Spring 2015

Local network coding on packet erasure channels -- From Shannon capacity to stability region

Wei-Chung Wang
Purdue University

Graduate School Form 30
(Updated 11/20/2014)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Wei-Cheng Kuo

Entitled
Local Network Coding on Packet Erasure Channels - From Shannon Capacity to Stability Region

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

CHIH-CHUN WANG

JAMES S. LEHNERT

JAMES V. KROGMEIER

MARK R. BELL

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification/Disclaimer (Graduate School Form 32), 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): CHIH-CHUN WANG

Approved by: Michael R. Melloch 03/04/2015

Head of the Department Graduate Program Date

LOCAL NETWORK CODING ON PACKET ERASURE CHANNELS

– FROM SHANNON CAPACITY TO STABILITY REGION

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Wei-Cheng Kuo

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2015

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

I always believe that there exists some meanings, which could not be understood at that moment, behind every choice in my life. Before joining Purdue, I never thought this choice, the PhD study at Purdue with Prof Chih-Chun Wang, could possibly bring me such a huge impact, from the problem solving techniques to the values of life. Six years has passed, but I could still remember the first time I saw Purdue when I got off the shuttle from the airport, the first meeting with Prof Wang, the first mid-term of ECE544, ... etc. Those things are so clear in my memory as if they just happened several days ago.

Some might say six years is pretty long. But the longer the stay, the more I cherish the moments here. Prof Chih-Chun Wang is not only my academic advisor, but also my mentor, who has wisely guided me through all those difficulties in these six years and helped me to build a strong mind. It is my best luck to work with him and to get inspired by his intelligence and his wisdom. This work would not happen and complete without his advises.

I would like to thank Prof James Lehnert, Prof Mark Bell, Prof Xiaojun Lin, and Prof James Krogmeier for serving in my candidacy and my committee and for their advices on my research. Thanks Jaemin Han, Imad Ahmad, Po-Kai Huang, Xiaohang Li, Can Zhao, and Shizen Zhao for their inspiring discussions and helps in the lab. Thanks Wei-Chang Yang, Stephanie Gong, Han-Ting Chiang, Sherwood Lee, and Prof Wang's wife for supporting my life at Purdue in these years. I would like to especially thank Mu-Sheng Lin, my best senior friend, for sharing his wisdom with me and for his help in my most difficult time.

This PhD dissertation is dedicated to my family. I could never accomplish these achievements without your endless support and love.

TABLE OF CONTENTS

Page

LIST OF TABLES . . . vii

LIST OF FIGURES . . . viii

ABSTRACT . . . x

1 INTRODUCTION . . . 1

1.1 Network Coding On Local Networks . . . 2

1.1.1 Network Coding On The Broadcast Channel . . . 2

1.1.2 Network Coding On The Butterfly Wireless Network . . . 3

1.1.3 Network Coding On The Opportunistic Routing . . . 4

1.1.4 A Critical Question . . . 4

1.2 From Shannon Capacity To Stability Region . . . 5

1.3 Our Contributions . . . 6

1.4 Thesis Outline . . . 8

2 MODEL FORMULATION . . . 9

2.1 The 1-to-M Broadcast Packet Erasure Channel . . . 9

2.2 The COPE Principle 2-Flow Wireless Butterfly Network With Opportunistic Routing and Broadcast Packet Erasure Channels . . . 10

2.2.1 A Useful Notation . . . 14

2.3 Chapter Summary . . . 15

3 SPACE BASED LINEAR NETWORK CODING . . . 16

3.1 Definitions . . . 16

3.2 An Instance of SBLNC Policies . . . 20

3.3 The Design Motivations of SBLNC Policies . . . 24

3.4 Chapter Summary . . . 27

4 THE SHANNON CAPACITY OF WIRELESS BUTTERFLY NETWORK . . . 28

Page

4.1 Related Works 28
4.2 Main Results 32
4.2.1 COPE Principle Relay Network Capacity 32
4.2.2 Capacity Outer Bound for COPE plus OpR 33
4.2.3 Capacity Inner Bound for COPE plus OpR 34
4.3 Capacity Approaching Coding Scheme 36
4.3.1 Achieving The Inner Bound of Proposition 4.2.3 36
4.3.2 Capacity of COPE Principle 2-Flow Wireless Butterfly Network Without Opportunistic Routing 42
4.4 Numerical Results 44
4.5 Chapter Summary 49
5 LINEAR NETWORK CODING SCHEDULING FOR 2-FLOW DOWNLINK TIME-VARYING BROADCAST PEC 50
5.1 The Problem of Inter-Session NC Scheduling 50
5.1.1 An Illustration of The Issues And Challenges 52
5.1.2 Channel Quality Varies Over Time In Wireless Scenarios 54
5.1.3 Contribution Overview 54
5.1.4 Related Works 55
5.2 Problem Formulation 56
5.2.1 The Broadcast Erasure Channel With Dynamic Packet Arrivals . 56
5.3 Existing Results And The Issues 57
5.3.1 Existing Results on Block INC Design 58
5.3.2 Stochastic Processing Networks (SPNs) 60
5.4 Chapter Summary 65
6 ROBUST AND OPTIMAL OPPORTUNISTIC SCHEDULING FOR DOWNLINK 2-FLOW NETWORK CODING WITH VARYING CHANNEL QUALITY AND RATE ADAPTION 66
6.1 The Proposed New INC Solution 66
6.1.1 Encoding 68

Page

6.1.2 Decoding and Buffer Management at Receivers 71
6.2 The Proposed Scheduling Solution 76
6.2.1 A Simple SPN model with Random Departure 76
6.2.2 The Proposed Scheduler For (0,1) Random SPNs 78
6.2.3 Performance Analysis 79
6.3 The Combined Solution 84
6.3.1 Extensions For Rate Adaption 87
6.4 Simulation Results 88
6.5 Chapter Summary 95
7 802.11-BASED INTER-SESSION NETWORK CODING MAC PROTOCOL WITH RATE ADAPTION 96
7.1 A Brief On 802.11 MAC 96
7.1.1 Network Topology 96
7.1.2 The Basic Mechanisms 97
7.1.3 Frame Structure 99
7.2 Practical MAC Protocol Based On 802.11n CSMA-CA 99
7.2.1 Additional Fields in 802.11 MAC Headers 100
7.2.2 When AP Wins The Medium Access Control 101
7.2.3 When STA Receives Packets 104
7.2.4 The Construction of The Event Probabilities Under Each Coding And Modulation Combination At AP 104
7.3 Chapter Summary 105
8 CONCLUSION AND FUTURE WORK 106
LIST OF REFERENCES 108
A The Converse Of The Shannon Capacity 112
B Detailed Achievability Analysis of SBLNC 125
C Bound-Matching of Wireless Butterfly Networks 133
D The Upper Bound of The Difference Between The Actual Queue And Intermediate Actual Queue 138

Page

E Proofs of Four Lemmas 142

F Proofs of Four Claims 150

G Sublinearly Growth of The Intermediate Actual Queue And The Aggregated Null Activities 156

H The Match of The Shannon Capacity And The Stability Region For 2-Flow Downlink Broadcast PEC 161

I The Lower Bound of The Summation of Random Variables 166

J The Non-Decreasing Property And The Sublinear Growth of The Deficits . . . 168

VITA 176

LIST OF TABLES

Table — Page

3.1 The resulting knowledge spaces at the end of Example 3.1.1. 19

4.1 The feature comparison between this thesis and [18] 29

4.2 Average sum-rates over 10000 random node replacements. 47

6.1 A summary of the REACTIVE-CODING operation 69

6.2 A summary of the transition probability of the virtual network in Fig. 6.1, where $p_{d_1 \vee d_2} \triangleq p_{d_1\overline{d_2}} + p_{\overline{d_1}d_2} + p_{d_1 d_2}$; $p_{d_1} \triangleq p_{d_1\overline{d_2}} + p_{d_1 d_2}$; NC1 stands for NON-CODING-1; CX stands for CLASSIC-XOR; DX1 stands for DEGENERATE-XOR-1; PM stands for PREMIXING; RC stands for REACTIVE-CODING. . 71

7.1 The address field in 802.11 MAC header. 98

LIST OF FIGURES

Figure Page

1.1 The illustration of local network coding gain on (a) the broadcast channel; (b) the COPE principle butterfly wireless network; and (c) the opportunistic routing, where the dashed arcs represent the broadcasting nature and the rectangle represents a packet. 2

2.1 (a) The 1-to-2 broadcast packet erasure channel; and (b) the 2-flow wireless butterfly network with opportunistic routing and packet erasure channels. . . 9

3.1 The illustration of the coding procedure in Example 3.1.1. We use a solid line to represent that the corresponding receiver has successfully received the packet and use a dot line to represent the case of erasure. 18

4.1 The illustration of the two–way relay channel for which s_1 sends X to s_2 and and s_2 sends Y to s_1. In (b), the common relay can send a linear combination $[X+Y]$ that benefits both destinations simultaneously. 31

4.2 An instance of the 2-flow wireless butterfly network with the success probabilities being indicated next to the corresponding arrows. We also assume that the success events between different node pairs are independent. 44

4.3 The capacity regions of the scenario in Fig. 4.2. The solid line indicates the throughput region of SBLNC. The dash line indicates the throughput region in [4]. The dot line indicates the throughput region of intra-session network coding (or random linear network coding). 44

4.4 (a) The relative location of (s_i, d_i). (b) Topology of two (s_i, d_i) pairs. 46

4.5 The cumulative distribution of the relative gap between the outer and the inner bounds when there is no fairness constraint. The outer and the inner bounds are described in Propositions 4.2.2 and 4.2.3, respectively. 46

5.1 The virtual networks of two INC schemes. 51

5.2 The two components of optimal dynamic INC design. 53

5.3 The time-varying broadcast packet erasure channel. 56

5.4 An SPN with random departure. 64

6.1 The virtual network of the proposed new INC solution. 67

Figure Page

6.2 The backlog of four different schemes for a time-varying channel with $\mathsf{cq}(t)$ uniformly distributed on $\{1,2\}$, and the packet delivery probability being $\vec{p} = (0, 0.5, 0.5, 0)$ if $\mathsf{cq}(t) = 1$ and $\vec{p} = (0, 0, 0, 1)$ if $\mathsf{cq}(t) = 2$. 89

6.3 The backlog comparison with $\mathsf{cq}(t)$ chosen from $\{1,2,3,4\}$ and the probability vectors are $\vec{p}^{(1)} = (0.14, 0.06, 0.56, 0.24)$, $\vec{p}^{(2)} = (0.14, 0.56, 0.06, 0.24)$, $\vec{p}^{(3)} = (0.04, 0.16, 0.16, 0.64)$, and $\vec{p}^{(4)} = (0.49, 0.21, 0.21, 0.09)$. 90

6.4 The backlog of four different schemes for rate adaptation with two possible (error-correcting-code rate,modulation) combinations. The back-pressure-based INC scheme in [39] is used in both aggressive and conservative 5-OP INC, where the former always chooses the high-throughput (rate,modulation) combination while the latter always chooses the low-throughput (rate,modulation) combination. 92

7.1 The MAC header in 802.11 [47]. 98

ABSTRACT

Kuo, Wei-Cheng Ph.D., Purdue University, May 2015. Local Network Coding on Packet Erasure Channels – From Shannon Capacity to Stability Region. Major Professor: Chih-Chun Wang.

Network Coding (NC) has emerged as a ubiquitous technique of communication networks and has extensive applications in both practical implementations and theoretical developments. While the Avalanche P2P file system from Microsoft, the MORE routing protocol, and the COPE coding architecture from MIT have implemented the idea of NC and exhibited promising performance improvements, a significant part of the success of NC stems from the continuing theoretic development of NC capacity, e.g., the Shannon capacity results for the single-flow multi-cast network and the packet erasure broadcast channel with feedback. However, characterizing the capacity for the practical wireless multi-flow network setting remains a challenging topic in NC. For example, the difficulties of finding the optimal NC strategy over multiple flows under varying-channel qualities and the rate adaption scenarios hinder any further advancement in this area. Despite the difficulty of characterizing the full capacity for large networks, there are evidences showing that even when using only local operations, NC can still recover substantial NC gain. We believe that a deeper understanding of multi-flow local network coding will play a key role in designing the next-generation high-throughput coding-based wireless network architecture.

This thesis consists of three parts. In the first part, we characterize the full Shannon capacity region of the "*COPE*" principle when applied to a 2-flow wireless butterfly network with broadcast packet erasure channels. The capacity results allow for random overhearing probabilities, arbitrary scheduling policies, network-wide channel state information (CSI) feedback after each transmission, and potential use of non-linear network codes. We propose a theoretical outer bound and a new class of linear network codes, named the

Space-Based Linear Network Coding (SBLNC), that achieves the capacity outer bound. Numerical experiments show that SBLNC provides close-to-optimal throughput even in the scenario with opportunistic routing.

In the second part, we further consider the complete network dynamics of stochastic arrivals and queueing and study the corresponding stability region. Based on dynamic packet arrivals, the resulting solution would be one step closer to practical implementation, when compared to the previous block-code-based capacity study. For the 2-flow downlink scenario, we propose the first opportunistic INC + scheduling solution that is provably optimal for time-varying channels, i.e., the corresponding stability region matches the optimal Shannon capacity. Specifically, we first introduce a new *binary INC* operation, which is distinctly different from the traditional wisdom of XORing two overheard packets. We then develop a *queue-length-based* scheduling scheme, which, with the help of the new INC operation, can robustly and optimally adapt to time-varying channel quality. We then show that the proposed algorithm can be easily extended for rate adaptation and it again robustly achieves the optimal throughput.

