networks and wireless networks with opportunistic scheduling capability, or (ii) single-hop intersession network coding for wireless networks without opportunistic scheduling capability. The rest are under new proposed settings. We believe that such new proposed settings might result in an easy characterizations and implementable algorithms for intersession network coding.

7.2 Future Research Directions

7.2.1 Integrating Our Solution with TCP

The technique we provided in Chapter 2 achieves the optimal solution with pairwise intersession network coding. However, it is difficult to implement this approach in practice, because the internet which is the most common wireline network uses TCP. Therefore, it is crucial for our approach to be integrated with TCP and the internet. Many challenges arise in this direction. One of these challenge is that TCP uses window flow control and the objective function that it tries to maximize is implicit. Also, TCP uses special kind of feedback messages, handshaking mechanism, and specific packet format. Therefore, it is crucial to reverse engineer TCP and study the effect of integrating pairwise intersession network coding with TCP.

7.2.2 Distributed Scheduling Algorithm for Pairwise Intersession Network Coding under more General Interference Models.

The frameworks in Chapters 3 and 4 decompose the problem into different problems for each different layer. The most difficult problem is the MAC layer problem which might be NP-hard in some cases. The requirements are to develop distributed and low complexity algorithm with good performance guarantees, because it has to run at each time slot. The developed algorithms should also take into account that network coding exploits the broadcast advantage of wireless networks. This makes the problem with network coding different from that without network coding. The algorithm developed in Chapters 3 and 4 is based on the assumption of the node exclusive model. Also its worst case performance guarantee is $\frac{1}{3}$ of the optimal solution. Without network coding there have been many scheduling algorithms developed for more general interference models with good performance guarantees [111–117]. Insights provided by these algorithms can be used as the first step towards developing distributed MAC layer algorithms for network coding with good performance guar-

antees. The characterization of broadcast link size in Chapter 4 can also help in reducing the complexity of any proposed algorithm.

7.2.3 The Capacity Region of Single-hop Intersession Network Coding without Opportunistic Scheduling and More than Two sessions

In Chapter 5 we provided a coding scheme that can achieve the capacity of the network under the assumption of single-hop intersession network coding, no opportunistic scheduling, and erasure channel model. The scheme achieves the capacity region when the number of sessions is two. When the number of sessions is more than two, the coding scheme provides improvement over existing schemes, but no proof of the capacity achievement was provided. One extension of the work in Chapter 5 would be to prove or disprove that the provided coding scheme achieves the capacity. If it does not achieve the capacity, it would be interesting to know how much close to the capacity does it perform. Another extension would be to design network codes for other channel models like gaussian channels.

7.2.4 More Complex Pairwise Intersession Network Coding

The main result in [66,67] is a characterization theorem for the necessary and sufficient conditions for transmitting two symbols through two sessions such that one symbol belongs to each session. One extension of the results in [66,67] is to consider sending more than one symbol through each session. The superposition approach in Chapter 2 does not achieve the capacity in general. One example where the capacity can not be achieved using the superposition approach is the network in Fig. 7.1. For this network the rate region $(R_1 = 1, R_2 = 2)$ is not achievable using the superposition approach, while the linear coding scheme in the figure achieves this capacity region. It can be shown using the results in [11], that $(R_1 = 1, R_2 = 2)$ is the capacity region of the network. As can be noted from Fig. 7.1, both intersession and intrasession network coding are required to achieve the capacity. We expect the necessary and

sufficient conditions for two sessions such that more than one symbol is sent through each session to be cut based conditions.

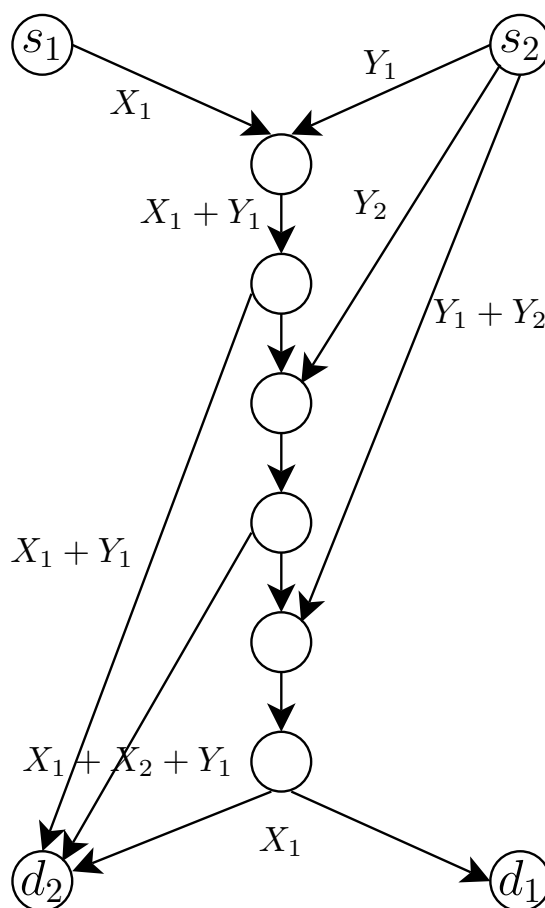


Fig. 7.1. An example where the superposition approach in Chapter 2 does not achieve the capacity of the network.

7.2.5 k -hop intersession network coding without opportunistic scheduling

In this dissertation we have studied pairwise intersession network coding that is not limited to single-hop coding with the assumption of opportunistic scheduling. When the opportunistic scheduling assumption was removed, we developed optimal coding schemes policies for single-hop intersession network coding. It worths investigating

intersession network coding over k -hops when opportunistic scheduling is not possible. Fig. 7.2 gives a two-hop example where the capacity region represented by $(R_1 = 0.5, R_2 = 0.5)$ is achievable, but neither opportunistic routing nor a superposition of single-hop intersession network coding and opportunistic routing can achieve the rate region.

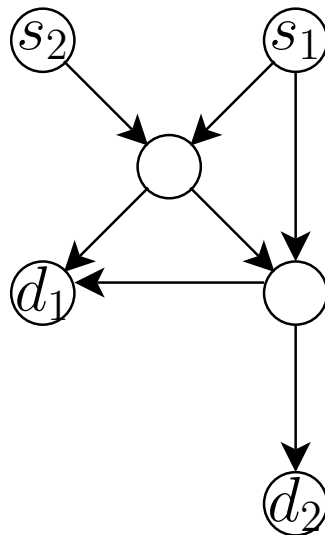


Fig. 7.2. An example to illustrate the importance of characterizing intersession network coding for more than one hop. Every link has a delivery probability of 0.5.

7.2.6 Reduced Complexity and provably efficient coding scheme for 2-hop relay network with PEC model

The generalization of our scheme in Chapter 5 that is presented in [104] requires exponential number of decisions with respect to the number of sessions at the relay node. This makes this scheme impractical when the number of sessions is more than 5. One direction worthy of study is to propose schemes that requires linear or polynomial number of decisions at the relay node and to study their performance loss.

7.2.7 Distributed Algorithms for Intersession Network Coding in lossy Multihop Wireless Networks.

One interesting direction worthy of investigation is to achieve the gains in Chapter 6 in a distributed way, or to propose suboptimal distributed schemes and study their performance loss compared to the one in Chapter 6.

LIST OF REFERENCES

LIST OF REFERENCES

- [1] P. B. et al., “On distributed communications, vols. i-xi,” *RAND Corporation Research Documents.*, Aug 1964.
- [2] Chen and Wai-Kai, *Theory of nets : flows in networks.* New York: Wiley, 1990.
- [3] K. Menger, “Zur allgemeinen kurventheorie,” *Fund. Math.*, vol. 10, p. 95115, 1927.
- [4] R. Ahlswede, N. Cai, S.-Y, R. Li, and R. W. Yeung, “Network information flow,” *IEEE Trans. on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [5] K. Bharath-Kumar and J. M. Jaffe, “Routing to multiple destinations in computer networks,” *IEEE Trans. Commun.*, vol. 31, p. 343351, 1983.
- [6] R. Li, R. W. Yeung, and N. Cai, “Linear network coding,” *IEEE Trans. on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.
- [7] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, “Polynomial time algorithms for multicast network code construction,” *IEEE Trans. on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.
- [8] R. Koetter and M. Medard, “Beyond routing: An algebraic approach to network coding,” in *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, New York, June 2002.
- [9] T. Ho, R. Koetter, M. Medard, M. Effros, J. Shi, and D. Karger, “A random linear network coding approach to multicast,” *IEEE Trans. on Information Theory*, vol. 52, pp. 4413–4430, october 2006.
- [10] N. Harvey, R. Kleinberg, and A. Lehman, “On the capacity of information network,” *IEEE Trans. inform. Theory*, vol. 52, no. 6, p. 23452364, 2006.
- [11] X. Yan, J. Yang, and Z. Zhang, “An outer bound for multisource multisink network codign with minimum cost consideration,” *IEEE Trans. inform. Theory*, vol. 52, no. 6, p. 23732385, 2006.
- [12] G. Kramer and S. Savari, “Edge-cut bounds on network coding rates,” *Journal of Network and Systems Management*, vol. 14, no. 1, p. 4967, 2006.
- [13] J. Cannons and K. Zeger, “Network coding capacity with a constrained number of coding nodes,” *IEEE Trans. inform. Theory*, vol. 54, no. 3, p. 12871291, 2008.
- [14] K. Jain, V. Vazirani, R. Yeung, and G. Yuval, “On the capacity of multiple unicast sessions in undirected graphs,” in *Proc. IEEE Int’l Symp. Inform. Theory*, July 2006.