In the third part, we propose an 802.11-based MAC layer protocol which incorporates the rate adaption solution developed in the second part. The new MAC protocol realizes the promised intersession network coding gain for two-flow downlink traffic with short decoding delay. Furthermore, we delicately retain the CSMA-CA distributed contention mechanism with only 17 bits new header field changes, and carefully ensure the backward compatibility. In summary, the new solution demonstrates concrete throughput improvement without alternating the too much packet-by-packet traffic behavior. Such a feature is critical in practical implementation since it allows the network coding solution to be transparent to any arbitrary upper layer applications.

1. INTRODUCTION

As the number of smartphone users growing to the majority of wireless carrier customers, the demand of wireless data rate has increased rapidly and expanded beyond the the traditional wireline service requirements. How to increase the wireless data rate supporting multiple users simultaneously with certain scarce resources of the communication network thus eventually becomes a critical and urgent topic. There are many possible solutions, e.g., ultra wide band communication and the multiple-input multiple-out antenna. Nonetheless, network coding is one of the most promising directions which could potentially provide considerable end-to-end throughput improvement and protect the data privacy of individual users.

Inspired by the butterfly network, Ahlswede *et al.* proposed the concept of network coding in 2000 [1]. Since then, network coding has emerged as a ubiquitous technique of modern data communication networks. The extensive applications of network coding spread from practical implementations to theoretical results. The Avalanche P2P file system from Microsoft removes the need of receiving all individual pieces of the original file as in the BitTorrent system. The MORE protocol from MIT alleviates the use of a scheduler to coordinate the transmission as in previous opportunistic routing protocols. The COPE architecture from MIT incorporate network coding across multiple sessions and demonstrates that the existing TCP/IP network layer transmission still has great potential to further increase the overall throughput by 40% to 200%. All the above implementations are based on the concept of network coding. Furthermore, in the areas of the network security, the data center, and the analog signal processing, network coding has exhibited great potential on augmenting their current performance. Meanwhile, a significant part of the success of network coding stems from the continuing theoretic development of network coding capacity. The seminal work [2] in 2003 utilized network coding as the backbone and prosed “random linear network coding” to achieve the Shannon capacity of single-flow (or single-session)

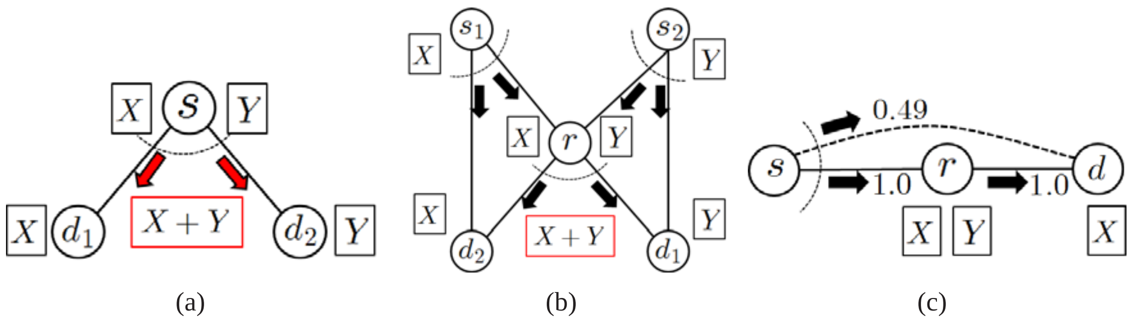


Fig. 1.1.: The illustration of local network coding gain on (a) the broadcast channel; (b) the COPE principle butterfly wireless network; and (c) the opportunistic routing, where the dashed arcs represent the broadcasting nature and the rectangle represents a packet.

multi-cast networks. And the feedback capacity of the broadcast channel, the long-term open question in the area of Shannon capacity, has also been resolved by network coding for the packet erasure channel case [3, 4].

1.1 Network Coding On Local Networks

However, even though COPE [5] has exhibited the great potential for network coding being applied to multi-user (or multi-flow) wireless data networks, its corresponding theoretical capacity remains largely unsolved. The difficulties of finding the optimal network coding strategy over multiple flows hinder the further advancement in this area. Despite the difficulty of characterizing the full capacity for large networks, there are evidences showing that even when using only local operations, network coding can still recover substantial network coding gain. In the following, we are going to present three examples that can demonstrate substantial network coding gain even on the local operations.

1.1.1 Network Coding On The Broadcast Channel

The first example is the network coding gain on the broadcast packet erasure channel. Figure 1.1(a) illustrates the scenario where the network coding can benefit the throughput. As shown in Figure 1.1(a), the dashed arc represents the node s can broadcast packets to d_1

and d_2 simultaneously with certain probabilities. Assume d_1 would like to convey packet X and d_2 would like to convey packet Y. However, in the first two transmissions by r, unfortunately, d_1 receives Y and d_2 receives X. But then in the next time slot, the node s can transmit the combined packet $X + Y$ and both the destinations can recover the desired packets if it receives the packet $X + Y$. Without network coding, the node s needs to keep transmit X and Y separately until d_1 and d_2 receive the desired packet. Recently, [3] and [4] successfully characterized the full capacity region of the 1-hop broadcast packet erasure channel with ≤ 3 coexisting flows.

1.1.2 Network Coding On The Butterfly Wireless Network

The second example is the network coding gain on the COPE principle butterfly wireless network. Figure 1.1(b) illustrates its scenario and the dashed arc represent the corresponding source node can broadcast packets to the connected end nodes. Suppose source s_1 would like to send a packet X to destination d_1; source s_2 would like to send a packet Y to d_2; and they are allowed to share a common relay r. Also suppose that when s_1 (resp. s_2) sends X (resp. Y) to r, destination d_2 (resp. d_1) can overhear packet X (resp. Y). We further assume that after the first two transmissions, both d_1 and d_2 can use feedback to inform r the overhearing status at d_1 and d_2, respectively. Then instead of transmitting two packets X and Y separately, the relay node r can send the linear combination $[X+Y]$. Each destination d_i can then decode its desired packet by subtracting the overheard packet from the linear combination $[X + Y]$. In the above simple example, the traditional store-and-forward transmission scheme requires at least 4 transmissions (s_1 to r, s_2 to r, r to d_1, and r to d_2). But with the network coding in COPE scheme, it only requires 3 transmissions.

Despite its simple nature, the exact capacity region of the COPE principle remains an open problem even for the simplest case of two coexisting flows. Several attempts have since been made to quantify some suboptimal achievable rate regions of the COPE principle [6–15]. One difficulty of deriving the capacity region is due to the use of feedback in the COPE principle. It is shown in [16] that although feedback could strictly enhance the capacity in a multi-unicast environment, the exact amount of throughput improvement is hard to quantify. [17] proposes one queue-based approach for the general wireline and wireless networks considering both inter-session and intra-session network coding. However, the results in [17] mainly focus on the benefits from the side information to decide either inter-session or intra-session network coding should be applied, which is more related to [18]. [18] circumvents the difficulty of feedback-based analysis by considering a special class of 2-staged coding schemes. Although the results in [18] fully capture the benefits of message side information [16, 17, 19–23] , they capture only partially the feedback benefits, which leads again to a strictly suboptimal achievable rate region.

1.1.3 Network Coding On The Opportunistic Routing

The third example is the network coding gain on the opportunistic routing. Figure 1.1(c) illustrates its scenario and the dashed arc represent the node s can broadcast packets to both r and d with certain probabilities indicated. Here we only have one session from s to d and d would like to receive both packets X and Y. After two transmissions from s, the relay r receives both X and Y while d only receives X. Then the relay r can directly transmit the combined packet $X + Y$ and d can recover Y linear operations. Without network coding, then the opportunistic routing scheme requires a scheduler to inform r what has been received by d. And then r can transmit Y. Keeping track of which packets have been heard or not is a daunting task and network coding drastically simplify it. Recent works [24–26] take the advantage illustrated in this example to remove the need of a central scheduler and experimentally show that with network coding in a 20-node wireless testbed, the unicast throughput can be 22% higher than the existing opportunistic routing protocols and 95% higher than the current state-of-art best routing protocol for wireless mesh networks.

1.1.4 A Critical Question

All the above schemes can augment the end-to-end throughput in the multi-flow wireless network. An interesting question thus rises: can we combine all of them together?

Or can we optimize all of them simultaneously? Furthermore, can we do its analytically Shannon capacity? To answer these questions, we believe that a deeper understanding of multi-flow local network coding will play a key role, which will also benefit designing the next-generation high-throughput coding-based wireless network architecture. The analysis in this thesis is on the packet level as the COPE operating on the current TCP/IP network layer. With the ARQ mechanisms in the data link layer, the packet erasure channel setting thus is a natural choice. Hence the anlysis of Shannon capacity of 2-flow wireless butterfly network with broadcast packet erasure channel turns to our primary objective.

1.2 From Shannon Capacity To Stability Region

The analysis of Shannon capacity is an essential part to establish the possible solutions for the communication networks of interest. However, Shannon capacity is still quite far away from practical implementations. In the analysis of Shannon capacity, the results are derived based on the assumption that the input block code is fixed and the block code length can be infinitely long. This assumption is apparently impractical because of the memory buffer limit in real systems. Furthermore, this assumption would also induce the extremely large decoding delay and control overhead. These flaws deviate the analysis of Shannon capacity from the practical implementations.

Hence we further broaden our attention to the stability region of network coding schemes. The stability region of the network is defined as the set of all end-to-end traffic load that can be supported under the appropriate selection of the network control policy [27]. The analysis of stability region considers the complete dynamics of stochastic arrivals and queueing. The assumption of dynamic arrivals greatly alleviates the flaws in the block-based Shannon capacity analysis, including the problems of memory buffer limit, decoding delay, and the large control overhead, and promotes the entire analysis one step closer to the practical implementations.

However, there are several difficulties which block the extension from Shannon capacity to the stability region. With fixed block codes as the input in the Shannon capacity, the overall throughput in the end of the transmissions can be analyzed by the law of large number. This does not hold for the case of stochastic arrivals (which means the packets are endlessly injected into the network) and other tools are required to analyze the queue dynamics at each time instance. Tassiulas *et al.* [28] introduced Lyaponov drift to resolve this problem and leaded to the establishment of the network stability analysis research.

Other than the difference between the analysis tools, however, in the existing store-and-forward stability analysis, the stability region is defined on considering all possible scheduling, routing, and resource allocation, but no coding allowed inside the network [27]. The nature of combining packets inside the network provides further challenges for online coding and scheduling implementation. Several attempts have been made to resolve this problem [17, 29]. However, the existing proposed solutions all tends to circumvent the problem of combining packets by converting the network coding scheduling problem back to the existing store-and-forward scheduling problem. This kind of conversions highly relies on case-by-case discussion and lack of the generality to be systematically applied to other network topologies. A general network control algorithm which can incorporate the nature of combining packets inside the network thus is an important subject for the network coding stability analysis.

1.3 Our Contributions

Our contributions consists of two parts. In the first part, we characterize the full Shannon capacity of the COPE principle when applied to a 2-flow wireless butterfly network with broadcast packet erasure channels. The capacity results allow for random overhearing probabilities, arbitrary scheduling policies, network-wide channel state information (CSI) feedback after each transmission, and potential use of non-linear network codes. An information-theoretic outer bound is derived that takes into account the delayed CSI feedback of the underlying broadcast packet erasure channels. We then propose a new class of linear network codes, named the Space-Based Linear Network Coding (SBLNC). SBLNC provides a systematic approach to keep tracking the evolution of knowledge space at each node. We prove that SBLNC can achieve the capacity region of the 2-flow wireless butterfly network without considering the opportunistic routing. Furthermore, numerical experiments show that SBLNC provides close-to-optimal throughput even in the scenario with opportunistic routing.

In the second part of the contributions, we propose a new optimal dynamic INC design for 2-flow downlink traffic with time-varying packet erasure channels. Our detailed contributions are summarized as follows.

Contribution 2.1: We introduce a new pair of INC operations such that (i) The underlying concept is distinctly different from the traditional wisdom of XORing two overheard packets; (ii) The overall scheme uses only the ultra-low-complexity binary XOR operation; and (iii) The new set of INC operations is guaranteed to achieve the block-code-based Shannon capacity.

Contribution 2.2: The introduction of new INC operations leads to a new vr-network for which the existing "*vr-network decoupling* + *BP*" approach in [30] no longer holds. We generalize the results of Stochastic Processing Networks (SPNs) [31, 32] and successfully apply it to the new vr-network. The end result is an opportunistic, dynamic INC solution that is completely *queue-length-based* and can robustly adapt to time-varying channels while achieving the largest possible stability region.

Contribution 2.3: The proposed solution can also be readily generalized for rate-adaptation. Through numerical experiments, we have shown that a simple extension of the proposed scheme can opportunistically and optimally choose the order of modulation and the rate of the error correcting codes used for each packet transmission while achieving the optimal stability region, i.e., equal to the Shannon capacity.

Contribution 2.4: A byproduct of our results is a scheduling scheme for SPNs with *random departure*. The new results relax the previous assumption of *deterministic departure*, a major limitation of the existing SPN model, by considering stochastic packet departure behavior. The new scheduling solution could thus further broaden the applications of SPN scheduling to other real-world scenarios.

In the third part of contributions, we propose an 802.11-based MAC layer protocol which incorporates the rate adaption solution developed in the second part. The new MAC protocol realizes the promised intersession network coding gain for two-flow downlink traffic with short decoding delay. Furthermore, we delicately retain the CSMA-CA distributed contention mechanism with only 17 bits new header field changes, and carefully ensure the backward compatibility. In summary, the new solution demonstrates concrete throughput improvement without alternating the too much packet-by-packet traffic behavior. Such a feature is critical in practical implementation since it allows the network coding solution to be transparent to any arbitrary upper layer applications.