- [15] J. Liu, D. Goeckel, and D. Towsley, "Bounds on the gain of network coding and broadcasting in wireless networks,," in *26th IEEE Conference on Computer Communications (INFOCOM) Anchorage, Alaska, USA,*, May 2007.
- [16] A. Keshavarz-Haddad and R. Riedi, "Bounds on the benefit of network coding: Throughput and energy saving in wireless networks,," in *27th IEEE Conference on Computer Communications (INFOCOM). Phoenix, USA,*, April 2008.
- [17] D. Lun, N. Ratnakar, M. Médard, R. Koetter, D. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2608–2623, 2006.
- [18] L. Chen, T. Ho, S. H. Low, M. Chiang, and J. C. Doyle, "Optimization based rate control for multicast with network coding," in *Proc. of IEEE Conference on Computer Communications (INFOCOM), Anocharage, AK,* May 2007.
- [19] Y. Wu and S.-Y. Kung, "Distributed utility maximization for network coding based multicasting: a shortest path approach," *IEEE J. on Selected Areas in Communications*, vol. 24, pp. 1475–1488, Apr 2006.
- [20] Y. Wu, M. Chiang, and S.-Y. Kung, "Distributed utility maximization for network coding based multicasting: a critical cut approach," in *Proc. Workshop on Network Coding, Theory, & Applications (NetCod)*, Apr 2006.
- [21] T. Cui, L. Chen, and T. Ho, "Optimization based rate control for multicast with network coding: A multipath formulation," in *IEEE Conf. Decision and Contr. New Orleans,*, Dec 2007.
- [22] C.-C. Wang and X. Lin, "Fast resource allocation for network-coded traffic - a coded-feedback approach," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM), mini conference, Rio de Janeiro, Brazil,* April 2009.
- [23] Y. Sagduyu and A. Ephremides, "On joint mac and network coding in wireless ad hoc networks," *IEEE Trans. inform. Theory*, vol. 53, no. 10, p. 36973713, 2007.
- [24] Y. Sagduyu and A. Ephremides, "Cross-layer optimization of mac and network coding in wireless queueing tandem networks," *IEEE Trans. inform. Theory*, vol. 54, no. 2, p. 554571, 2008.
- [25] T. Cui, L. Chen, and T. Ho, "Distributed optimization in wireless networks using broadcast advantage," in *IEEE Conf. Decision and Contr. New Orleans,*, Dec 2007.
- [26] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proc. ACM Special Interest Group on Data Commun. (SIGCOMM), Pisa, Italy,* Sept 2006.
- [27] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks,," in *27th IEEE Conference on Computer Communications (INFOCOM). Phoenix, USA,*, April 2008.
- [28] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *ACM Special Interest Group on Data Commun. (SIGCOMM) Kyoto, Japan,*, Aug 2007.

- [29] M. Wang and B. Li, “ R^2 : Random push with random network coding in live peer-to-peer streaming,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1655–1666, 2007.
- [30] “Avalanche: Peer-assisted content distribution, <http://research.microsoft.com/camsys/avalanche/>.”
- [31] C. Gkantsidis and P. Rodriguez, “Network coding for large scale content distribution,” in *proceedings of IEEE Infocom, Miami, FL*, Mar 2005.
- [32] A. G. Dimakis, P. G. Godfrey, M. J. Wainwright, , and K. Ramchandran, “Network coding for distributed storage systems,” in *proceedings of IEEE INFOCOM, Anchorage, AK*, May.
- [33] M. Gjoka, C. Fragouli, P. Sattari, and A. Markopoulou, “Loss tomography in general topologies with network coding,” in *IEEE GLOBECOM*, Dec 2007.
- [34] T. A. I. Hou, Y. Tsai and I. Gupta, “Adapcode: Adaptive network coding for code updates in wireless sensor networks,,” in *27th IEEE Conference on Computer Communications (INFOCOM)*. Phoenix, AZ,, April 2008.
- [35] R. K. M. M. E. T. Ho, B. Leong and D. R. Karger, “Byzantine modification detection in multicast networks with randomized network coding,” *IEEE Trans. inform. Theory*, vol. 54, no. 6, p. 27982803, 2008.
- [36] E. Kehdi and B. Li, “Null keys: Limiting malicious attacks via null space properties of network coding,,” in *28th IEEE Conference on Computer Communications (INFOCOM)*. Rio de Janeiro, Brazil,, April 2009.
- [37] Y. Wu, P. Chou, and S.-Y. Kung, “Minimum-energy multicast in mobile ad hoc networks using network coding,” *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1906–1918, 2005.
- [38] A. Lehman and E. Lehman, “Complexity classification of network information flow problems,” in *Proceedings of ACM-SIAM SODA*, Jan 2004.
- [39] R. Dougherty, C. Freiling, and K. Zeger, “Insufficiency of linear coding in network information flow,” *IEEE Trans. on Information Theory*, vol. 51, no. 8, pp. 2745–2759, 2005.
- [40] T. Ho, Y. Chang, and K. J. Han, “On constructive network coding for multiple unicasts,” in *44th Allerton Conf., Monticello, IL*, Sept 2006.
- [41] A. Eryilmaz and D. S. Lun, “Control for inter-session network coding,” in *Proc. of Workshop on Network Coding, Theory, & Applications (NetCod)*, San Diego, Jan 2007.
- [42] A. H. Mohsenian-Rad, J. Huang, V. Wong, S. Jaggi, and R. Schober, “A game-theoretic analysis of inter-session network coding,” in *Proc. IEEE Int’l Conf. Comm. (ICC)*, Germany, June 2009.
- [43] M. Kim, M. Medard, U. O’Reilly, and D. Traskov, “An evolutionary approach to inter-session network coding,” in *IEEE Conference on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, April 2009.

- [44] S. Omiwade, R. Zheng, and C. Hua, "Practical localized network coding in wireless mesh networks," in *IEEE Commun. Society Conf. Sensor & Ad Hoc Commun. & Networks (SECON) San Francisco, USA*, June 2008.
- [45] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.
- [46] S. Low and D. Lapsley, "Optimization flow control-I: basic algorithm and convergence," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 861–874, 1999.
- [47] S. Low, "A duality model of tcp and queue management algorithms," *IEEE/ACM Trans. Networking*, vol. 11, no. 4, pp. 525–536, 2003.
- [48] S. Kunniyur and R. Srikant, "End-to-end congestion control: utility functions, random losses and ecn marks," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 689–702, 2003.
- [49] S. Low and R. Srikant, "A mathematical framework for designing a low-loss low-delay internet," *Network and Spatial Economics*, vol. 4, no. 1, pp. 75–102, 2004.
- [50] X. Lin and N. Shroff, "Utility maximization for communication networks with multi-path routing," *IEEE Trans. Automat. Contr.*, vol. 51, no. 5, pp. 766–781, 2006.
- [51] W. Wang, M. Palaniswami, and S. H. Low, "Optimal flow control and routing in multi-path networks," *Performance Evaluation*, vol. 52, no. 2, 2003.
- [52] S. Deb and R. Srikant, "Congestion control for fair resource allocation in networks with multicast flows," *IEEE Trans. on Networking*, pp. 274–285, 2004.
- [53] X. Liu, E. Chong, and N. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer Networks*, vol. 41, no. 4, pp. 451–474, 2003.
- [54] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *IEEE/ACM Trans. Networking*, vol. 14, no. 2, pp. 302–315, 2006.
- [55] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1452–1463, 2006.
- [56] M. Chiang, S. H. Low, R. A. Calderbank, and J. C. Doyle, "Layering as optimization decomposition," *Proceedings of IEEE*, vol. 95, no. 1, pp. 255–312, 2007.
- [57] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu, "Joint congestion control and distributed scheduling in multihop wireless networks with a nodeexclusive interference model," in *27th IEEE Conference on Computer Communications (INFOCOM). Barcelona, Spain*, April 2006.
- [58] X. Wang and K. Kar, "Cross-layer rate control for end-to-end proportional fairness in wireless networks with random access," in *ACM Mobihoc, Urbana-Champaign, IL*, May 2005.

- [59] M. Neely, E. Modiano, and C. Li, “Fairness and optimal stochastic control for heterogeneous networks,” in *proceedings of IEEE Infocom, Miami, FL*, Mar 2005.
- [60] R. Bhatia and M. Kodialam, “On power efficient communication over multi-hop wireless networks: Joint routing, scheduling and power control,” in *proceedings of IEEE INFOCOM*, Mar 2004.
- [61] L. Lin, X. Lin, and N. B. Shroff, “Low-complexity and distributed energy minimization in multi-hop wireless networks,” in *proceedings of IEEE INFOCOM*, May 2007.
- [62] R. Cruz and A. Santhanam, “Optimal routing, link scheduling and power control in multi-hop wireless networks,” in *22nd IEEE Conference on Computer Communications (INFOCOM). San Francisco, USA*, April 2003.
- [63] D. Bertsekas and J. N. Tsitsikalis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.
- [64] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex Analysis and optimization*. Athena Scientific, 2003.
- [65] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [66] C.-C. Wang and N. B. Shroff, “Beyond the butterfly – a graph-theoretic characterization of the feasibility of network coding with two simple unicast sessions,” in *Proc. IEEE Int’l Symp. Inform. Theory*, June 2007.
- [67] C.-C. Wang, A. Khreishah, and N. Shroff, “On characterizing the throughput degradation for network coding with two sessions,” in *Mathematical Programming Society International Conference on Continuous Optimization, Hamilton, ON, Canada*, August 2007.
- [68] T. Bonald and L. Massoulié, “Impact of fairness on Internet performance,” in *Proc. of ACM Joint International Conference on Measurement and Modeling of Computer Systems (Sigmetrics)*, Cambridge, MA, June 2001.
- [69] J. Mo and J. Walrand, “Fair end-to-end window-based congestion control,” *IEEE/ACM Trans. on Networking*, vol. 8, no. 5, pp. 556–567, 2000.
- [70] X. Lin and N. B. Shroff, “An optimization based approach for quality-of-service routing in high-bandwidth networks,” *IEEE/ACM Trans. on Networking*, vol. 14, no. 6, pp. 1348–1361, 2006.
- [71] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee, “Achieving minimum-cost multicast: A decentralized approach based on network coding,” in *Proc. of IEEE Conference on Computer Communications (INFOCOM), Miami, FL*, March 2005.
- [72] P. Gupta and T. Javidi, “Towards throughput and delay-optimal routing for wireless ad-hoc networks,” in *Asilomar Conference on Signals, Systems and Computers*, Nov 2007.

- [73] V. Subramanian and D. Leith, "Draining-time based scheduling algorithm," in *IEEE CDC*, Dec 2007.
- [74] L. Ying, S. Shakottai, and A. Reddy, "On combining shortest-path and back-pressure routing over multihop wireless networks," in *IEEE Conference on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, April 2009.
- [75] S. Biswas and R. Morris., "Opportunistic routing in multi-hop wireless networks," in *Proc. ACM Special Interest Group on Data Commun. (SIGCOMM)*, Philadelphia, PA, USA, Sept 2005.
- [76] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: The case for coding-aware routing," in *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, Anocharage, AK, May 2007.
- [77] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *ACM Int'l. Conf. on Mobile Computing & Networking (MobiCom) Montréal, Canada,*, Sep 2007.
- [78] C.-C. Wang and N. B. Shroff, "Intersession network coding for two simple multicast sessions," in *45th Allerton Conf., Monticello, IL*, Sep 2007.
- [79] X. Lin and N. B. Shroff, "Utility maximization for communication networks with multi-path routing," *IEEE Trans on Automatic Control*, vol. 51, pp. 766–781, may 2006.
- [80] Y. Xi and E. M. Yeh, "Distributed algorithms for minimum cost multicast with network coding in wireless networks," in *Second Workshop on Network Coding, Theory, and Applications, Boston, MA*, April 2006.
- [81] Y. Xi and E. M. Yeh, "Distributed algorithms for minimum cost multicast with network coding," in *43rd Allerton Conf., Monticello, IL*, Sep 2005.
- [82] D. Traskov, N. Ratnakar, D. S. Lun, R. Koetter, and M. Medard, "Network coding for multiple unicasts: An approach based on linear optimization," in *Proc. IEEE Int'l Symp. Inform. Theory*, 2006.
- [83] J. Chen, P. Drushel, and D. Subramanian, "An efficient multi-path forwarding method," in *IEEE Conference on Computer Communications (INFOCOM)*, March 1998.
- [84] W. T. Zaumen and J. J. Garcia-Luna-Aceves, "Loop-free multipath routing using generalized diffusing computations," in *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, March 1998.
- [85] P. Chou, Y. Wu, , and K. Jain, "Practical network coding," in *Proc. of 41st Allerton Conf., Monticello, IL*, October 2003.
- [86] J. W. Lee, R. R. Mazumdar, and N. B. Shroff, "Non-convex optimization and rate control for multi-class services in the Internet," *IEEE/ACM Trans. on Networking*, vol. 13, pp. 827–840, Aug 2005.
- [87] J. Price and T. Javidi, "Network coding for resource redistribution in a unicast network," in *45th Annual Allerton Conf. on Comm., Contr., and Computing Monticello, IL,*, Sep 2007.

- [88] J. Le, J. Lui, and D. Chiu, “How many packets can we encode? — an analysis of practical wireless network coding,” in *27th IEEE Conference on Computer Communications (INFOCOM)*. Phoenix, USA., April 2008.
- [89] Y. Wu, P. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, “Network planning in wireless ad hoc networks: a cross-layer approach,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 136–150, 2005.
- [90] X. Lin and N. Shroff, “The impact of imperfect scheduling on cross-layer congestion control in wireless networks,” in *IEEE CDC, Paradise Island, Bahamas*, Dec 2004.
- [91] A. Khreishah, C.-C. Wang, and N. Shroff, “Rate control with pairwise intersession network coding,” *IEEE/ACM Trans. Networking*, submitted and under review, 2008.
- [92] A. Khreishah, C.-C. Wang, and N. B. Shroff, “Optimization based rate control for communication networks with inter-session network coding,” in *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, Phoenix, AZ, April 2008.
- [93] D. Bertsekas, A. Nedić, and A. Ozdaglar, *Convex Analysis and optimization*. Athena Scientific, 2003.
- [94] A. Khreishah, C.-C. Wang, and N. Shroff, “Cross-layer optimization for wireless multihop networks with pairwise intersession network coding,” tech. rep., Purdue university, 2008.
- [95] D. Baker, J. Wieselthier, and A. Ephremides, “Distributed algorithm for scheduling the activation of links in a self-organizing mobile radio network,” in *IEEE Int’l Conf. Commun.*, 1982.
- [96] B. Hajek and G. Sasaki, “Link scheduling in polynomial time,” *IEEE Trans. Inform. Theory*, vol. 34, no. 5, 1988.
- [97] B. Miller and C. Bisdikian, *Bluetooth Revealed: The Insider Guide to an Open Specification for Global Wireless Communications*. Prentice Hall, 2000.
- [98] C. Joo, X. Lin, and N. Shroff, “Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks,” in *27th IEEE Conference on Computer Communications (INFOCOM)*. Phoenix, USA., April 2008.
- [99] G. Zussman, A. Brzezinski, and E. Modiano, “Multihop local pooling for distributed throughput maximization in wireless networks,” in *27th IEEE Conference on Computer Communications (INFOCOM)*. Phoenix, USA., April 2008.
- [100] Y. Wu, P. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, “Network planning in wireless ad hoc networks: a cross-layer approach,” *IEEE JSAC*, vol. 23, 2005.
- [101] A. Khreishah, C.-C. Wang, and N. Shroff, “Cross-layer optimization for wireless multihop networks with pairwise intersession network coding,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 606–621, 2009.

- [102] G. Sharma, N. B. Shroff, and R. R. Mazumdar, "On the complexity of scheduling in wireless networks," in *proceedings of ACM MOBICOM*, Sept 2006.
- [103] Y. Wu, "Broadcasting when receivers know some messages a priori," in *Proc. IEEE Intl Symp. Inform. Theory. Nice, France,*, June 2007.
- [104] C.-C. Wang and N. Shroff, "On the capacity region of 1-hop intersession network coding - a broadcast packet erasure channel analysis with general message side information," in *Submitted to ISIT,*, 2010.
- [105] L. Georgiadis and L. Tassiulas, "Broadcast erasure channel with feedback - capacity and algorithms," in *Proc. 5th Workshop on Network Coding, Theory, & Applications (NetCod). Lausanne, Switzerland,*, June 2009.
- [106] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [107] M. J. Neely and R. Urgaonkar, "Optimal backpressure routing in wireless networks with multi-receiver diversity," *Ad Hoc Networks (Elsevier)*, vol. 7, no. 5, pp. 862–881, 2009.
- [108] D. Koutsonikolas, C.-C. Wang, and Y. Hu, "Ccaack: Efficient network coding based opportunistic routing through cumulative coded acknowledgments," in *in Proceedings of the 29th Conference on Computer Communications (INFOCOM), San Diego, USA,*, March 2010.
- [109] X. Zhang and B. Li, "Dice: a game theoretic framework for wireless multipath network coding," in *in the Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2008), Hong Kong, China,*, May 2008.
- [110] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *ACM Int'l. Conf. on Mobile Computing & Networking (MobiCom) San Diego, USA,*, Sep 2003.
- [111] X. Lin and S. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," *IEEE Trans. Automat. Contr.*, vol. 54, no. 2, p. 231242, 2009.
- [112] P. Marbach, A. Eryilmaz, and A. Ozdaglar, "Achievable rate region of csma schedulers in wireless networks with primary interference constraints," in *IEEE CDC, New Orleans, LA,* Dec 2007.
- [113] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," in *proceedings of IEEE INFOCOM*, May 2007.
- [114] C. Joo and N. B. Shroff, "Performance of random access scheduling schemes in multi-hop wireless networks," in *proceedings of IEEE INFOCOM*, May 2007.
- [115] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proc. of ACM Joint International Conference on Measurement and Modeling of Computer Systems (Sigmetrics), Saint-Malo, France,* June 2006.