1.4 Thesis Outline

In the next chapter, we formulate the local network model which incorporate the broadcast PEC with feedback, the COPE principle, and the opportunistic routing all together. The stability region problem is also formulated. In Chapter 3, we describe the central idea of this thesis – Spaced-Based Linear Network Coding. In Chapter 4, we characterize the full Shannon capacity of 2-flow wireless butterfly network with broadcast packet erasure channels. In Chapter 5, we start to discuss the linear network coding stability region and introduce its analogy the stochastic processing network. In Chapter 6, we propose the modified Deficit Maximum Weight algorithm and fully characterize the stability region of the 2-receiver multi-input broadcast packet erasure channel. In Chapter 7, we propose a 802.11-based MAC protocol which incorporates the rate adaption solution developed in Chapter 6. In Chapter 8, we conclude this thesis and discuss the possible extensions and applications.

2. MODEL FORMULATION

In this chapter, we will first formulate the 1-to-M broadcast packet erasure channel as a mathematical model. We then propose a general wireless butterfly model which incorporates the broadcast packet erasure channels with feedback, the COPE principle, and the opportunistic routing all together. A useful probability function which can intuitively describe the probability of interest is defined. We finally discuss the dynamic network coding and scheduling decision in the 1-to-2 broadcast packet erasure channel. We first define a useful notation. For any positive integer M, define $[M] \triangleq \{1, \cdots, M\}$.

2.1 The 1-to-M Broadcast Packet Erasure Channel

Given a finite field $\mathsf{GF}(q)$. A 1-to-M broadcast packet erasure channel (PEC) takes an input packet $X_s \in \mathsf{GF}(q)$ from the source s and outputs an M-dimension vector $\mathbf{Y} = (Y_{s\to d_1}, Y_{s\to d_2}, ..., Y_{s\to M})$, where $Y_{s\to d_i} \in \{X_s, *\}$ for all $i \in [M]$. Figure 2.1(a) illustrates a 1-to-2 broadcast packet erasure channel. Here $*$ denotes the erasure symbol. $Y_{s\to d_i} = *$

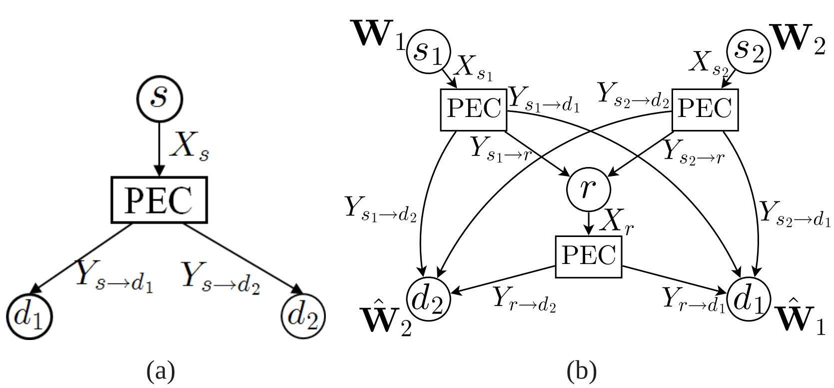


Fig. 2.1.: (a) The 1-to-2 broadcast packet erasure channel; and (b) the 2-flow wireless butterfly network with opportunistic routing and packet erasure channels.

means that the i-th receiver does not receive the input X_s. We also assume that there is no other type of noise, i.e., the received $Y_{s\to d_i}$ is either X_s or $*$.

We consider only *stationary and memoryless PECs*, i.e., the erasure pattern is independently and identically distributed (i.i.d.) for each channel usage. The characteristics of a memoryless 1-to-M PEC can be fully described by 2^M *successful reception probabilities* $p_{s\to T\overline{[M]\backslash T}}$ indexed by any subset $T \subset [M]$. That is, $p_{s\to T\overline{[M]\backslash T}}$ denotes the probability that a packet X_s sent from source s is heard by and only by the i-th destination for all $i \in T$. For example, suppose $M = 3$, then $p_{s\to\{1,3\}\overline{\{2\}}}$ denotes the probability that a packet X_s is heard by d_1 and d_3 but not by d_2.

For any time t, we use an M-dimensional *channel status vector* $\mathbf{Z}_s(t)$ to represent the channel reception status of the 1-to-M broadcast packet erasure channel:

$$\mathbf{Z}_s(t) = (Z_{s\to d_1}(t), Z_{s\to d_2}(t), ..., Z_{s\to d_m}(t)) \in \{*, 1\}^M$$

where "$*$" and "1" represent erasure and successful reception, respectively. That is, when s transmits a packet $X_s(t) \in \mathsf{GF}(q)$ in time t, the destination d_m receives $Y_{s\to d_m}(t) = X_s(t)$ if $Z_{s\to d_m}(t) = 1$ and receives $Y_{s\to d_m}(t) = *$ if $Z_{s\to d_m}(t) = *$. For simplicity, we use $Y_{s\to d_m}(t) = X_s(t) \circ Z_{s\to d_m}(t)$ as shorthand. Since here we only consider stationary and memoryless PECs, $\mathbf{Z}_s(t)$ is i.i.d. over time.

2.2 The COPE Principle 2-Flow Wireless Butterfly Network With Opportunistic Routing and Broadcast Packet Erasure Channels

Here we are going to construct a local network model which incorporates the network coding gain on (1) the COPE principle, (2) the opportunistic routing, and (3) the broadcast packet erasure channel with feedback. The COPE principle 2-flow wireless butterfly network with opportunistic routing and broadcast packet erasure channels is modeled as follows. We consider a 5-node 2-hop relay network with two source-destination pairs (s_1, d_1) and (s_2, d_2) and a common relay r interconnected by three broadcast PECs. See Fig. 2.1(b) for the illustration. Specifically, source s_i can use a 1-to-3 broadcast PEC to communicate with $\{d_1, d_2, r\}$ for $i = 1, 2$, and relay r can use a 1-to-2 broadcast PEC to communicate with $\{d_1, d_2\}$. To accommodate the discussion of opportunistic routing, we allow s_i to directly communicate with d_i, see Fig. 2.1(b). When opportunistic routing is not permitted (as in the case when focusing exclusively on the COPE principle and the broadcast packet erasure channels), we simply choose the PEC channel success probabilities $p_{s_i \to \cdot}$ such that the probability that d_i can hear the transmission from s_i is zero.

We assume slotted transmission. Within an overall time budget of n time slots, source s_i would like to convey nR_i packets $\mathbf{W}_i \triangleq (W_{i,1}, \cdots, W_{i,nR_i})$ to destination d_i for all $i \in \{1, 2\}$ where R_i is the rate for flow i. For each $i \in \{1, 2\}$, $j \in [nR_i]$, the information packet $W_{i,j}$ is assumed to be independently and uniformly randomly distributed over $\mathsf{GF}(q)$.

For any time t, we use an 8-dimensional *channel status vector* $\mathbf{Z}(t)$ to represent the channel reception status of the entire network:

$$\begin{aligned}\mathbf{Z}(t) = (&Z_{s_1 \to d_1}(t), Z_{s_1 \to d_2}(t), Z_{s_1 \to r}(t), Z_{s_2 \to d_1}(t),\\ &Z_{s_2 \to d_2}(t), Z_{s_2 \to r}(t), Z_{r \to d_1}(t), Z_{r \to d_2}(t)) \in \{*, 1\}^8\end{aligned}$$

where "$*$" and "1" represent erasure and successful reception, respectively. That is, when s_1 transmits a packet $X_{s_1}(t) \in \mathsf{GF}(q)$ in time t, relay r receives $Y_{s_1 \to r}(t) = X_{s_1}(t)$ if $Z_{s_1 \to r}(t) = 1$ and receives $Y_{s_1 \to r}(t) = *$ if $Z_{s_1 \to r}(t) = *$. For simplicity, we use $Y_{s_1 \to r}(t) = X_{s_1}(t) \circ Z_{s_1 \to r}(t)$ as shorthand.

In this thesis, we consider the node-exclusive interference model. That is, we allow only one node to be scheduled in each time slot. The scheduling decision at time t is denoted by $\sigma(t)$, which takes value in the set $\{s_1, s_2, r\}$. For example, $\sigma(t) = s_1$ means that node s_1 is scheduled for time slot t. For convenience, when s_1 is not scheduled at time t, we simply set $Y_{s_1 \to r}(t) = *$. As a result, the scheduling decision can be incorporated into the following expression of $Y_{s_1 \to r}(t)$:

$$Y_{s_1 \to r}(t) = X_{s_1}(t) \circ Z_{s_1 \to r}(t) \circ 1_{\{\sigma(t) = s_1\}}.$$

Similar notation is used for all other received signals. For example, $Y_{r\to d_2}(t) = X_r(t) \circ Z_{r\to d_2}(t) \circ 1_{\{\sigma(t)=r\}}$ is what d_2 receives from r in time t, where $X_r(t)$ is the packet sent by r in time t.

We assume that the 3 PECs are memoryless and stationary. Namely, we allow arbitrary joint distribution for the 8 coordinates of $\mathbf{Z}(t)$ but assume that $\mathbf{Z}(t)$ is i.i.d. over the time axis t. We also assume $\mathbf{Z}(t)$ is independent of the information messages $\mathbf{W}_1$ and $\mathbf{W}_2$.

For simplicity, we use brackets $[\cdot]_1^t$ to denote the collection from time 1 to t. For example, $[\sigma, \mathbf{Z}, Y_{s_1\to d_2}]_1^t \triangleq \{\sigma(\tau), \mathbf{Z}(\tau), Y_{s_1\to d_2}(\tau) : \forall\tau \in [1,t]\}$. Also, for any $S \subseteq \{s_1, s_2, r\}$ and $T \subseteq \{r, d_1, d_2\}$, we define

$$\mathbf{Y}_{S\to T}(t) \triangleq \{Y_{s\to d}(t) : \forall s \in S, \forall d \in T\}.$$

For example, $\mathbf{Y}_{\{s_1,r\}\to\{d_1,d_2\}}(t)$ is the collection of $Y_{s_1\to d_1}(t)$, $Y_{s_1\to d_2}(t)$, $Y_{r\to d_1}(t)$, and $Y_{r\to d_2}(t)$.

Given the rate vector (R_1, R_2), a joint scheduling and network coding (NC) scheme is defined by n scheduling decision functions

$$\forall t \in [n],\ \sigma(t) = f_{\sigma,t}([\mathbf{Z}]_1^{t-1}), \tag{2.1}$$

$3n$ encoding functions at s_1, s_2, and r, respectively: For all $t \in [n]$

$$X_{s_i}(t) = f_{s_i,t}(\mathbf{W}_i, [\mathbf{Z}]_1^{t-1}), \quad \forall i \in \{1,2\}, \tag{2.2}$$

$$X_r(t) = f_{r,t}([\mathbf{Y}_{\{s_1,s_2\}\to r}, \mathbf{Z}]_1^{t-1}), \tag{2.3}$$

and 2 decoding functions at d_1 and d_2, respectively:

$$\hat{\mathbf{W}}_i = f_{d_i}([\mathbf{Y}_{\{s_1,s_2,r\}\to d_i}, \mathbf{Z}]_1^n), \quad \forall i \in \{1,2\}. \tag{2.4}$$

By (2.1), we allow $\sigma(t)$, the scheduling decision at time t, to be a function of the network-wide reception status vectors before time t. By (2.2), the encoding decision at s_i

is a function depending on the information messages and past channel status. Encoding at r depends on what r received in the past and the past channel status vector, see (2.2). In the end, d_i decodes $\mathbf{W}_i$ based on what d_i has received and the past channel status of the entire network.[1] We allow the encoding and decoding functions $f_{s_i,t}$, $f_{r,t}$, and f_{d_i} to be linear or nonlinear.

This setting models the scenario in which there is a dedicated, error-free, low-rate control channel that can broadcast the previous network channel status $\mathbf{Z}(t-1)$ causally to all network nodes. The total amount of control information is no larger than 8 bits per time slot, which is much smaller than the actual payload of each packet $\approx 10^4$ bits. As a result, the perfect feedback channel can be easily implemented by piggybacking[2] on the data packets. The scheduling decision $\sigma(t)$ can be computed centrally (by a central controller) or distributively by each individual node since we allow all nodes to have the knowledge of the reception status of the entire network.

Definition 2.2.1 *Fix the distribution of* $\mathbf{Z}(t)$*. A rate vector* (R_1, R_2) *is achievable if for any* $\epsilon > 0$*, there exists a joint scheduling and NC scheme with sufficiently large* n *and* $\mathsf{GF}(q)$ *such that*

$$\max_{\forall i \in \{1,2\}} \mathsf{Prob}(\mathbf{W}_i \neq \hat{\mathbf{W}}_i) < \epsilon.$$

The capacity region is defined as the closure of all achievable rate vectors (R_1, R_2)*.*

Remark: In (2.1), the scheduling decision $\sigma(t)$ does not depend on the information messages $\mathbf{W}_i$, which means that we prohibit the use of timing channels [33,34]. Even when we allow the usage of timing channels, we conjecture that the overall capacity improvement with the timing channel techniques is negligible. A heuristic argument is that each successful packet transmission gives $\log_2(q)$ bits of information while the timing information (to transmit or not) gives roughly 1 bit of information. When focusing on sufficiently large

[1]Since the scheduling decision $\sigma(t)$ is a function of $[\mathbf{Z}]_1^{t-1}$, all the encoding functions in (2.2) and (2.3), and the decoding functions in (2.4) also know implicitly the scheduling decision $\sigma(t)$.
[2]Some pipelining may be necessary to mitigate the propagation delay of the feedback control messages.

$\mathsf{GF}(q)$, additional gain of timing information is thus likely to be absorbed in our timing-information-free capacity characterization. In our setting, r is the only node that can mix packets from two different data flows. Further relaxation such that s_1 and s_2 can hear each other and perform coding accordingly is beyond the scope of this work.

2.2.1 A Useful Notation

In our network model, there are 3 broadcast PECs associated with s_1, s_2, and r, respectively. We sometimes term those PECs the s_i-PEC, $i = 1, 2$, and the r-PEC. Since only one node can be scheduled in each time slot, we can assume that the reception events of each PEC are independent from that of the other PECs. As a result, the distribution of the network-wide channel status vector $\mathbf{Z}(t)$ can be described by the probabilities $p_{s_i \to T\overline{\{r,d_1,d_2\}\backslash T}}$ for all $i \in \{1, 2\}$ and for all $T \subseteq \{r, d_1, d_2\}$, and $p_{r\to U\overline{\{d_1,d_2\}\backslash U}}$ for all $U \subseteq \{d_1, d_2\}$. Totally there are $8 + 8 + 4 = 20$ parameters. By allowing some of the coordinates of $\mathbf{Z}(t)$ to be correlated, our setting can also model the scenario in which destinations d_1 and d_2 are situated in the same physical node and thus have perfectly correlated channel success events.

For notational simplicity, we also define the following three *probability* functions $p_{s_i}(\cdot)$, $i = 1, 2$, and $p_r(\cdot)$, one for each of the PECs. The input argument of each function p_s (s being one of $\{s_1, s_2, r\}$) is a collection of the elements in $\{d_1, d_2, r, \overline{d_1}, \overline{d_2}, \overline{r}\}$. The function $p_s(\cdot)$ outputs the probability that the reception event is *compatible* to the specified collection of $\{d_1, d_2, r, \overline{d_1}, \overline{d_2}, \overline{r}\}$. For example,

$$p_{s_1}(d_2\overline{r}) = p_{s_1\to d_2\overline{d_1 r}} + p_{s_1\to d_1 d_2 \overline{r}} \tag{2.5}$$

is the probability that the input of the s_1-PEC is successfully received by d_2 but not by r. Herein, d_1 is a *don't-care* receiver and $p_{s_1}(d_2\overline{r})$ thus sums two joint probabilities together (d_1 receives it or not) as described in (2.5). Another example is $p_r(d_2) = p_{r\to d_1 d_2} + p_{r\to\overline{d_1}d_2}$, which is the probability that a packet sent by r is heard by d_2. To slightly abuse the notation, we further allow $p_s(\cdot)$ to take multiple input arguments separated by the comma sign ",".

With this new notation, $p_s(\cdot)$ then represents the probability that the reception event is compatible to *at least* one of the input arguments. For example,

$$p_{s_1}(d_1\overline{d_2}, r) = p_{s_1\to d_1\overline{d_2 r}} + p_{s_1\to d_1\overline{d_2}r} + p_{s_1\to d_1 d_2 r}$$
$$+ p_{s_1\to \overline{d_1}d_2 r} + p_{s_1\to \overline{d_1 d_2}r}.$$

That is, $p_{s_1}(d_1\overline{d_2}, r)$ represents the probability that $(Z_{s_1\to d_1}, Z_{s_1\to d_2}, Z_{s_1\to r})$ equals one of the following 5 vectors $(1,*,*)$, $(1,*,1)$, $(1,1,1)$, $(*,1,1)$, and $(*,*,1)$. Note that these 5 vectors are compatible to either $d_1\overline{d_2}$ or r or both. Another example of this $p_s(\cdot)$ notation is $p_{s_1}(d_1, d_2, r)$, which represents the probability that a packet sent by s_1 is received by *at least* one of the three nodes d_1, d_2, and r.

2.3 Chapter Summary

In this chapter, we formulate the model of the 1-to-M broadcast packet erasure channel in Section 2.1. In Section 2.2, we then construct a wireless butterfly network model including the COPE principle, the opportunistic routing, and the broadcast packet erasure channels with feedback. The corresponding Shannon capacity region is also defined in Section 2.2.

3. SPACE BASED LINEAR NETWORK CODING

In this chapter, we introduce a the central idea of this thesis. We will present a class of network coding scheme named the "Space-Based Linear Network Code (SBLNC)" scheme. An example of SBLNC scheme policies will be provided and will also be used to explain the design motivations of the SBLNC policies. The SBLNC schemes will later be used to achieve(1) the Shannon capacity of the 2-flow wireless butterfly network with packet erasure channels and (2) the stability region of the 2-flow 1-to-2 broadcast packet erasure channel with feedback.

3.1 Definitions

Here we use the 2-flow wireless butterfly network with packet erasure channels in Section 2.2 as an example and construct the corresponding SBLNC scheme. We first provide some basic definitions that will be used when describing an SBLNC scheme.

For $i = 1, 2$, a *flow-i coding vector* $\mathbf{v}^{(i)}$ is an nR_i-dimensional row vector with each coordinate being a scalar in $\mathsf{GF}(q)$. Any linear combination of the message symbols $W_{i,1}$ to W_{i,nR_i} can thus be represented by $\mathbf{v}^{(i)}\mathbf{W}_i^{\mathsf{T}}$ where $\mathbf{W}_i^{\mathsf{T}}$ is the transpose of $\mathbf{W}_i$. We use the superscript "(i)" to emphasize that we are focusing on a flow-i vector.

We define the *flow-i message space* by $\Omega_i \triangleq (\mathsf{GF}(q))^{nR_i}$, an nR_i-dimensional linear space. In the following, we define the following 6 *knowledge spaces* S_r, S_{d_1}, S_{d_2}, T_r, T_{d_1}, and T_{d_2} for the 5-node relay network in Fig. 2.1(b).

The knowledge spaces S_r, S_{d_2}, S_{d_1} are linear subspaces of Ω_1 and represent the knowledge about the flow-1 packets at nodes r, d_2, and d_1, respectively. Symmetrically, the knowledge spaces T_r, T_{d_1}, and T_{d_2} are linear subspaces of Ω_2 and represent the knowledge about the flow-2 packets at nodes r, d_1, and d_2, respectively. In the following, we discuss the detailed construction of S_r, S_{d_2}, and S_{d_1} and the construction of T_r to T_{d_2} follows symmetrically.[1]

- *In the end of any time t, $S_r(t) \subset \Omega_1$ is the* linear span *of a group of $\mathbf{v}^{(1)}$, denoted by $\mathbf{V}^{(1)}_{s_1\to r}$. The group $\mathbf{V}^{(1)}_{s_1\to r}$ contains the $\mathbf{v}^{(1)}$ vectors sent by s_1 during time 1 to t and have been received successfully by r. Throughout the paper, we use the convention that the linear span of an empty set is a set containing the zero vector, i.e.,* $\mathsf{span}\{\emptyset\} = \{0\}$. *For example, if r has not yet received any packet from s_1, then by convention $S_r(t) = \{0\}$.*

- *In the end of time t, $S_{d_2}(t) \subset \Omega_1$ is the linear span of two groups of $\mathbf{v}^{(1)}$, denoted by $\mathbf{V}^{(1)}_{s_1\to d_2}$ and $\mathbf{V}^{(1)}_{r,N\to d_2}$. The first group $\mathbf{V}^{(1)}_{s_1\to d_2}$ contains the $\mathbf{v}^{(1)}$ vectors corresponding to the packets sent by s_1 during time 1 to t and have been received successfully by d_2. The second group $\mathbf{V}^{(1)}_{r,N\to d_2}$ contains the $\mathbf{v}^{(1)}$ vectors corresponding to the packets sent by r during time 1 to t that* are not mixed with any other flow-2 packets. *The letter "N" in the subscript stands for Not-inter-flow-coded transmission.*

- *In the end of time t, $S_{d_1}(t) \subset \Omega_1$ is the linear span of three groups of $\mathbf{v}^{(1)}$, denoted by $\mathbf{V}^{(1)}_{s_1\to d_1}$, $\mathbf{V}^{(1)}_{r,N\to d_1}$, and $\mathbf{V}^{(1)}_{r,C\to d_1}$. The first group $\mathbf{V}^{(1)}_{s_1\to d_1}$ contains the $\mathbf{v}^{(1)}$ vectors corresponding to the packets sent by s_1 during time 1 to t and have been received successfully by d_1. The second group $\mathbf{V}^{(1)}_{r,N\to d_1}$ contains the $\mathbf{v}^{(1)}$ vectors corresponding to the packets sent by r during time 1 to t that* are not mixed with any other flow-2 packets. *The third group $\mathbf{V}^{(1)}_{r,C\to d_1}$ contains the $\mathbf{v}^{(1)}$ vectors* that can be decoded from the inter-flow coded packets[2] sent by r during time 1 to t. *The letter "C" in the subscript stands for inter-flow-Coded transmission.*

In sum, we use S and T to distinguish whether we are focusing on flow-1 or flow-2 packets, respectively, and we use the subscripts to describe the node of interest. One can easily see that these six knowledge spaces evolve over time since each node may receive

[1]The construction of T_{d_1}(resp. T_{d_2}) follows the construction of S_{d_2}(resp. S_{d_1}).

[2]When the relay r sends a linear combination of both flow-1 and -2 packets.

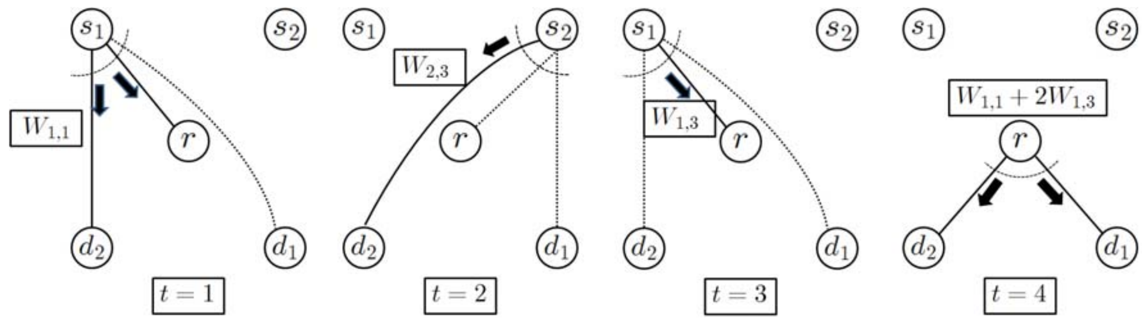


Fig. 3.1.: The illustration of the coding procedure in Example 3.1.1. We use a solid line to represent that the corresponding receiver has successfully received the packet and use a dot line to represent the case of erasure.

more and more packets that can be used to obtain/decode new information. We use the following example to illustrate the definitions of S_r to T_{d_2}.

Example 3.1.1 *Consider* $\mathsf{GF}(3)$ *and* $nR_1 = 3$ *and* $nR_2 = 2$*. That is, flow-1 contains 3 message symbols* $W_{1,1}$ *to* $W_{1,3}$ *and flow-2 contains 2 message symbols* $W_{2,1}$ *and* $W_{2,2}$*.* Ω_1 *and* Ω_2 *are thus 3-dimensional and 2-dimensional linear spaces in* $\mathsf{GF}(3)$*, respectively. Consider the first four time slots* $t = 1$ *to* 4 *for our discussion.*

When $t = 1$*, suppose that* s_1 *is scheduled; an uncoded flow-1 message symbol* $W_{1,1}$ *is transmitted; and the packet is heard by and only by* d_2 *and* r*. See Fig. 3.1(a) for illustration, for which we use the solid lines to represent that* d_2 *and* r *have received the packet. We use the dashed line to denote that* d_1 *does not receive the packet. When* $t = 2$*, suppose that* s_2 *is scheduled; an uncoded flow-2 message symbol* $W_{2,1}$ *is transmitted; and the packet is heard by and only by* d_2*, see Fig. 3.1(b). When* $t = 3$*, suppose that* s_1 *is scheduled; an uncoded flow-1 symbol* $W_{1,3}$ *is transmitted; and the packet is heard by and only by* r*. When* $t = 4$*, suppose that* r *is scheduled;* r *sends a linear combination* $[W_{1,1} + 2W_{1,3}]$ *of the two flow-1 packets it has received thus far; and the packet* $[W_{1,1} + 2W_{1,3}]$ *is heard by both* d_1 *and* d_2*. We now describe the six knowledge spaces* S_r *to* T_{d_2} *in the end of* $t = 4$*. By Figs. 3.1(a) and 3.1(d),* d_2 *has received two flow-1 packets* $W_{1,1}$ *and* $[W_{1,1} + 2W_{1,3}]$*, one from* s_1 *and one from* r*. Therefore, by the end of* $t = 4$*, the flow-1 knowledge space at* d_2 *becomes* $S_{d_2}(4) = \mathsf{span}\{(1, 0, 0), (1, 0, 2)\}$*. Also, neither* r *nor* d_1 *has received any flow-2*

Table 3.1: The resulting knowledge spaces at the end of Example 3.1.1.

Flow-1		Flow-2	
$S_{d_1}(4)$	$\mathsf{span}\{(1,0,2)\}$	$T_{d_2}(4)$	$\mathsf{span}\{(1,0)\}$
$S_r(4)$	$\mathsf{span}\{(1,0,0),(0,0,1)\}$	$T_r(4)$	$\{(0,0)\}$
$S_{d_2}(4)$	$\mathsf{span}\{(1,0,0),(1,0,2)\}$	$T_{d_1}(4)$	$\{(0,0)\}$

packets by the end of $t = 4$. Therefore, T_r and T_{d_1}, the flow-2 knowledge spaces at r and d_1, respectively, contain only the zero element. The other knowledge spaces S_{d_1}, S_r, and T_{d_2} in the end of $t = 4$ can be derived similarly and they are summarized in Table 3.1.