- [116] S. Sanghavi, L. Bui, and R. Srikant, “Distributed link scheduling with constant overhead,” in *Proc. of ACM Joint International Conference on Measurement and Modeling of Computer Systems (Sigmetrics)*, San Diego, CA, June 2007.
- [117] L. Jiang and J. Walrand, “A distributed csma algorithm for throughput and utility maximization in wireless networks,” in *45th Annual Allerton Conf. on Comm., Contr., and Computing Monticello, IL*, Sep 2008.
- [118] R. Diestel, *Graph Theory*. New York: Springer-Verlag, 2005.

APPENDICES

A. SUPPORTING RESULTS FOR CHAPTER 2

A.1 Notations used for the proof of Proposition 2.4.1

Let

$$V(\vec{x}) = \begin{cases} \sum_{i=1}^N U_i(\sum_{k=1}^{|\mathcal{P}_i|} x_i^k + \sum_{j=i} \sum_{l=1}^{|\mathcal{P}ICC_{i,j}|} x_{ij}^l) & \vec{x} \geq 0 \\ -\infty & \text{else.} \end{cases}$$

Then V is the extended concave objective function, and we can write our problem in the following matrix form:

$$\max V(\vec{x})$$

subject to: $\mathbf{A}\vec{x} \leq \vec{C}$ and $\mathbf{B}\vec{x} = 0$, and the Lagrangian can be written as:

$$L(\vec{x}, \vec{\lambda}, \vec{\mu}, \vec{y}) = V(\vec{x}) - \vec{x}^T \begin{bmatrix} \mathbf{A}^T \mathbf{B}^T \end{bmatrix} \begin{bmatrix} \vec{\lambda} \\ \vec{\mu} \end{bmatrix} - \frac{1}{2}(\vec{x} - \vec{y})^T \mathbf{C}(\vec{x} - \vec{y}), \quad (\text{A.1})$$

where \mathbf{A} , \mathbf{B} , and \mathbf{C} are constructed as follows. Let $M(i) = \sum_{j:i=j} |\mathcal{P}ICC_{i,j}|$, \mathbf{A} is a matrix with $|E|$ rows and $(\sum_i |\mathcal{P}_i| + 2M(i))$ columns, where the e -th row is filled with $H_i^k(e)$ and $H_{ij}^l(e)$ in the same order as the corresponding x_i^k and x_{ij}^l appear in \vec{x} . Matrix \mathbf{B} is defined as a matrix with $\sum_i M(i)$ rows and $(\sum_i |\mathcal{P}_i| + 2M(i))$ columns, such that the (a, b) entry in \mathbf{B} is 1 if the b -th entry in \vec{x} is a variable of the form x_{ij}^l and the a -th entry in $\vec{\mu}$ is μ_{ij}^l . The (a, b) entry in \mathbf{B} is -1 if the b -th entry in \vec{x} is a variable of the form x_i^k and the a -th entry in $\vec{\mu}$ is μ_{ji}^k . For all other cases, the (a, b) entry in \mathbf{B} is 0. We define \mathbf{C} as a diagonal matrix of size $(\sum_i |\mathcal{P}_i| + 2M(i))$. If the a -th variable of \vec{x} is x_i^k or x_{ij}^l for some i , the a -th diagonal elements of \mathbf{C} will be α_i .

Throughout the proof we use the following norm $\|A\|_{\mathbf{D}} = A^T \mathbf{D}^{-1} A$, where $\mathbf{D} = \begin{bmatrix} \mathbf{E} & 0 \\ 0 & \mathbf{F} \end{bmatrix}$. Here, \mathbf{E} is an $|E|$ elements diagonal matrix with the e -th entry being β_e .

Matrix \mathbf{F} is another $(\sum_i M(i))$ diagonal matrix, in which the a -th diagonal element is β_{ij}^l where i, j, l are the same indices of the a -th element of $\vec{\mu}$.

A.2 Lemma A.2.1

Lemma A.2.1 Fix \vec{y} . Let $\begin{bmatrix} \vec{\lambda}_1 \\ \vec{\mu}_1 \end{bmatrix}$ and $\begin{bmatrix} \vec{\lambda}_2 \\ \vec{\mu}_2 \end{bmatrix}$ be two implicit cost vectors and let \vec{x}_1^* and \vec{x}_2^* be the corresponding maximizers of the Lagrangian, then:

$$\begin{bmatrix} (\vec{\lambda}_1 - \vec{\lambda}_2)^T & (\vec{\mu}_1 - \vec{\mu}_2)^T \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} (\vec{x}_1^* - \vec{x}_2^*) \leq -(\vec{x}_1^* - \vec{x}_2^*)^T \mathbf{C} (\vec{x}_1^* - \vec{x}_2^*).$$

Proof Let $\vec{x}_0^* = \arg \max_{\vec{x}} L(\vec{x}, \vec{\lambda}, \vec{\mu}, \vec{y})$. By taking the subgradient of (A.1) with respect to \vec{x} , we can see that there must exist a subgradient of (A.1) at \vec{x}_0^* such that:

$$\nabla V(\vec{x}_0^*) - \begin{bmatrix} \mathbf{A}^T & \mathbf{B}^T \end{bmatrix} \begin{bmatrix} \vec{\lambda} \\ \vec{\mu} \end{bmatrix} - \mathbf{C}(\vec{x}_0^* - \vec{y}) = 0. \quad (\text{A.2})$$

Substituting $\begin{bmatrix} \vec{\lambda} \\ \vec{\mu} \end{bmatrix}$ in (A.2) by $\begin{bmatrix} \vec{\lambda}_1 \\ \vec{\mu}_1 \end{bmatrix}$ and $\begin{bmatrix} \vec{\lambda}_2 \\ \vec{\mu}_2 \end{bmatrix}$, respectively, and taking the difference, we have:

$$\begin{bmatrix} \mathbf{A}^T & \mathbf{B}^T \end{bmatrix} \begin{bmatrix} \vec{\lambda}_1 - \vec{\lambda}_2 \\ \vec{\mu}_1 - \vec{\mu}_2 \end{bmatrix} = \left[\nabla V(\vec{x}_1^*) - \nabla V(\vec{x}_2^*) \right] - \mathbf{C}(\vec{x}_1^* - \vec{x}_2^*).$$

Since V is concave we have:

$$\left[\nabla V(\vec{x}_1^*) - \nabla V(\vec{x}_2^*) \right]^T (\vec{x}_1^* - \vec{x}_2^*) \leq 0.$$

Hence

$$\begin{aligned} & \begin{bmatrix} (\vec{\lambda}_1 - \vec{\lambda}_2)^T & (\vec{\mu}_1 - \vec{\mu}_2)^T \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} (\vec{x}_1^* - \vec{x}_2^*) \\ &= \left[\nabla V(\vec{x}_1^*) - \nabla V(\vec{x}_2^*) \right]^T (\vec{x}_1^* - \vec{x}_2^*) \\ &\quad - (\vec{x}_1^* - \vec{x}_2^*)^T \mathbf{C} (\vec{x}_1^* - \vec{x}_2^*) \\ &\leq -(\vec{x}_1^* - \vec{x}_2^*)^T \mathbf{C} (\vec{x}_1^* - \vec{x}_2^*). \end{aligned}$$



A.3 Lemma A.3.1

Lemma A.3.1 Let $\mathcal{L} = \sum_e \left(\sum_{i=1}^N \sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e) + \sum_{(i,j):i=j} \sum_{l=1}^{|\mathcal{P}^{IC}c_{ij}|} (H_{ij}^l(e))^2 \right)$. The sufficient condition for $2\mathbf{C} - \mathbf{A}^T\mathbf{E}\mathbf{A} - \mathbf{B}^T\mathbf{F}\mathbf{B}$ to be positive definite is that the step sizes β_e, β_{ij}^l fall in the following region:

$$(\mathcal{L} \cdot \max_e \beta_e + 2 \max_{i,j,l} \beta_{ij}^l) < 2 \min_i (\alpha_i)$$

Proof If $2\mathbf{C} - \mathbf{A}^T\mathbf{E}\mathbf{A} - \mathbf{B}^T\mathbf{F}\mathbf{B}$ is positive definite, then

$$\vec{\delta x}^T (2\mathbf{C} - \mathbf{A}^T\mathbf{E}\mathbf{A} - \mathbf{B}^T\mathbf{F}\mathbf{B}) \vec{\delta x} > 0,$$

for all nonzero column vectors $\vec{\delta x}$, which is equivalent to

$$2\vec{\delta x}^T \mathbf{C} \vec{\delta x} > \vec{\delta x}^T (\mathbf{A}^T\mathbf{E}\mathbf{A} + \mathbf{B}^T\mathbf{F}\mathbf{B}) \vec{\delta x}.$$