The above definitions also lead to the following self-explanatory lemma.

Lemma 3.1.1 *The two destinations d_1 and d_2 can decode the desired message symbols $\mathbf{W}_1$ and $\mathbf{W}_2$, respectively, if and only if by the end of time n*

$$S_{d_1}(n) = \Omega_1 \text{ and } T_{d_2}(n) = \Omega_2.$$

For simplicity, we use $S_i(t)$ and $T_i(t)$ to denote the knowledge space $S_{d_i}(t)$ and $T_{d_i}(t)$ for $i = 1, 2$. We also omit the input argument "(t)" if the time index is clear from the context. To conclude this subsection, we introduce the notation of the sum space $(A \oplus B) \triangleq \mathsf{span}\{\mathbf{v} : \forall \mathbf{v} \in A \cup B\}$. Notice that $A \oplus B$ and $A \cup B$ are different. For example, in a 2-dimensional linear space with $\mathsf{GF}(2)$, we assume $A = \mathsf{span}\{(1,0)\}$ and $B = \mathsf{span}\{(1,1)\}$. Then $A \cup B = \{(0,0),(1,0),(2,0),(1,1),(2,2)\}$, but $A \oplus B = \mathsf{span}\{(1,0),(1,1)\} = \{(0,0),(1,0),(2,0),(1,1),(2,2),(2,1),(1,2),(0,1),(0,2)\}$. By simple algebra, we have

Lemma 3.1.2 *For any two linear subspaces A and B in Ω, the following equality always holds.*

$$\mathrm{Rank}(A \oplus B) = \mathrm{Rank}(A) + \mathrm{Rank}(B) - \mathrm{Rank}(A \cap B).$$

3.2 An Instance of SBLNC Policies

In the following, we will introduce a new class of network codes, named the Space-Based Linear Network Code (SBLNC). An SBLNC scheme contains a finite number of *policies*. Each policy Γ contains a linear subspace $A^{(\Gamma)}$, named *the inclusion space/set*, and a finite collection of subspaces $B_l^{(\Gamma)}$ for $l = 1$ to $L^{(\Gamma)}$, named *the exclusion spaces/sets*. For each time slot t, the SBLNC chooses one of the specified policies and uses it to generate the coded packet. For example, say node s is scheduled for transmission and we decide to use a policy Γ for encoding. Then s will first choose arbitrarily a coding vector $\mathbf{v}^{(i)}$ from the set $A^{(\Gamma)} \backslash \left(\bigcup_{l=1}^{L^{(\Gamma)}} B_l^{(\Gamma)} \right)$, and then transmit a linearly encoded packet $X = \mathbf{v}^{(i)} \mathbf{W}_i^{\mathsf{T}}$. That is, the coding vector must be in the inclusion set $A^{(\Gamma)}$ but not in any of the exclusion sets $B_l^{(\Gamma)}$. Obviously, a policy can be used/chosen only when the corresponding set $A^{(\Gamma)} \backslash \left(\bigcup_{l=1}^{L^{(\Gamma)}} B_l^{(\Gamma)} \right)$ is non-empty. For notational simplicity, we say a policy is *feasible* if the corresponding $A^{(\Gamma)} \backslash \left(\bigcup_{l=1}^{L^{(\Gamma)}} B_l^{(\Gamma)} \right)$ is non-empty.

For illustration, consider the following policy for node s_1, named Policy $\Gamma_{s_1,0}$. When Policy $\Gamma_{s_1,0}$ is used/chosen, we let source node s_1 choose arbitrarily a coding vector $\mathbf{v}^{(1)}$ from $\Omega_1 \backslash (S_1 \oplus S_2 \oplus S_r)$ and send the corresponding coded packet $X_{s_1} = \mathbf{v}^{(1)} \mathbf{W}_1^{\mathsf{T}}$. That is, the inclusion set is $A^{(\Gamma_{s_1,0})} = \Omega_1$ and the exclusion set is $B^{(\Gamma_{s_1,0})} = S_1 \oplus S_2 \oplus S_r$.

Continue the example in Section 3.1 for which the knowledge spaces are summarized in Table 3.1. In the beginning of $t = 5$ (or equivalently in the end of $t = 4$), we have $A^{(\Gamma_{s_1,0})} = \Omega_1$ and $B_1^{(\Gamma_{s_1,0})} = S_1 \oplus S_2 \oplus S_r = \{(a, 0, c) : \forall a, c \in \mathsf{GF}(q)\}$. As a result, if we choose Policy $\Gamma_{s_1,0}$ for $t = 5$, any coding vectors of the form (a, b, c) with $b \neq 0$ are in the set $\Omega_1 \backslash (S_1 \oplus S_2 \oplus S_r)$. There are totally 18 such vectors since $\mathsf{GF}(3)$ is used. Source s_1 can then choose arbitrarily from any one of the 18 vectors and send $X = aW_{1,1} + bW_{1,2} + cW_{1,3}$ in time $t = 5$.

In the following, we define 13 policies that will be used to achieve the Shannon capacity of the 2-flow wireless butterfly network with packet erasure channels.

There are 5 policies governing the coding operations at source s_1, which are named Policy $\Gamma_{s_1,j}$ for $j = 0$ to 4. When Policy $\Gamma_{s_1,j}$ is used, s_1 sends $X_{s_1}(t) = \mathbf{v}^{(1)} \mathbf{W}_1^{\mathsf{T}}$ for some

$\mathbf{v}^{(1)}$. That is, source s_1 only mixes/encodes flow-1 packets together. In the following, we describe how to choose the vector $\mathbf{v}^{(1)}$ for each individual policy.

§ *Policy* $\Gamma_{s_1,0}$*:* Choose $\mathbf{v}^{(1)}$ arbitrarily from

$$\Omega_1 \backslash (S_1 \oplus S_2 \oplus S_r). \tag{3.1}$$

§ *Policy* $\Gamma_{s_1,1}$*:* Choose $\mathbf{v}^{(1)}$ arbitrarily from

$$(S_2 \oplus S_r) \backslash \left((S_1 \oplus S_r) \cup (S_1 \oplus S_2) \right). \tag{3.2}$$

§ *Policy* $\Gamma_{s_1,2}$*:* Choose $\mathbf{v}^{(1)}$ arbitrarily from

$$S_2 \backslash (S_1 \oplus S_r). \tag{3.3}$$

§ *Policy* $\Gamma_{s_1,3}$*:* Choose $\mathbf{v}^{(1)}$ arbitrarily from

$$S_r \backslash \left(S_1 \oplus (S_2 \cap S_r) \right). \tag{3.4}$$

§ *Policy* $\Gamma_{s_1,4}$*:* Choose $\mathbf{v}^{(1)}$ arbitrarily from

$$(S_2 \cap S_r) \backslash S_1. \tag{3.5}$$

Policy $\Gamma_{s_2,j}$, $j = 0$ to 4 are symmetric versions of *Policy* $\Gamma_{s_1,j}$ that concern source s_2 and mix/encode flow-2 packets instead. More explicitly, source s_2 sends $X_{s_2} = \mathbf{v}^{(2)} \mathbf{W}_2^{\mathsf{T}}$ for which the coding vector $\mathbf{v}^{(2)}$ is chosen according to the following specification.

§ *Policy* $\Gamma_{s_2,0}$*:* Choose $\mathbf{v}^{(2)}$ arbitrarily from

$$\Omega_2 \backslash (T_1 \oplus T_2 \oplus T_r). \tag{3.6}$$

§ *Policy* $\Gamma_{s_2,1}$*:* Choose $\mathbf{v}^{(2)}$ arbitrarily from

$$(T_1 \oplus T_r)\backslash\left((T_2 \oplus T_r) \cup (T_1 \oplus T_2)\right). \tag{3.7}$$

§ *Policy* $\Gamma_{s_2,2}$*:* Choose $\mathbf{v}^{(2)}$ arbitrarily from

$$T_1\backslash(T_2 \oplus T_r). \tag{3.8}$$

§ *Policy* $\Gamma_{s_2,3}$*:* Choose $\mathbf{v}^{(2)}$ arbitrarily from

$$T_r\backslash\left(T_2 \oplus (T_1 \cap T_r)\right). \tag{3.9}$$

§ *Policy* $\Gamma_{s_2,4}$*:* Choose $\mathbf{v}^{(2)}$ arbitrarily from

$$(T_1 \cap T_r)\backslash T_2. \tag{3.10}$$

There are 3 policies $\Gamma_{r,j}$, $j = 1, 2, 3$, governing the coding operations at the relay r, which are described as follows.

§ *Policy* $\Gamma_{r,1}$*:* The relay r chooses arbitrarily a vector $\mathbf{v}^{(1)}$ from

$$S_r\backslash\left((S_r \cap S_2) \oplus S_1\right) \tag{3.11}$$

and sends an intra-flow-coded flow-1 packet $X_r = \mathbf{v}^{(1)}\mathbf{W}_1^{\mathsf{T}}$.

§ *Policy* $\Gamma_{r,2}$*:* The relay r chooses arbitrarily a vector $\mathbf{v}^{(2)}$ from

$$T_r\backslash\left((T_r \cap T_1) \oplus T_2\right) \tag{3.12}$$

and sends an intra-flow-coded flow-2 packet $X_r = \mathbf{v}^{(2)}\mathbf{W}_2^{\mathsf{T}}$.

§ *Policy* $\Gamma_{r,3}$ is for the relay node r to send an interflow-coded packet $X_r = \mathbf{v}^{(1)}\mathbf{W}_1^{\mathsf{T}} +$

$\mathbf{v}^{(2)}\mathbf{W}_2^{\mathsf{T}}$, with $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$ chosen as follows: If $(S_2 \cap S_r)\backslash S_1$ is non-empty, choose $\mathbf{v}^{(1)}$ arbitrarily from

$$(S_2 \cap S_r)\backslash S_1, \tag{3.13}$$

otherwise choose $\mathbf{v}^{(1)} = \mathbf{0}$, a zero vector. If $(T_1 \cap T_r)\backslash T_2$ is non-empty, choose $\mathbf{v}^{(2)}$ arbitrarily from

$$(T_1 \cap T_r)\backslash T_2, \tag{3.14}$$

otherwise choose $\mathbf{v}^{(2)} = \mathbf{0}$.

Continue from Example 3.1.1 in Section 3.1 with the knowledge spaces in the end of $t = 4$ described in Table 3.1. Consider Policy $\Gamma_{s_1,3}$ as defined in (3.4). Since $S_2 \cap S_r = S_r$ in the end of $t = 4$, we have $S_r\backslash\,(S_1 \oplus (S_2 \cap S_r)) \subseteq S_r\backslash(S_2 \cap S_r) = \emptyset$ being an empty set. Thus, in contrast with the fact that Policy $\Gamma_{s_1,0}$ is feasible in the beginning of $t = 5$ as shown in our previous discussion, Policy $\Gamma_{s_1,3}$ is infeasible in the beginning of $t = 5$.

One can repeat the above analysis and verify that out of all 13 policies, only 4 of them are feasible in the beginning of $t = 5$, which are $\Gamma_{s_1,0}$, $\Gamma_{s_1,4}$, $\Gamma_{s_2,0}$, and $\Gamma_{r,3}$. The network code designer can thus apply one of the four policies in $t = 5$.

Suppose the network designer chooses policy $\Gamma_{s_1,0}$ for $t = 5$ and sends a flow-1 coded packet with the coding vector being $\mathbf{v}^{(1)} = (2, 1, 0)$. Also suppose that the packet is received by r but by neither d_1 nor d_2. Then in the end of time $t = 5$, the knowledge space S_r evolves from the original $\mathsf{span}\{(1, 0, 0), (0, 0, 1)\}$ to the new space $\mathsf{span}\{(1, 0, 0), (0, 0, 1), (2, 1, 0)\}$. We now notice that the Policy $\Gamma_{s_1,0}$ is no longer feasible since with the new S_r, the exclusion space of $\Gamma_{s_1,0}$ becomes $S_1 \oplus S_2 \oplus S_r = \mathsf{span}\{(1, 0, 0), (0, 0, 1), (2, 1, 0)\}$ and $\Omega_1\backslash\,(S_1 \oplus S_2 \oplus S_r)$ is now empty. On the other hand, the new S_r also lets some previously infeasible policies become feasible. For example, consider Policy $\Gamma_{s_1,3}$. With the new S_r, we have $S_r = \mathsf{span}\{(1, 0, 0), (0, 0, 1), (2, 1, 0)\}$ and $S_1 \oplus (S_2 \cap S_r) = \mathsf{span}\{(1, 0, 0), (1, 0, 2)\}$. Therefore, $S_r\backslash\,(S_1 \oplus (S_2 \cap S_r)) \neq \emptyset$. Policy $\Gamma_{s_1,3}$ is thus feasible and can be used for transmission in $t = 6$. With similar analysis, one can verify that in the beginning of $t = 6$, we have 5 feasible policies: $\Gamma_{s_1,3}$, $\Gamma_{s_1,4}$, $\Gamma_{s_2,0}$, $\Gamma_{r,1}$, and $\Gamma_{r,3}$.

3.3 The Design Motivations of SBLNC Policies

We conclude this chapter by discussing the design motivations behind the proposed 13 policies. We first consider the relay policies $\Gamma_{r,1}$ to $\Gamma_{r,3}$ due to its conceptual simplicity. We then discuss the source policies $\Gamma_{s_i,0}$ to $\Gamma_{s_i,4}$.