We use δx_i to refer to the i -th element of $\vec{\delta x}$. By the Cauchy-Schwartz Inequality, we have

$$\vec{\delta x}^T (\mathbf{A}^T\mathbf{E}\mathbf{A}) \vec{\delta x} + \vec{\delta x}^T (\mathbf{B}^T\mathbf{F}\mathbf{B}) \vec{\delta x}$$

$$\begin{aligned}
&= \beta_e \left(\sum_{i,k} H_i^k(e) \delta x_i^k \right. \\
&\quad \left. + \sum_{(i,j):i=j} \sum_{l=1}^l (H_{ij}^l(e)) \delta x_{ij}^l \right)^2 \\
&\quad + \sum_{(i,j):i<j} \beta_{ij}^l [(\delta x_{ij}^l - \delta x_{ji}^l)^2] \\
&\leq \beta_e \left(\sum_{i,k} H_i^k(e) \right. \\
&\quad \left. + \sum_{(i,j):i=j} \sum_{l=1}^l (H_{ij}^l(e))^2 \right) \delta x_i^2 \\
&\quad + \sum_{(i,j):i<j} \beta_{ij}^l (2\delta x_{ij}^l)^2 + 2(\delta x_{ji}^l)^2 \\
&\leq \max_e \beta_e \left(\sum_{i,k} H_i^k(e) \right. \\
&\quad \left. + \sum_{(i,j):i=j} \sum_{l=1}^l (H_{ij}^l(e))^2 \right) \delta x_i^2 \\
&\quad + (\max_{i,j,l} \beta_{ij}^l) 2 \delta x_i^2 \\
&= (\mathcal{L} \cdot \max_e \beta_e + 2 \max_{i,j,l} \beta_{ij}^l) \delta x_i^2
\end{aligned}$$

Therefore if the inequality that

$$(\mathcal{L} \cdot \max_e \beta_e + 2 \max_{i,j,l} \beta_{ij}^l) \delta x_i^2 < 2 \min_i (\alpha_i) \delta x_i^2$$

holds, then we have

$$2\vec{\delta x}^T \mathbf{C} \vec{\delta x} > \vec{\delta x}^T (\mathbf{A}^T \mathbf{E} \mathbf{A} + \mathbf{B}^T \mathbf{F} \mathbf{B}) \vec{\delta x}$$

which in turn implies that $2\mathbf{C} - \mathbf{A}^T \mathbf{E} \mathbf{A} - \mathbf{B}^T \mathbf{F} \mathbf{B}$ is positive definite. The proof is complete. ■

A.4 Proof of *Proposition 2.4.1*

In this section, we prove Proposition 2.4.1 the convergence of Algorithm A for the case that K tends to infinity first and then let the number of iterations goes to infinity.

Proof We will prove the convergence of Algorithm A when $K \rightarrow \infty$. To do so, we will prove the convergence of the first step during the proximal iteration. The convergence of the whole algorithm follows from [63] page 233. Fix $\vec{y}(t)$. Let $\vec{\lambda}_0, \vec{\mu}_0$ be a stationary point of (2.7) and (2.9), and let \vec{x}_0 be the corresponding primal variable. \vec{x}_0 is unique since $L(\cdot)$ is strictly concave with respect to \vec{x} , and $\vec{x}_0 = \arg \max_{\vec{x}} L(\vec{x}, \vec{\lambda}, \vec{\mu}, \vec{y})$. By the projection theorem in [63] page 211 we have,

$$\begin{aligned}
& \left\| \begin{bmatrix} \vec{\lambda}(t, \kappa + 1) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa + 1) - \vec{\mu}_0 \end{bmatrix} \right\|_{\mathbf{D}} \\
&= \left\| \begin{bmatrix} [\vec{\lambda}(t, \kappa) + \mathbf{E}(\mathbf{A}\vec{x}(t, \kappa) - \vec{C})]^+ - [\vec{\lambda}_0 + \mathbf{E}(\mathbf{A}\vec{x}_0 - \vec{C})]^+ \\ \vec{\mu}(t, \kappa) + \mathbf{FB}\vec{x}(t, \kappa) - \vec{\mu}_0 + \mathbf{FB}\vec{x}_0 \end{bmatrix} \right\|_{\mathbf{D}} \\
&\leq \left\| \begin{bmatrix} \vec{\lambda}(t, \kappa) + \mathbf{E}(\mathbf{A}\vec{x}(t, \kappa) - \vec{C}) - \vec{\lambda}_0 + \mathbf{E}(\mathbf{A}\vec{x}_0 - \vec{C}) \\ \vec{\mu}(t, \kappa) + \mathbf{FB}\vec{x}(t, \kappa) - \vec{\mu}_0 + \mathbf{FB}\vec{x}_0 \end{bmatrix} \right\|_{\mathbf{D}} \\
&= \left\| \begin{bmatrix} \vec{\lambda}(t, \kappa) - \vec{\lambda}_0 + \mathbf{EA}(\vec{x}(t, \kappa) - \vec{x}_0) \\ \vec{\mu}(t, \kappa) - \vec{\mu}_0 + \mathbf{FB}(\vec{x}(t, \kappa) - \vec{x}_0) \end{bmatrix} \right\|_{\mathbf{D}}.
\end{aligned}$$

This gives:

$$\left\| \begin{bmatrix} \vec{\lambda}(t, \kappa + 1) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa + 1) - \vec{\mu}_0 \end{bmatrix} \right\|_{\mathbf{D}}$$

$$\begin{aligned}
&\leq \left\| \begin{bmatrix} \vec{\lambda}(t, \kappa) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa) - \vec{\mu}_0 \end{bmatrix} \right\|_{\mathbf{D}} + \begin{bmatrix} \mathbf{EA}(\vec{x}(t, \kappa) - \vec{x}_0) \\ \mathbf{FB}(\vec{x}(t, \kappa) - \vec{x}_0) \end{bmatrix}^T \\
&\quad \mathbf{D}^{-1} \begin{bmatrix} \mathbf{EA}(\vec{x}(t, \kappa) - \vec{x}_0) \\ \mathbf{FB}(\vec{x}(t, \kappa) - \vec{x}_0) \end{bmatrix} \\
&\quad + 2 \begin{bmatrix} \vec{\lambda}(t, \kappa) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa) - \vec{\mu}_0 \end{bmatrix}^T \mathbf{D}^{-1} \begin{bmatrix} \mathbf{EA}(\vec{x}(t, \kappa) - \vec{x}_0) \\ \mathbf{FB}(\vec{x}(t, \kappa) - \vec{x}_0) \end{bmatrix} \\
&= \left\| \begin{bmatrix} \vec{\lambda}(t, \kappa) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa) - \vec{\mu}_0 \end{bmatrix} \right\|_{\mathbf{D}} + (\vec{x}(t, \kappa) - \vec{x}_0)^T \begin{bmatrix} \mathbf{A}^T & \mathbf{B}^T \end{bmatrix} \\
&\quad \begin{bmatrix} \mathbf{E}^T & 0 \\ 0 & \mathbf{F}^T \end{bmatrix} \mathbf{D}^{-1} \begin{bmatrix} \mathbf{E} & 0 \\ 0 & \mathbf{F} \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} (\vec{x}(t, \kappa) - \vec{x}_0) \\
&\quad + 2 \begin{bmatrix} \vec{\lambda}(t, \kappa) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa) - \vec{\mu}_0 \end{bmatrix}^T \\
&\quad \mathbf{D}^{-1} \begin{bmatrix} \mathbf{E} & 0 \\ 0 & \mathbf{F} \end{bmatrix} \begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{B} \end{bmatrix} (\vec{x}(t, \kappa) - \vec{x}_0) \\
&= \left\| \begin{bmatrix} \vec{\lambda}(t, \kappa) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa) - \vec{\mu}_0 \end{bmatrix} \right\|_{\mathbf{D}} + (\vec{x}(t, \kappa) - \vec{x}_0)^T \begin{bmatrix} \mathbf{A}^T & \mathbf{B}^T \end{bmatrix} \\
&\quad \mathbf{D} \mathbf{D}^{-1} \mathbf{D} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} (\vec{x}(t, \kappa) - \vec{x}_0) \\
&\quad + 2 \begin{bmatrix} \vec{\lambda}(t, \kappa) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa) - \vec{\mu}_0 \end{bmatrix}^T \mathbf{D}^{-1} \mathbf{D} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} (\vec{x}(t, \kappa) - \vec{x}_0)
\end{aligned}$$

By Lemma A.2.1 we have:

$$\left\| \begin{bmatrix} \vec{\lambda}(t, \kappa + 1) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa + 1) - \vec{\mu}_0 \end{bmatrix} \right\|_{\mathbf{D}}$$

$$\begin{aligned}
&\leq \left\| \begin{bmatrix} \vec{\lambda}(t, \kappa) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa) - \vec{\mu}_0 \end{bmatrix} \right\|_{\mathbf{D}} \\
&\quad + (\vec{x}(t, \kappa) - \vec{x}_0)^T \begin{bmatrix} \mathbf{A}^T & \mathbf{B}^T \end{bmatrix} \mathbf{D} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} (\vec{x}(t, \kappa) - \vec{x}_0) \\
&\quad - 2(\vec{x}(t, \kappa) - \vec{x}_0)^T \mathbf{C} (\vec{x}(t, \kappa) - \vec{x}_0) \\
&= \left\| \begin{bmatrix} \vec{\lambda}(t, \kappa) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa) - \vec{\mu}_0 \end{bmatrix} \right\|_{\mathbf{D}} - (\vec{x}(t, \kappa) - \vec{x}_0)^T \mathbf{L} (\vec{x}(t, \kappa) - \vec{x}_0),
\end{aligned}$$

where $\mathbf{L} = 2\mathbf{C} - \begin{bmatrix} \mathbf{A}^T & \mathbf{B}^T \end{bmatrix} \mathbf{D} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$. When \mathbf{L} is positive definite, we have:

$$\left\| \begin{bmatrix} \vec{\lambda}(t, \kappa + 1) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa + 1) - \vec{\mu}_0 \end{bmatrix} \right\|_{\mathbf{D}} \leq \left\| \begin{bmatrix} \vec{\lambda}(t, \kappa) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa) - \vec{\mu}_0 \end{bmatrix} \right\|_{\mathbf{D}}. \quad (\text{A.3})$$

Therefore, if the step sizes β_e and $\beta_{i_j}^l$ satisfy the condition in Proposition 2.4.1, then by Lemma A.3.1, \mathbf{L} is positive definite. Accordingly $\left\| \begin{bmatrix} \vec{\lambda}(t, \kappa) - \vec{\lambda}_0 \\ \vec{\mu}(t, \kappa) - \vec{\mu}_0 \end{bmatrix} \right\|_{\mathbf{D}}$ will be a nonnegative and decreasing sequence. Therefore, as $K \rightarrow \infty$, $\vec{x}(t, K) \rightarrow \vec{x}_0$. ■

A.5 Proposition A.5.1

Recall the following theorem from [78].

Theorem A.5.1 *Define two sets of integral graphs, \mathcal{G}_b and \mathcal{G}_g , as follows.*

1. \mathcal{G}_b contains the full butterfly as described in Fig. 1.3(a), and all graphs obtained from the full butterfly via edge contraction. See [118] for the definition of edge contraction and subdivision.
2. \mathcal{G}_g contains the full grail as described in Fig. 1.4(a), and all graphs obtained from the full grail via edge contraction.

Suppose there exists a network coding solution to the two unicast-session problem. Then one of the following two conditions must hold.

- There exist two EDPs connecting (s_1, d_1) and (s_2, d_2) .
- G'' contains an integral subgraph $F = (V^F, E^F)$ such that (i) $\{s_1, s_2, d_1, d_2\} \in V^F$ and (ii) there exists a $G_q \in \mathcal{G}_b \cup \mathcal{G}_g$ such that F is a subdivision of G_q . Namely, F can be obtained from G_q by replacing each edge of G_q with an interior-vertex-disjoint path, also known as an independent path.

Proposition A.5.1 For F as defined in Theorem A.5.1, if F contains two edges that connect the same pair of vertices, then F contains 2 EDPs connecting (s_1, d_1) and (s_2, d_2)

Proof F is a subdivision of G_q . If F contains two edges that connect the same pair of vertices then so does G_q . It is easy to check that for all subgraphs in \mathcal{G}_b or in \mathcal{G}_g , if there are two edges connecting the same pair of vertices, there exist two edge-disjoint paths connecting (s_1, d_1) and (s_2, d_2) . The proof is complete. ■

A.6 Proof of Proposition 2.6.1

Proof For any PICC if there exist 2EDPs between (s_1, s_2) and (s_1, s_2) , we can send packets through these paths and achieve the required rate without network coding, which means that the PICC of interest is redundant. We exploit this observation in the following to show that a PICC is redundant.

If G'' is declared “redundant” by rule 2, we have 2EDPs between (s_1, d_1) and (s_2, d_2) , because of the disjointness of $E(P_{s_1, d_1}) \cup E(Q_{s_1, d_1})$ and $E(P_{s_2, d_2}) \cup E(Q_{s_2, d_2})$. In the following, we will prove rule 3. Without loss of generality, we can use the integral graph G'' to represent the l -th PICC between sessions i and j . G'' always satisfy condition 2 of Proposition 1.3.1. If condition 1 is satisfied, there is no need to include G'' in the optimization problem because there are 2 EDPs. Assume condition 1 is not satisfied. We have two cases. Case 1: F in Theorem A.5.1 is the same as G'' .

By (2.1), if $H_{ij}^l(e) = 1$, then link e is modelled as two edges connecting the same pair of vertices. G'' contains 2 EDPs by Proposition A.5.1. Case 2: F is a proper subgraph of G'' . The edges and vertices in F are subsets of the edges and vertices in G'' , and F satisfies the necessary and sufficient conditions for pairwise linear network coding. Therefore, the same solution in G'' can be achieved in F by consuming fewer resources and hence G'' is a redundant PICC. Rule 1 follows by using the same technique. ■

B. SUPPORTING RESULTS FOR CHAPTER 3

B.1 Sketches of the proof of Proposition 3.5.1:

Proof We first notice that by (3.6), q_{ij}^l is a constant and is thus bounded away from infinity. For the following, choose the following Lyapunov function $V(\mathbf{q}) = \sum_e \frac{(q_e)^2}{2\beta_e}$ for the dual variables $\mathbf{q} = \{q_e : \forall e\}$. Then

$$V(\mathbf{q}[t+1]) - V(\mathbf{q}[t]) = \sum_e q_e[t] \left(\sum_{i,k} H_i^k(e)x_i^k + \frac{1}{2} \sum_{(i,j):i=j} H_{ij}^l x_{ij}^l - r_e \right) + \text{const}, \quad (\text{B.1})$$

where **const** is a constant bounded away from positive and negative infinity.

Since the fixed rate assignment \mathbf{x} is in the interior of $\gamma\Lambda$, the first term of (B.1) can be rewritten as

$$-\epsilon \sum_e q_e[t] \left(\sum_{i,k} H_i^k(e)x_i^k + \frac{1}{2} \sum_{(i,j):i=j} H_{ij}^l x_{ij}^l \right) \quad (\text{B.2})$$

for some $\epsilon > 0$. Combining (B.1) and (B.2), we have that any component of \mathbf{q} tends to infinity will lead to negative difference of the Lyapunov function $V(q)$. As a result, all dual variables q_e and q_{ij}^l are bounded away from infinity. Since the primal variables have only bounded domains, the proof is complete. \blacksquare

B.2 Sketches of the proof of Proposition 3.5.2:

Proof Let $\rho_i \triangleq \frac{\lambda_i}{\mu_i}$. Since $\{\rho_i\}_i$ is in the interior of $\gamma\Lambda$, by the definition of Λ , we can find ϵ , ρ_i^k and ρ_{ij}^l satisfying

$$\begin{aligned} \rho_i &= \rho_i^k + \rho_{ij}^l, \forall i, \rho_{ij}^l = \rho_{ji}^l, \forall i, j, l, \\ &\text{and } (1 + 2\epsilon) \sum_i H_i^k(e) \rho_i^k + \frac{1}{2} \sum_{j,l:j=i} H_{ij}^l(e) \rho_{ij}^l \leq \gamma r_e^* \\ &\text{for some } \mathbf{r}^* \in \text{Co}(\mathcal{R}). \end{aligned} \quad (\text{B.3})$$

Let n_i denote the number of users in the system. The probability law of n_i is determined by a Markov process. Its transition rate is given by:

$$\begin{cases} n_i[t] \rightarrow n_i[t] + 1 & \text{with rate } \lambda_i \\ n_i[t] \rightarrow n_i[t] - 1 & \text{with rate } \mu_i \left(\sum_k x_i^k[t] + \sum_{j,l:j=i} \sum_l x_{ij}^l[t] \right) n_i[t] \end{cases}$$

A heuristic fluidity model argument is provided as follows. By the rate and scheduling update rules, we have

$$\begin{aligned} \mathbf{x}_i[t] &= \arg \max_{\mathbf{x}} U_i \sum_k x_i^k + \sum_{j,l:j=i} x_{ij}^l - q_e[t] \sum_k H_i^k(e) x_i^k + \frac{1}{2} \sum_{j,l:j=i} H_{ij}^l(e) x_{ij}^l \\ &\quad - \sum_{j:j>i} q_{ij}^l[t] x_{ij}^l - \sum_{j:j<i} q_{ji}^l[t] x_{ij}^l \\ &\quad - \frac{\alpha_i}{2} \sum_k (x_i^k - y_i^k)^2 + \sum_{j,l:j=i} (x_{ij}^l - y_{ij}^l)^2 \\ &\quad q_e[t] r_e[t] \geq \gamma \max_{[r]} q_e[t] r_e \end{aligned} \quad (\text{B.4})$$

Therefore

$$\begin{aligned}
\frac{dV_n}{dt} &= - \epsilon \sum_i \left(K \kappa_i n_i + \alpha_i \rho_i \right) \sum_k y_i^k + \sum_{j,l:j=i} y_{ij}^l \\
&\quad + \sum_i \frac{K \kappa_i n_i}{\rho_i} + \alpha_i \sum_k y_i^k + \sum_{j,l:j=i} y_{ij}^l \\
&\quad (1 + \epsilon) \rho_i - n_i \sum_k x_i^k + \sum_{j,l:j=i} x_{ij}^l \\
&= - \epsilon \sum_i \left(K \kappa_i n_i + \alpha_i \rho_i \right) \sum_k y_i^k + \sum_{j,l:j=i} y_{ij}^l \\
&\quad + \sum_i \frac{K(1 + \epsilon) \kappa_i}{\sum_k x_i^k + \sum_{j,l:j=i} x_{ij}^l} \\
&\quad + \alpha_i \sum_k y_i^k + \sum_{j,l:j=i} y_{ij}^l \\
&\quad (1 + \epsilon) \rho_i - n_i \sum_k x_i^k + \sum_{j,l:j=i} x_{ij}^l \\
&\quad + (A),
\end{aligned}$$

where

$$\begin{aligned}
(A) &\triangleq - \sum_i K \kappa_i \frac{(1 + \epsilon)}{\sum_k x_i^k + \sum_{j,l:j=i} x_{ij}^l} - \frac{n_i}{\rho_i} \\
&\quad (1 + \epsilon) \rho_i - n_i \sum_k x_i^k + \sum_{j,l:j=i} x_{ij}^l \leq 0.
\end{aligned}$$