The Relay Policies

We first notice that for all relay r policies $\Gamma_{r,1}$, $\Gamma_{r,2}$, and $\Gamma_{r,3}$, the corresponding inclusion space is either a subspace of S_r or a subspace of T_r. The reason is that for node r to send a coded packet, the encoded packet must already be in S_r or T_r, the knowledge spaces of r. As a result, the transmitted vector $\mathbf{v}^{(1)}$ (or $\mathbf{v}^{(2)}$) must be drawn from a subset of S_r (or T_r).

It is clear that *a good network code should try to serve two flows simultaneously in order to maximize the throughput.* We now focus on Policy $\Gamma_{r,3}$. First notice that by (3.14), $\mathbf{v}^{(2)}$ is drawn from $(T_1 \cap T_r)$. This means that the value of $\mathbf{v}^{(2)}\mathbf{W}_2^\mathsf{T}$ is already known by destination d_1 since $\mathbf{v}^{(2)}$ being in the flow-2 knowledge space T_1 at d_1. Hence whenever d_1 receives the packet $X_r(t) = \mathbf{v}^{(1)}\mathbf{W}_1^\mathsf{T} + \mathbf{v}^{(2)}\mathbf{W}_2^\mathsf{T}$, it can extract its desired information and recover $\mathbf{v}^{(1)}\mathbf{W}_1^\mathsf{T}$ by substracting $\mathbf{v}^{(2)}\mathbf{W}_2^\mathsf{T}$. We then note that Policy $\Gamma_{r,3}$ ensures that whenever (3.13) is not empty the selected $\mathbf{v}^{(1)}$ is not in S_1, the flow-1 knowledge space at d_1. Hence upon the reception of such a coded packet, $\mathrm{Rank}(S_1)$ will increase by one. By Lemma 3.1.1 destination d_1 is one step closer to fully decode its desired message $\mathbf{W}_1$. Symmetrically, by (3.13) d_2 has already known the value of $\mathbf{v}^{(1)}\mathbf{W}_1^\mathsf{T}$ and thus d_2 can compute the value of $\mathbf{v}^{(2)}\mathbf{W}_2^\mathsf{T}$ upon the reception of the inter-flow coded packet generated by Policy $\Gamma_{r,3}$. Since $\mathbf{v}^{(2)}$ is not in T_2, d_2 can decode one extra linear combination of flow-2 packets. Policy $\Gamma_{r,3}$ thus serves both d_1 and d_2 simultaneously.

Although Policy $\Gamma_{r,3}$ can serve both destinations simultaneously, there is a limit on how much information can be sent by $\Gamma_{r,3}$. That is, if we use only Policy $\Gamma_{r,3}$ and nothing else, the information that can be received by d_1 through Policy $\Gamma_{r,3}$ is at most $(S_r \cap S_2)$ since all $\mathbf{v}^{(1)}$ are drawn from $(S_r \cap S_2)$. The largest flow-1 knowledge space that d_1 can possibly attain is thus $S_1 \oplus (S_r \cap S_2)$, where S_1 represents the flow-1 information that d_1 has accumulated by overhearing the transmission directly from its two-hop neighbor s_1, and $(S_r \cap S_2)$ represents the information that can be conveyed by $\Gamma_{r,3}$. Note that it is possible that S_r is not a subspace of $S_1 \oplus (S_r \cap S_2)$, which means that relay r still possesses some flow-1 information that cannot be conveyed to d_1 by $\Gamma_{r,3}$ alone. $\Gamma_{r,1}$ is devised to address this problem. That is, the $\mathbf{v}^{(1)}$ vector chosen from (3.11) is (i) from the knowledge space of r, and (ii) not in $S_1 \oplus (S_r \cap S_2)$, the largest flow-1 knowledge space that d_1 can attain when using exclusively Policy $\Gamma_{r,3}$. Such $\mathbf{v}^{(1)}$ vector thus represents an information packet that is complementary to the inter-flow-coded Policy $\Gamma_{r,3}$.

The Source Policies

Here without loss of generality we focus on source 1 policies. Even though the policies can be chosen arbitrarily whenever they are feasible, in the following discussion, we will explain in a way that they are executed sequentially from Policy $\Gamma_{s_1,0}$ to Policy $\Gamma_{s_1,4}$ to better catch up the intuitions.

Before explaining the source policies (3.1)–(3.5), we first discuss what we expect to achieve during the source 1 transmission. There is one thing we must achieve. Meanwhile, there are two things we desire to achieve. The one we must achieve is $\Omega_1 = (S_1 \oplus S_r)$ at the end of the source 1 transmission since $s_1 \to r \to d_1$ and $s_1 \to d_1$ are the only two routes from s_1 to d_1. For the two things we desire to achieve, the first one is $\Omega_1 = (S_1 \oplus (S_2 \cap S_r))$. The reason is that the condition, $\Omega_1 = (S_1 \oplus (S_2 \cap S_r))$, makes Policy $\Gamma_{r,1}$ always being infeasible, and hence we can exploit the coding benefit from Policy $\Gamma_{r,3}$. The second one is trivially $\Omega_1 = S_1$.

With those things mentioned above in mind, we then examine and explain each source policy. Policy $\Gamma_{s_1,j}$, $j = 0, 1, 2$, are designed for achieving $\Omega_1 = S_1 \oplus S_r$ with $S_1 \oplus S_r$ being a subset of the exclusion sets for these policies. This can be observed step by step. After finishing all possible transmissions from Policy $\Gamma_{s_1,0}$, we have $\Omega_1 = (S_1 \oplus S_2 \oplus S_r)$. After finishing all possible transmissions from Policy $\Gamma_{s_1,0}$ and $\Gamma_{s_1,1}$, it is either $(S_2 \oplus S_r) \subset (S_1 \oplus S_r)$ or $(S_2 \oplus S_r) \subset (S_1 \oplus S_2)$. In the first case, $\Omega_1 = (S_1 \oplus S_2 \oplus S_r) = (S_1 \oplus S_r)$. For the second case, $\Omega_1 = (S_1 \oplus S_2 \oplus S_r) = (S_1 \oplus S_2)$. Then with the help from Policy $\Gamma_{s_1,2}$, $S_2 \subset (S_1 \oplus S_r)$ after finishing all possible transmissions from Policy $\Gamma_{s_1,2}$, the second case in Policy $\Gamma_{s_1,1}$ comes to the result, $\Omega_1 = (S_1 \oplus S_2 \oplus S_r) = (S_1 \oplus S_2) = (S_1 \oplus S_r)$. So far we have examined that Policy $\Gamma_{s_1,j}$, $j = 0, 1, 2$, help us to achieve $\Omega_1 = (S_1 \oplus S_r)$. However, one may ask why we need three policies to achieve this instead of simply one policy $\Omega_1 \backslash (S_1 \oplus S_r)$. This question will be answered right after the discussion of achieving $\Omega_1 = (S_1 \oplus (S_2 \cap S_r))$.

Policy $\Gamma_{s_1,j}$, $j = 0, 1, 2, 3$, are designed for achieving $\Omega_1 = (S_1 \oplus (S_2 \cap S_r))$ with $S_1 \oplus (S_2 \cap S_r)$ being a subset of the exclusion sets for these policies. Similar to the above observation for $\Omega_1 = S_1 \oplus S_r$, one can verify that after finishing all possible transmissions from Policy $\Gamma_{s_1,0}$ to $\Gamma_{s_1,3}$, we have $\Omega_1 = (S_1 \oplus (S_2 \cap S_r))$. Furthermore, Policy $\Gamma_{s_1,1}$ is carefully designed for achieving $\Omega_1 = (S_1 \oplus (S_2 \cap S_r))$ in the most efficient way. To explain why Policy $\Gamma_{s_1,1}$ is the most efficient policy for achieving $\Omega_1 = (S_1 \oplus (S_2 \cap S_r))$, we first observe

$$
\begin{aligned}
&\mathrm{Rank}(S_1 \oplus (S_2 \cap S_r)) = \mathrm{Rank}(S_1) + \mathrm{Rank}(S_2 \cap S_r) - \mathrm{Rank}(S_1 \cap S_2 \cap S_r) \\
&= \mathrm{Rank}(S_1) + \mathrm{Rank}(S_2) + \mathrm{Rank}(S_r) - \mathrm{Rank}(S_2 \oplus S_r) - \mathrm{Rank}(S_1 \cap S_2 \cap S_r)
\end{aligned}
$$

Notice that $S_2 \oplus S_r$ is the inclusion set for Policy $\Gamma_{s_1,1}$; and S_1, S_2, and S_r are subsets of the exclusion sets for Policy $\Gamma_{s_1,1}$. Then with Policy $\Gamma_{s_1,1}$ being chosen, $\mathrm{Rank}(S_1 \oplus (S_2 \cap S_r))$ is expected to increase $p_1(d_1) + p_1(d_2) + p_1(r) - 0 - p_1(d_1 d_2 r)$. In this way, we maximize the positive terms and minimize the negative terms at the same time. This maximum increment of $\mathrm{Rank}(S_1 \oplus (S_2 \cap S_r))$ leads to the most efficient way for achieving $\Omega_1 = (S_1 \oplus (S_2 \cap S_r))$. This also answer the question why we need three policies for achieving $\Omega_1 = (S_1 \oplus S_r)$ instead of simply one policy, $\Omega_1 \backslash (S_1 \oplus S_r)$: Policy $\Gamma_{s1,0}$ sets up the environment for executing Policy $\Gamma_{s1,1}$, and Policy $\Gamma_{s1,2}$ helps Policy $\Gamma_{s1,1}$ back to achieving $\Omega_1 = (S_1 \oplus S_r)$. Finally, Policy $\Gamma_{s1,j}$, $j = 0, 1, 2, 3, 4$, are designed for achieving $\Omega_1 = S_1$ with similar observations.

To summarize, these designed source 1 policies help us to achieve $\Omega_1 = S_1 \oplus S_r$, $\Omega_1 = S_1 \oplus (S_2 \oplus S_r)$, and $\Omega_1 = S_1$ simultaneously and in the most efficient way.

3.4 Chapter Summary

In this chapter, we introduce the central idea of this thesis, the space-based linear network coding. We use the 2-flow wireless butterfly network with packet erasure channels as an example and construct the corresponding SBLNC scheme. In Section 3.1, we introduce the necessary definition for constructing SBLNC policies. In Section 3.2, we present an example of SBLNC scheme with 13 designed policies. Later we will use these 13 policies to achieve the Shannon capacity of the 2-flow wireless butterfly network with packet erasure channels. In Section 3.3, we discuss the design motivations of the SBLNC policies.

4. THE SHANNON CAPACITY OF WIRELESS BUTTERFLY NETWORK

In Section 2.2, we formulate a local network model including the broadcast packet erasure channel with feedback, the COPE principle, and the opportunistic routing. In this chapter, we will first discuss some related work and compare their settings with the one in Section 2.2. We then propose the corresponding outerbound and the inner bound which can be achieved by the SBLNC scheme described in Section 3.2. Finally we demonstrate the numerical results including the capacity region comparison and the sum rate throughput comparison.

4.1 Related Works

Recently, [3] and [4] successfully characterized the full capacity region of the 1-hop broadcast packet erasure channel with ≤ 3 coexisting flows. For comparison, our work focuses on the 2-hop network in Fig. 1.1(b) while [3] and [4] focus on the 1-hop broadcast channel. For the 2-hop network in Fig. 1.1(b), the network designer faces both the *scheduling problem*: which node (out of the two source nodes s_1, s_2, and the relay node r) to transmit at the current time slot, and the *network coding problem*: how to combine the heard/overheard packets and generate the network coded packets. For the 1-hop broadcast channel considered in [3] and [4], there is no scheduling problem since there is only one base station and the base station transmits all the time. As will be seen shortly, for a 2-hop erasure network, the feedback/control messages may propagate through the entire network and affect dynamically the scheduling and coding decisions for all three nodes s_1, s_2, and r, which further complicates the analysis.

Several attempts [17, 18] also have be made to approach the wireless butterfly network with packet erasure channels. Both of [17,18] take the side-information coding benefit into

Table 4.1: The feature comparison between this thesis and [18]

	Features in [18]	Features in this thesis
The outer bound	(1) Sequential scheduling, (2) Batched feedback, (3) Nonlinear coding functions.	(1) Dynamic scheduling, (2) Per-packet feedback, (3) Nonlinear coding functions.
The inner bound	(1) Sequential scheduling, (2) Batched feedback, (3) Linear coding functions.	(1) Sequential scheduling, (2) Per-packet feedback, (3) Linear coding functions.

concern. But [17] only proposed a suboptimal achievable scheme while [18] characterized the full capacity region. In the following, we will discuss the major differences between [18] and this thesis.

There are three major differences between the setting of this this thesis and in [18]. First, a deterministic sequential scheduling policy was used in [18], which schedules nodes s_1, s_2, and r in strict order. Namely, s_1 transmits first. After s_1 stops, s_2 can begin to transmit. Only after s_2 stops transmission can r start its own transmission. For comparison, our setting allows for dynamically choosing the schedule $\sigma(t)$ for each time slot t. Since we allow the schedule $\sigma(t)$ to depend on the past reception status $[\mathbf{Z}]_1^{t-1}$, the use of $\sigma(t)$ also includes any store-$\&$-forward-based scheduling policies as special cases, such as the back-pressure and the maximal weighted matching schemes (see [26] for references). Our results thus quantify the best achievable rates with jointly designed scheduling and coding policies.