In the following we will use the following notation $y_{i(j)}^{k/l}$ to express that we are iterating over all y_i^k and y_{ij}^l . For example $\sum_{k,l,j:j=i} y_{i(j)}^{k/l} \triangleq \sum_{k=1}^{|\mathcal{P}_i|} y_i^k + \sum_{j:j=i} \sum_{l=1}^{|\mathcal{P}_{ICC_{ij}}|} y_{ij}^l$. Similar notation is used for other variables such as ρ and q . We define F_0 and F_1 such that:

$$\begin{aligned}
F_0 &= (1 + \epsilon) \left(\sum_i \alpha_i \sum_{k,j,l:j=i} y_{i(j)}^{k/l} \sum_{k',j',l':(k',j',l')=(k,j,l)} \rho_{i(j')}^{k'/l'} \right) \\
&\quad - \epsilon \sum_i \alpha_i \rho_i \sum_{k,j,l} y_{i(j)}^{k/l} + (A)
\end{aligned}$$

$F_1 = (1 + \epsilon) \sum_i \alpha_i M_{\Sigma x} \sum_{k,j,l:j=i} \rho_{i(j)}^{k/l}$. Here $M_{\Sigma x}$ is the maximum rate assigned to any user.

Since $\mathbf{x}_i[t]$ solves (B.4) and $U_i(\cdot) = \kappa_i \log(\cdot)$, there exist $\delta_i^k, \delta_{ij}^l \geq 0$ and $\delta_{i,M} \geq 0$ such that

$$\begin{aligned} & \frac{\kappa_i}{\sum_{k',j',l':j'=i} x_{i(j')}^{k'/l'}} - \alpha_i (x_{i(j)}^{k/l} - y_{i(j)}^{k/l}) + \delta_{i(j)}^{k/l} - \delta_{i,M} \\ & = q_{e,i(j)}^{k/l}, \quad i, j, k, l : i \neq j \end{aligned}$$

where

$$\begin{aligned} q_{e,i}^k &= q_e H_i^k(e) \\ q_{e,ij}^l &= \begin{cases} \frac{1}{2} \sum_e q_e H_{ij}^l(e) + q_{ij}^l & \text{if } i < j \\ \frac{1}{2} \sum_e q_e H_{ij}^l(e) - q_{ji}^l & \text{if } i > j. \end{cases} \end{aligned}$$

We have $\delta_{i(j)}^{k/l} x_{i(j)}^{k/l} = 0$ and $\delta_{i,M} \left(\sum_{k,j,l:j=i} x_{i(j)}^l - M_x \right) = 0$ due to the complementary slackness conditions. In our online technical report [94] for each i we have explicitly construct a positive integer J_i such that

$$\delta_{i,M} \leq \frac{\kappa_i}{M_{\Sigma x}} + \alpha_i \frac{\sum_{k,j,l:l=j} y_{i(j)}^{k/l}}{J_i}, \forall i. \quad (\text{B.5})$$

By choosing $F_2 = \max_i \left\{ \frac{\kappa_i}{M_{\Sigma x}} + \alpha_i \frac{\sum_{k,j,l:l=j} y_{i(j)}^{k/l}}{J_i} \right\}$, F_2 is an upper bound for all $\delta_{i,M}$. Let $K = \frac{1}{1+\epsilon}$, we have

$$\begin{aligned} \frac{dV_n}{dt} &\leq -\epsilon \sum_i K \kappa_i n_i \\ &+ \sum_e q_e (1 + \epsilon) \sum_i \sum_k H_i^k(e) \rho_i^k + \frac{1}{2} \sum_{j,l:j=i} H_{ij}^l(e) \rho_{ij}^l \\ &\quad - \sum_i \sum_k n_i H_i^k(e) x_i^k + \frac{1}{2} \sum_{j,l:j=i} n_i H_{ij}^l(e) x_{ij}^l \\ &+ \sum_{(i,j):i < j} \sum_l (q_{ij}^l (1 + \epsilon) (\rho_{ij}^l - \rho_{ji}^l) - q_{ij}^l (n_i x_{ij}^l - n_j x_{ji}^l)) \\ &+ F_0 + F_1 + F_2 (1 + \epsilon) \sum_i \sum_{k,j,l:j=i} \rho_{i(j)}^{k/l}. \end{aligned} \quad (\text{B.6})$$

The details of how (B.6) is obtained in our online technical report [94]. Since

$$\begin{aligned} \frac{dV_q(\mathbf{q})}{dt} = & \sum_e q_e \sum_i n_i \sum_k H_i^k(e) x_i^k + \frac{1}{2} \sum_{j,l:j=i} H_{ij}^l(e) x_{ij}^l - r_e \\ & + \sum_{(i,j):i<j} q_{ij}^l (n_i x_{ij}^l - n_j x_{ji}^l), \end{aligned}$$

and $\rho_{ij}^l = \rho_{ji}^l$, the overall drift $\frac{dV}{dt} = \frac{dV_n}{dt} + \frac{dV_q}{dt}$ becomes

$$\begin{aligned} \frac{dV}{dt} \leq & - \epsilon \sum_i K \kappa_i n_i - \epsilon \sum_e q_e \sum_i \sum_k H_i^k(e) \rho_i^k + \frac{1}{2} \sum_{j,l:j=i} H_{ij}^l(e) \rho_{ij}^l \\ & + F_0 + F_1 + F_2(1 + \epsilon) \sum_{i,k,j,l:j=i} \rho_{i(j)}^{k/l}. \end{aligned}$$

Here, we used (B.3). The Lyapunov function will have a negative drift and the system is stable. A full proof that takes into account the second-order variation can be obtained accordingly. ■

B.3 Proof of Proposition 3.6.1

Proof We prove this theorem by induction. We perform coding operations sequentially from the most upstream edges to the most downstream edges. Let M_e represent the symbol transmitted along edge e . We have the following induction hypothesis: If Case 1 is satisfied, then for any edge e we have:

$$M_e = \begin{cases} X \text{ or } X + Y & \text{if } e \in P_{s_1, d_1} \\ Y \text{ or } X + Y & \text{if } e \in P_{s_2, d_2} \\ X \text{ or } Y & \text{if } e \in P_{s_2, d_1} \text{ or } e \in Q_{s_2, d_2}. \end{cases} \quad (\text{B.7})$$

Case 2 is a symmetric version of Case 1 and the discussion is thus omitted.

If Case 3 is satisfied, for any edge e we have:

$$M_e = \begin{cases} X \text{ or } X + Y & \text{if } e \in P_{s_1, d_1} \\ Y \text{ or } X + Y & \text{if } e \in P_{s_2, d_2} \\ X \text{ or } Y & \text{if } e \in P_{s_2, d_1} \text{ or } e \in Q_{s_1, d_2} \end{cases} \quad (\text{B.8})$$

Both hypotheses (B.7), (B.8) are satisfied on the immediate outgoing edges of sources s_1 and s_2 which carry X and Y respectively. To show that the induction holds for all edges, we need to consider 13 scenarios when Case 1 is satisfied as in Table B.1 and 5 scenarios when Case 3 is satisfied as in Table B.2. The entries in the second column in both Table B.1 and B.2 represent the paths that share edge e and the entries in the third column represent the corresponding coding operation on edge e according to either (B.7) or (B.8).

Since Case 3 is simpler than Case 1, we discuss Case 3 first and then move on to Case 1.

For Case 3 we have $P_{s_1,d_1} = Q_{s_1,d_1}$, $P_{s_2,d_2} = Q_{s_2,d_2}$. Therefore, we have four distinct paths

P_{s_1,d_1} , P_{s_2,d_2} , P_{s_2,d_1} , and Q_{s_1,d_2} in the PICC. If an edge is shared by three paths, it is either the case that the PICC is insignificant by Rule 1 or condition 2 of Theorem 1.3.1 is not satisfied. There are $\binom{4}{2} = 6$ scenarios in which two paths meet at a single edge. Since by Rule 1, P_{s_2,d_1} and Q_{s_1,d_2} do not meet, we are left with 5 scenarios to consider as in Table B.2

In the following, we prove that the induction hypothesis follows for the 5 scenarios in Case 3.

Scenario 1 as in Table B.2 The paths that meet at edge e are P_{s_1,d_1} and P_{s_2,d_2} . The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be $X + Y$ according to the hypothesis, which is the intersection of the first two cases of (B.8). The coded symbol $X + Y$ will be forwarded and the invariant holds for the target edge e . If the symbols that enter edge v , the tail of e are different, then node v can decode both X and Y and compute the coded symbol $X + Y$ according to Case 3.4 (or equivalently Case 1.4) in Section 3.6 and send $X + Y$ along e . The hypothesis holds in this scenario that $e \in P_{s_1,d_1} \cap P_{s_2,d_2}$.