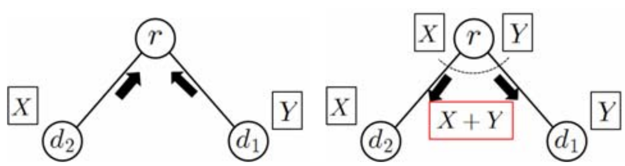
Secondly, in [18] no feedback is allowed when s_1 and s_2 transmit. More specifically, suppose jointly s_1 and s_2 take $t_{s_1} + t_{s_2}$ time slots to finish transmission. Then only in the beginning of time $(t_{s_1}+t_{s_2}+1)$ are we allowed to send the channel status $[\mathbf{Z}]_1^{t_{s_1}+t_{s_2}}$ to r. No further feedback is allowed until time n, the end of overall transmission. For comparison, our setting allows constantly broadcasting network-wide channel status $[\mathbf{Z}]_1^{t-1}$ to s_1, s_2, and r, as discussed in Section 2.2. This setting thus includes the Automatic Repeat reQuest (ARQ) mechanism as a special case [3, 18]. Broadcasting the control information $[\mathbf{Z}]_1^{t-1}$ to all the network nodes also eliminates the need of estimating/learning the reception status of the neighbors. Thirdly, [18] focuses on an arbitrary number of coexisting flows while this work focuses exclusively on the 2-flow scenario.

Last but not least, in the way of describing the inner bound in [18] and this work, they look similar because of the linear programming expressions and the use of the law of large number. However, they are essentially different. It is extremely difficult for the techniques in [18] to be extended to this work. The major obstacle is that in the setting of this thesis, the linear spaces constructed by the received packets keep evolving over time via per-packet feedback. Hence it is necessary to develop the SBLNC scheme to analysis the space evolution.

The practical COPE implementation contains three major components: (i) Opportunistic listening: Each destination is in a promiscuous monitoring mode and stores all the overheard packets; (ii) Opportunistic coding: The relay node decides which packets to be coded together opportunistically, based on the overhearing patterns of its neighbors; and (iii) Learning the states of the neighbors: Although in the practical COPE implementation reception reports are periodically sent to advertise the overhearing patterns of the next-hop neighbors of the relay, the relay node still needs to extrapolate the overhearing status of its neighbors since there is always a time lag due to the infrequent periodic feedback.

Our setting closely captures the opportunistic listening component of COPE by modeling the wireless packet transmission as a random broadcast PEC. In (2.1)–(2.3), the channel status vector is used to make the coding and scheduling decisions, which captures the opportunistic coding component of COPE. In COPE, the reception reports are broadcast periodically, which is captured by the control information $[\mathbf{Z}]_1^{t-1}$. In sum, our capacity region is a superset of any achievable rates of any COPE-principle-based schemes [5] when focusing on the 5-node 2-hop relay network in Fig. 1.1(b) with the node exclusive interference model.

Remark: The setting in Section 2.2 also includes the wireless erasure 2-way relay channel model (Fig. 4.1(a) and 4.1(b)) as a special case. Specifically, if we set the overhearing probabilities: $p_i(d_j) = 1$ for all $i \neq j$, then the capacity region of the setting in Section 2.2 is also the capacity region of the wireless erasure 2-way relay channel in Fig. 4.1.



(a) (b)

Fig. 4.1.: The illustration of the two–way relay channel for which s_1 sends X to s_2 and and s_2 sends Y to s_1. In (b), the common relay can send a linear combination $[X + Y]$ that benefits both destinations simultaneously.

4.2 Main Results

In this section, we provide our results based on two cases: The case of considering only the COPE principle and the case of combining COPE with the opportunistic routing technique. The main difference is that for the former setting, we assume that no transmission can be heard by its 2-hop neighbors, i.e., $p_i(d_i) = 0$ for all $i = 1, 2$. For the latter setting, we allow $p_i(d_i)$ to be non-zero.

For the case of using exclusively the COPE principle, the full capacity region has been characterized in Section 4.2.1 while for the case of COPE plus opportunistic routing, a pair of outer and inner bounds are provided in Sections 4.2.2 and 4.2.3, respectively.

4.2.1 COPE Principle Relay Network Capacity

Proposition 4.2.1 *Consider any 2-flow wireless butterfly network with packet erasure channels with $p_i(d_i) = 0$ for all $i = 1, 2$. The rate pair (R_1, R_2) is in the capacity region if and only if there exist three non-negative time sharing parameters t_{s_1}, t_{s_2} and t_r such that jointly (R_1, R_2) and (t_{s_1}, t_{s_2}, t_r) satisfy*

$$t_{s_1} + t_{s_2} + t_r \leq 1 \tag{4.1}$$

$$\forall i \in \{1, 2\}, R_i \leq t_{s_i} p_i(r) \tag{4.2}$$

$$\frac{R_1}{p_r(d_1)} + \frac{(R_2 - t_{s_2} p_2(d_1))^+}{p_r(d_1, d_2)} \leq t_r \tag{4.3}$$

$$\frac{(R_1 - t_{s_1} p_1(d_2))^+}{p_r(d_1, d_2)} + \frac{R_2}{p_r(d_2)} \leq t_r \tag{4.4}$$

where $(\cdot)^+ \triangleq \max(0, \cdot)$ is the projection to non-negative reals.

The proof of the achievability part of Proposition 4.2.1 is relegated to Section 4.3.2 and the converse proof is relegated to Appendix A.

The intuition behind (4.1) to (4.4) is as follows. (4.1) is a time sharing bound, which follows from the total time budget being n and the node-exclusive interference model.

Inequality (4.2) is a simple cut-set bound. That is, the message W_i has to be sent from s_i to the common relay r first. Therefore, the rate is upper bounded by the link capacity from s_i to r.

Inequality (4.3) and (4.4) combine the capacity results on message-side-information [18] and the capacity results on channel output feedback for broadcast channels [3, 4]. A very heuristic, not rigorous explanation of (4.3) is as follows. $\frac{R_1}{p_r(d_1)}$ represents how many time slots it takes to send all the flow-1 packets to d_1 as if there is no flow-2. $t_{s_2}p_2(d_1)$ characterizes how much flow-2 information can be "overheard" by d_1, and $(R_2 - t_{s_2}p_2(d_1))^+$ thus represents how much flow-2 information that has not been heard by d_1 but still needs to be sent to d_2. Since those flow-2 packets cannot be "coded" together with any flow-1 packets, they need to be sent separately by themselves in addition to the $\frac{R_1}{p_r(d_1)}$ time slots used to send flow-1 packets. In general, it takes $\frac{\left(R_2 - t_{s_2}p_2(d1)\right)^+}{p_r(d_2)}$ for those packets to arrive at d_2. However, [3] shows that the use of feedback can further reduce the time to $\frac{\left(R_2 - t_{s_2}p_2(d_1)\right)^+}{p_r(d_1,d_2)}$. As a result, (4.3) governs the transmission since the total transmission time of relay r is nt_r time slots. (4.4) is symmetric to (4.3).

4.2.2 Capacity Outer Bound for COPE plus OpR

The capacity results in Proposition 4.2.1 can be generalized as an outer bound for the case when the destination d_i may overhear directly the transmission of s_i, i.e., $p_i(d_i) > 0$.

Proposition 4.2.2 *Consider any 2-flow wireless butterfly network with packet erasure channels in Fig. 2.1(b) with arbitrary channel characteristics. If a rate vector (R_1, R_2) is achievable, there exist three non-negative scalars t_{s_1}, t_{s_2}, and t_r satisfying*

$$t_{s_1} + t_{s_2} + t_r \leq 1 \tag{4.5}$$

$$\forall i \in \{1, 2\}, \quad R_i \leq t_{s_i}p_i(d_i, r) \tag{4.6}$$

$$\frac{(R_1 - t_{s_1}p_1(d_1))^+}{p_r(d_1)} + \frac{(R_2 - t_{s_2}p_2(d_1, d_2))^+}{p_r(d_1, d_2)} \leq t_r \tag{4.7}$$

$$\frac{(R_1 - t_{s_1}p_1(d_1, d_2))^+}{p_r(d_1, d_2)} + \frac{(R_2 - t_{s_2}p_2(d_2))^+}{p_r(d_2)} \leq t_r. \tag{4.8}$$

This proposition can be proven by canonical techniques in the information theory outer bound problems as in [35]. And hence we put the detailed proof in Appendix A for reader's reference.

Remark: One can easily see that when the channel probabilities satisfy $p_i(d_i) = 0$ for all $i = 1, 2$, the outer bound in Proposition 4.2.2 collapses to the capacity region in Proposition 4.2.1. Proposition 4.2.2 is thus a strict generalization of the converse part of Proposition 4.2.1.

4.2.3 Capacity Inner Bound for COPE plus OpR

An inner bound for the general case of $p_i(d_i) \geq 0$ is described as follows.

Proposition 4.2.3 *A rate vector (R_1, R_2) is achievable by a linear network code if there exist 3 non-negative variables t_{s_1}, t_{s_2}, t_r, 10 non-negative variables, $\omega_{s_i}^k$, where $i \in \{1, 2\}$ and $k \in \{0, 1, 2, 3, 4\}$, 4 non-negative variables $\omega_{r,N}^k$, $\omega_{r,C}^k$ for $k = 1, 2$, such that jointly the 17 variables*[1] *and (R_1, R_2) satisfy the following four groups of inequalities:*

Group 1 has 5 inequalities, named the time budget constraints.

$$\forall i = 1, 2, \quad \sum_{k=0}^{4} \omega_{s_i}^k \leq t_{s_i} \tag{4.9}$$

$$\forall i = 1, 2, \quad \omega_{r,N}^1 + \omega_{r,N}^2 + \omega_{r,C}^i \leq t_r \tag{4.10}$$

$$t_{s_1} + t_{s_2} + t_r < 1 \tag{4.11}$$

[1]In the achieving algorithm in Section 4.3, the t variables correspond to the time slots that each of the sources and the relay is used; and the ω variables correspond to the time slots each policy is used.

Group 2 has 12 *inequalities, named* the packet conservation laws at the source nodes. *Consider any* $i, j \in \{1, 2\}$ *satisfying* $i \neq j$. *For each* (i, j) *pair (out of the two choices* $(1, 2)$ *and* $(2, 1)$*), we have the following 6 inequalities.*

$$\omega^0_{s_i} p_i(d_i, d_j, r) \leq R_i \tag{4.12}$$

$$\omega^1_{s_i} p_i(d_i, r) \leq \omega^0_{s_i} p_i(d_j \overline{d_i r}) \tag{4.13}$$

$$\omega^1_{s_i} p_i(d_i, d_j) \leq \omega^0_{s_i} p_i(r \overline{d_i d_j}) \tag{4.14}$$

$$\omega^2_{s_i} p_i(d_i, r) \leq \omega^0_{s_i} p_i(d_j \overline{d_i r}) - \omega^1_{s_i} p_i(d_i, r) \tag{4.15}$$

$$\omega^3_{s_i} p_i(d_i, d_j) \leq \omega^0_{s_i} p_i(r \overline{d_i d_j}) - \omega^1_{s_i} p_i(d_j, d_i r) \tag{4.16}$$

$$\begin{aligned} \omega^4_{s_i} p_i(d_i) \leq\ & \omega^0_{s_i} p_i(d_j r \overline{d_i}) \\ & + \omega^1_{s_i} (p_i(d_j) + p_i(r) - p_i(d_i d_j r)) \\ & + \omega^2_{s_i} p_i(r \overline{d_i}) + \omega^3_{s_i} p_i(d_j \overline{d_i}) \end{aligned} \tag{4.17}$$

Group 3 has 4 inequalities, named the packet conservation laws at the relay node. *For each* (i, j) *pair with* $i \neq j$*, we have the following 2 inequalities.*

$$\begin{aligned} \omega^i_{r,N} p_r(d_i, d_j) \leq\ & \omega^0_{s_i} p_i(r \overline{d_i d_j}) - \omega^1_{s_i} p_i(d_j, d_i r) \\ & - \omega^3_{s_i} p_i(d_i, d_j) \end{aligned} \tag{4.18}$$

$$\begin{aligned} \omega^i_{r,C} p_r(d_i) \leq\ & \omega^0_{s_i} p_i(d_j r \overline{d_i}) \\ & + \omega^1_{s_i} (p_i(d_j) + p_i(r) - p_i(d_i d_j r)) \\ & + \omega^2_{s_i} p_i(r \overline{d_i}) + \omega^3_{s_i} p_i(d_j \overline{d_i}) \\ & - \omega^4_{s_i} p_i(d_i) + \omega^i_{r,N} p_r(d_j \overline{d_i}) \end{aligned} \tag{4.19}$$

Group 4 has 2 inequalities, named the decodability conditions. *Consider* $i = 1, 2$. *For each* i*, we have the following inequality.*

$$\left(\sum_{k=0}^{4} \omega_{s_i}^k\right) p_i(d_i) + \left(\omega_{r,N}^i + \omega_{r,C}^i\right) p_r(d_i) \geq R_i \tag{4.20}$$

An heuristic but not rigorous explanation is as follows. The time budget constraints (4.9)–(4.11) describe the fact that the each transmitting node can only select policies within its own time budget and the overall time budget is one in ratio. The conservation laws (4.12)–(4.19) describe the fact that for one policy being eligible to be selected, it must be a non-empty set, which is equivalent to quantify the space dimensions of the policies. The decodability conditions describe the fact that to be able to reconstruct the required packets at two destinations, a certain amount of packets must be received by two destinations.

Proposition 4.2.3 will be proved by explicit construction of an achievability scheme based on the SBLNC scheme described in the next section. The detailed proof of Proposition 4.2.3 is relegated to Section 4.3.1.