Scenario 2 and 3 as in Table B.2 The paths that meet at edge e are P_{s_1,d_1} and Q_{s_1,d_2} (P_{s_1,d_1} and P_{s_2,d_1}). The symbols carried by the paths on the respective

Table B.1

The list of possible coding operations a node has to perform if the PICC satisfies Case 1

PICC satisfies Case 1		
	Paths sharing edge e	Symbols transmitted on edge e
Scenario 1	$P_{s_2,d_1} P_{s_2,d_2}$	Y
Scenario 2	$P_{s_2,d_1} P_{s_1,d_1}$	X
Scenario 3	$P_{s_2,d_1} Q_{s_2,d_2}$	Y
Scenario 4	$P_{s_2,d_1} P_{s_2,d_2} Q_{s_2,d_2}$	Y
Scenario 5	$P_{s_2,d_1} P_{s_1,d_1} Q_{s_2,d_2}$	X
Scenario 6	$P_{s_1,d_1} P_{s_2,d_2}$	$X + Y$
Scenario 7	$P_{s_2,d_2} Q_{s_2,d_2}$	Y
Scenario 8	$Q_{s_1,d_2} P_{s_1,d_1}$	X or $X + Y$
Scenario 9	$Q_{s_1,d_2} P_{s_2,d_2}$	Y or $X + Y$
Scenario 10	$Q_{s_1,d_2} Q_{s_2,d_2}$	Y or $X + Y$
Scenario 11	$Q_{s_1,d_2} P_{s_1,d_1} P_{s_2,d_2}$	$X + Y$
Scenario 12	$Q_{s_1,d_2} P_{s_2,d_2} Q_{s_2,d_2}$	Y
Scenario 13	$P_{s_1,d_1} Q_{s_2,d_2}$	X

Table B.2

The list of possible coding operations a node has to perform if the PICC satisfies Case 3

PICC satisfies Case 3		
	Paths sharing edge e	Symbols transmitted on edge e
Scenario 1	$P_{s_1,d_1} P_{s_2,d_2}$	$X + Y$
Scenario 2	$P_{s_1,d_1} Q_{s_1,d_2}$	X
Scenario 3	$P_{s_1,d_1} P_{s_2,d_1}$	X
Scenario 4	$P_{s_2,d_2} Q_{s_1,d_2}$	Y
Scenario 5	$P_{s_2,d_2} P_{s_2,d_1}$	Y

previous edges can be the same or not. If they are the same, the symbols must be X according to the hypothesis, which is the intersection of the first and third cases of (B.8). The symbol X will be forwarded and the invariant holds for the target edge e . If the symbols that enter edge v , the tail of e are different, then node v can compute X according to Case 3.3 (or equivalently Case 1.3) in Section 3.6 and send X along e . The hypothesis holds in this scenario that $e \in P_{s_1,d_1} \cap Q_{s_1,d_2}$ ($e \in P_{s_1,d_1} \cap P_{s_2,d_1}$).

Scenario 4 and 5 as in Table B.2 The paths that meet at edge e are P_{s_2,d_2} and Q_{s_1,d_2} (P_{s_2,d_2} and P_{s_2,d_1}). The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be Y according to the hypothesis, which is the intersection of the last two cases of (B.8). The symbol Y will be forwarded and the invariant holds for the target edge e . If the symbols that enter edge v , the tail of e are different, then node v can compute the symbol Y according to Case 3.2 (or equivalently Case 1.2) in Section 3.6 and send Y along e . The hypothesis holds in this scenario that $e \in P_{s_2,d_2} \cap Q_{s_1,d_2}$ ($e \in P_{s_2,d_2} \cap P_{s_2,d_1}$).

For Case 1 we have $P_{s_1,d_1} = Q_{s_1,d_1}$. Therefore, we have five distinct paths $P_{s_1,d_1}, P_{s_2,d_2}, P_{s_2,d_1}, Q_{s_2,d_2}$, and Q_{s_1,d_2} in the PICC. By Rule 1 P_{s_2,d_1} and Q_{s_1,d_2} will not meet at a single edge. Also $P_{s_1,d_1}, P_{s_2,d_2}, Q_{s_2,d_2}$ will not meet at a single edge due to Rule 2.1. Therefore, any scenario in which an edge is used by five or four of the paths is impossible. We have $\binom{5}{3} = 10$ different scenarios in which edge e is used by three distinct paths. The scenarios in which an edge is used by $P_{s_1,d_1}, P_{s_2,d_2}, P_{s_2,d_1}$ or $P_{s_1,d_1}, Q_{s_2,d_2}, Q_{s_1,d_2}$ violate condition 2 of Theorem 1.3.1. The scenario in which an edge is used by $P_{s_1,d_1}, P_{s_2,d_2}, Q_{s_2,d_2}$, will be removed because it satisfies Rule 2.1. The scenarios in which an edge is used by $(P_{s_1,d_1}, P_{s_2,d_1}, Q_{s_1,d_2})$, $(P_{s_2,d_2}, Q_{s_2,d_1}, Q_{s_1,d_2})$, or $(P_{s_2,d_1}, Q_{s_2,d_2}, Q_{s_1,d_2})$ will be removed because it satisfies Rule 1. As a result, we need to only consider 4 scenarios in which e is used by three distinct paths (Scenarios 4, 5, 11, 12 in Table B.1). Since by Rule 1 P_{s_2,d_1} and Q_{s_1,d_2} do not use the same edge we have only 9 scenarios in which e is used by two paths. The total is 13 scenarios as in Table B.1.

In the following, we prove that the induction hypothesis holds for the 13 scenarios of Case 1.

Scenarios 1, 3, 4, 7 and 12 as in Table B.1 The paths that meet at edge e are $(P_{s_2,d_1}$ and $P_{s_2,d_2})$, $(P_{s_2,d_1}$ and $Q_{s_2,d_2})$, $(P_{s_2,d_1}$, P_{s_2,d_2} , and $Q_{s_2,d_2})$, $(P_{s_2,d_2}$ and $Q_{s_2,d_2})$, or $(Q_{s_1,d_2}$, P_{s_2,d_2} , and $Q_{s_2,d_2})$. The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be Y according to the hypothesis, which is the intersection of the last two cases of (B.7). The symbol Y will be forwarded and the invariant holds for the target edge e . If the symbols that enter edge v , the tail of e are different, then node v can compute the symbol Y according to Case 1.2 in Section 3.6 and send Y along e . The hypothesis holds in these scenarios.

Scenarios 2, 5 and 13 as in Table B.1 The paths that meet at edge e are $(P_{s_2,d_1}$ and $P_{s_1,d_1})$, $(P_{s_2,d_1}$, P_{s_1,d_1} , and $Q_{s_2,d_2})$, or $(P_{s_1,d_1}$ and $Q_{s_2,d_2})$. The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be X according to the hypothesis, which is the intersection of the first and third cases of (B.7). The symbol X will be forwarded and the invariant holds for the target edge e . If the symbols that enter edge v , the tail of e are different, then node v can compute X according to Case 1.3 in Section 3.6 and send X along e . The hypothesis holds in these scenarios.

Scenarios 6 and 11 as in Table B.1 The paths that meet at edge e are $(P_{s_1,d_1}$ and $P_{s_2,d_2})$ or $(Q_{s_1,d_2}$, P_{s_1,d_1} , and $P_{s_2,d_2})$. The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be $X + Y$ according to the hypothesis, which is the intersection of the first two cases of (B.7). The coded symbol $X + Y$ will be forwarded and the invariant holds for the target edge e . If the symbols that enter edge v , the tail of e are different, then node v can decode both X and Y and compute the coded symbol $X + Y$ according to Case 1.4 in Section 3.6 and send $X + Y$ along e . The hypothesis holds in these scenarios.

Scenarios 9 and 10 as in Table B.1 The paths that meet at edge e are $(Q_{s_1,d_2}$ and $P_{s_2,d_2})$, or $(Q_{s_1,d_2}$ and $Q_{s_2,d_2})$. The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be either Y or $X + Y$ according to the hypothesis. The symbol Y or $X + Y$ will be forwarded and the invariant holds for the target edge e . If the symbols that enter edge v , the tail of e are different, then node v can compute the symbol Y according to Case 1.2 in Section 3.6 and send Y along e . The hypothesis holds in these scenarios.

Scenario 8 as in Table B.1 The paths that meet at edge e are $(Q_{s_1,d_2}$ and $P_{s_1,d_1})$. The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be either X or $X + Y$ according to the hypothesis. The symbol X or $X + Y$ will be forwarded and the invariant holds for the target edge e . If the symbols that enter edge v , the tail of e are different, then node v can compute X according to Case 1.3 in Section 3.6 and send X along e . The hypothesis holds in this scenario. ■

VITA

VITA

Abdallah Khreishah received the B.S. degree from Jordan University of Science and Technology (JUST), Irbid, Jordan in 2004, and the M.S. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, in 2006. Since then he has been working toward the Ph.D. degree at Purdue University. His research interests include network coding, congestion control, opportunistic routing, and cross layer design in wireless networks.