4.3 Capacity Approaching Coding Scheme

In this section, we will first prove the capacity inner bound *Proposition 4.2.3* for the setting of the COPE principle when allowing opportunistic routing. We will then prove that the inner bound coincides with the capacity characterization in Proposition 4.2.1 for the COPE principle without the opportunistic routing component.

4.3.1 Achieving The Inner Bound of Proposition 4.2.3

We prove Proposition 4.2.3 by properly scheduling the 13 policies described in Section 3.2.

Consider any t_{s_1}, t_{s_2}, t_r, $\omega_{s_i}^k$, $i \in \{1, 2\}$ and $k \in \{0, 1, 2, 3, 4\}$, $\omega_{r,\mathrm{N}}^k$, and $\omega_{r,\mathrm{C}}^k$, $k = 1, 2$, satisfying the inequalities (4.9) to (4.20) in Proposition 4.2.3. For any $\epsilon > 0$, we can always construct another set of t' and ω' variables such that the new t' and ω' variables satisfy (4.9) to (4.19) with strict inequality, and satisfy the following inequality

$$\left(\sum_{k=0}^{4} \omega_{s_i}^{k}\right) p_i(d_i) + \left(\omega_{r,\mathrm{N}}^{i} + \omega_{r,\mathrm{C}}^{i}\right) p_r(d_i) > R_i - \epsilon \tag{4.21}$$

instead of (4.20). Based on the above observation, we will assume that the given t_{s_1}, t_{s_2}, t_r, $\omega_{s_i}^k$, $\omega_{r,\mathrm{N}}^k$, and $\omega_{r,\mathrm{C}}^k$ satisfy (4.9) to (4.19) and (4.21) with strict inequality. In the following, we will construct an SBLNC solution such that the scheme "properly terminates" within the allocated n time slots with close-to-one probability and after the SBLNC scheme stops, each d_i has received at least $n(R_i - \epsilon)$ number of its desired information packets.

We construct the SBLNC scheme as follows. We first schedule the s_1-policies sequentially from $\Gamma_{s_1,0}$ to $\Gamma_{s_1,4}$. Each policy $\Gamma_{s_1,k}$ lasts for $n \cdot \omega_{s_1}^k$ time slots. After finishing $\Gamma_{s_1,k}$ we move on to Policy $\Gamma_{s_1,k+1}$ until finishing all 5 s_1-policies. After finishing the s_1-policies, we move on to the s_2-policies. Again, we choose the s_2-policies sequentially from $\Gamma_{s_2,0}$ to $\Gamma_{s_2,4}$ and each policy lasts for $n \cdot \omega_{s_2}^k$ time slots. After the s_2-policies, we schedule the r-policies sequentially from $\Gamma_{r,1}$ to $\Gamma_{r,3}$. Policies $\Gamma_{r,1}$ and $\Gamma_{r,2}$ last for $n \cdot \omega_{r,\mathrm{N}}^1$ and $n \cdot \omega_{r,\mathrm{N}}^2$ time slots, respectively. Policy $\Gamma_{r,3}$ lasts for $n \cdot \max\{\omega_{r,\mathrm{C}}^1, \omega_{r,\mathrm{C}}^2\}$ time slots. Feedback is critical for the SBLNC scheme as it is used to decide the evolution of the knowledge spaces S_1, S_2,..., T_r, which in turn decides the sets in (3.1)–(3.14).

For the above construction, we first discuss its dependency on the finite field size q. Among the entire designed policies (3.1)–(3.14), observe that $\max L^{(\Gamma)}$ is two, which means there are at most two exclusion spaces for one policy. Thus for each of the designed policies being non-empty, the minimum requirement of q is no less than 2. To prove this statement, assume we have 3 linear spaces A, B, and C with the designed policy $A\backslash(B \cup C)$. For this policy being non-empty, it requires $q^{\mathrm{Rank}(A)} = |A| > |(B \cup C) \cap A|$. Furthermore, one can show that $\mathrm{Rank}(A) > \max\{\mathrm{Rank}(A \cap B), \mathrm{Rank}(A \cap C)\}$ implies $|A| > |A \cap B| + |A \cap C| - 1 (= q^{\mathrm{Rank}(A \cap B)} + q^{\mathrm{Rank}(A \cap C)} - 1) \geq |(B \cup C) \cap A|$ with $q \geq 2$. Thus as shown in [3], the feedback capacity of 2-user broadcast erasure channel can be achieved with $q = 2$. The scheme proposed here also works for $q = 2$.

To prove the correctness of the above construction, we need to show that the following two statements hold with close-to-one probability: (i) during each time slot, it is always possible to construct the desired coding vectors $\mathbf{v}^{(1)}$ (or $\mathbf{v}^{(2)}$). That is, we never schedule an infeasible policy throughout the operation; (ii) destination d_i can decode $n(R_i - \epsilon)$ of the desired information packets when the scheme terminates[2]. In addition to the above two statements, we will also prove that with close-to-one probability; and (iii) during the first $n \cdot \omega^1_{r,\mathrm{C}}$ (resp. $n \cdot \omega^2_{r,\mathrm{C}}$) time slots of scheduling $\Gamma_{r,3}$, the computed flow-1 vector $\mathbf{v}^{(1)}$ (resp. flow-2 vector $\mathbf{v}^{(2)}$) is not zero.

We first prove (ii) while assuming both (i) and (iii) are true. We notice that all the exclusion spaces of policies $\Gamma_{s_1,0}$ to $\Gamma_{s_1,4}$, and $\Gamma_{r,1}$ contain S_1 as a subset. As a result, all those packets carry some new flow-1 information that has not yet been received by d_1. If d_1 receives any of those packets, the rank of S_1 will increase by one. Similarly, during the first $n\omega^1_{r,\mathrm{C}}$ time slots of Policy $\Gamma_{r,3}$, the computed $\mathbf{v}^{(1)}$ vector does not belong to S_1, see (3.13). As a result, if d_1 receives any of those packets, the rank of S_1 will again increase by one. From the above reasoning, the expected value of $\mathrm{Rank}(S_1)$ in the end of the SBLNC scheme must satisfy

$$\begin{aligned}\mathsf{E}\{\mathrm{Rank}(S_1)\} =& p_1(d_1)\left(\sum_{k=0}^{4} n\omega^k_{s_1}\right) \\ &+ p_r(d_1)(n\omega^1_{r,\mathrm{N}} + n\omega^1_{r,\mathrm{C}}) \qquad (4.22)\\ >& n(R_1 - \epsilon) \qquad (4.23)\end{aligned}$$

where the right-hand side of (4.22) quantifies the expected number of packets received by d_1 during Policies $\Gamma_{s_1,0}$ to $\Gamma_{s_1,4}$, $\Gamma_{r,1}$, and the first $n\omega^1_{r,\mathrm{C}}$ time slots of $\Gamma_{r,3}$. (4.23) follows from (4.21). By the law of large number, $\mathrm{Rank}(S_1) > n(R_1 - \epsilon)$ with close-to-one probability when n is sufficiently large. The above inequality ensures that d_1 can decode $n(R_1 - \epsilon)$ of the flow-1 information packets at the end of the SBLNC scheme. By symmetry, d_2 can also

[2]As the existence guaranteed in Proposition 3, given t and ω variables satisfying inequality (4.9)–(4.20), inequalities (4.9)–(4.11) guarantee that we can finish transmission within the allocated n time slots.

decode $n(R_2 - \epsilon)$ of the flow-2 packets $\mathbf{W}_2$ in the end of time $t = n$. What remains to be shown is to prove that (i) and (iii) hold with close-to-one probability.

Next we prove (i) and (iii) by the first order analysis that assumes sufficiently large n. We first consider Policy $\Gamma_{s_1,0}$. For any time t, $\Gamma_{s_1,0}$ is a feasible policy if (3.1) is non-empty, which is equivalent to having

$$\begin{aligned}&\mathrm{Rank}(\Omega_1) - \mathrm{Rank}(\Omega_1 \cap (S_1 \oplus S_2 \oplus S_r)) \\ &= \mathrm{Rank}(\Omega_1) - \mathrm{Rank}(S_1 \oplus S_2 \oplus S_r) > 0. \end{aligned} \tag{4.24}$$

We first note that $\mathrm{Rank}(\Omega_1) = nR_1$ is a constant and does not change over time. Also note that $\mathrm{Rank}(S_1 \oplus S_2 \oplus S_r)$ increases monotonically over time since a node accumulates more "knowledge" over time. As a result, if we can prove that (4.24) holds in the end of the duration of (executing) Policy $\Gamma_{s_1,0}$, then throughout the entire duration of $\Gamma_{s_1,0}$, we can always find some $\mathbf{v}^{(1)}$ belong to (3.1).

To that end, we notice that when we choose $\Gamma_{s_1,0}$ as our coding policy, the coding vector $\mathbf{v}^{(1)}$ is chosen from (3.1). Since $\mathbf{v}^{(1)}$ does not belong to the exclusion space $S_1 \oplus S_2 \oplus S_r$, $\mathrm{Rank}(S_1 \oplus S_2 \oplus S_r)$ *increases by one if and only if at least one of* d_1, d_2, *and* r *receives the transmitted packet* $X_{s_1} = \mathbf{v}^{(1)}\mathbf{W}_1^{\mathsf{T}}$. Also note that in the beginning of Policy $\Gamma_{s_1,0}$, $\mathrm{Rank}(S_1 \oplus S_2 \oplus S_r) = 0$. As a result, in the end of the duration of $\Gamma_{s_1,0}$, we have

$$\begin{aligned}&\mathsf{E}\{\mathrm{Rank}(S_1 \oplus S_2 \oplus S_r)\} \\ &= 0 + n \cdot \omega_{s_1}^0 \cdot p_1(d_1, d_2, r) && (4.25)\\ &< nR_1 = \mathrm{Rank}(\Omega_1), && (4.26)\end{aligned}$$

where (4.25) follows from quantifying the expected number of time slots (out of totally $n\omega_{s_1}^0$ time slots) in which at least one of d_1, d_2, and r receives it. (4.26) follows from (4.12).

By the law of large numbers, (4.26) implies that (4.24) holds in the end of the duration of $\Gamma_{s_1,0}$ with close-to-one probability. As a result, with close-to-one probability Policy $\Gamma_{s_1,0}$ remains feasible during the assigned duration of $n \cdot \omega_{s_1}^0$ time slots.

We now consider Policy $\Gamma_{s_1,1}$. For any time t, $\Gamma_{s_1,1}$ is feasible if (3.2) is non-empty, which is equivalent to having

$$\begin{aligned}
q^{\mathrm{Rank}(S_2 \oplus S_r)} &= |S_2 \oplus S_r| > |(S_2 \oplus S_r) \cap ((S_1 \oplus S_r) \cup (S_1 \oplus S_2))| \\
&= |((S_2 \oplus S_r) \cap (S_1 \oplus S_r)) \cup ((S_2 \oplus S_r) \cap (S_1 \oplus S_2))| \\
\Leftrightarrow\ & \mathrm{Rank}(S_2 \oplus S_r) \\
&> \max\{\mathrm{Rank}((S_2 \oplus S_r) \cap (S_1 \oplus S_r)), \\
&\qquad \mathrm{Rank}((S_2 \oplus S_r) \cap (S_1 \oplus S_2))\}
\end{aligned} \tag{4.27}$$

where "$\Leftrightarrow$" holds assuming the underlying finite field $\mathsf{GF}(q)$ satisfying $q \geq 2$. When we choose Policy $\Gamma_{s_1,1}$ as our coding policy, the coding vector $\mathbf{v}^{(1)}$ is chosen from (3.2). Therefore, $\mathbf{v}^{(1)}$ must belong to the inclusion space $S_2 \oplus S_r$, which implies that no matter how many nodes in $\{d_1, d_2, r\}$ receive the packet, $\mathrm{Rank}(S_2 \oplus S_r)$ remains the same. Also note that similar to the case of $\Gamma_{s_1,0}$, $\mathrm{Rank}((S_2 \oplus S_r) \cap (S_1 \oplus S_r))$ and $\mathrm{Rank}((S_2 \oplus S_r) \cap (S_1 \oplus S_2))$ increase monotonically over time. As a result, if we can prove that (4.27) holds in the end of the duration of Policy $\Gamma_{s_1,1}$, then throughout the entire duration of $\Gamma_{s_1,1}$, we can always find some $\mathbf{v}^{(1)}$ belong to (3.2). The remaining task is thus to quantify the three different ranks $\mathrm{Rank}(S_2 \oplus S_r)$, $\mathrm{Rank}((S_2 \oplus S_r) \cap (S_1 \oplus S_r))$, and $\mathrm{Rank}((S_2 \oplus S_r) \cap (S_1 \oplus S_2))$ at the end of (the duration of) $\Gamma_{s_1,1}$. All the following discussions hold with close-to-one probability when focusing on the first order analysis of n.

First consider $\mathrm{Rank}(S_2 \oplus S_r)$. We know that $\mathrm{Rank}(S_2 \oplus S_r)$ remains the same during Policy $\Gamma_{s_1,1}$. Therefore, the value of $\mathrm{Rank}(S_2 \oplus S_r)$ is decided by how much it increases during $\Gamma_{s_1,0}$. Since any $\mathbf{v}^{(1)}$ in Policy $\Gamma_{s_1,0}$ does not belong to $S_2 \oplus S_r$ (see (3.1)), every time one of d_2 and r receives a packet of $\Gamma_{s_1,0}$, $\mathrm{Rank}(S_2 \oplus S_r)$ will increase by one. As a result, in the end of $\Gamma_{s_1,1}$ we have

$$\mathsf{E}\{\mathrm{Rank}(S_2 \oplus S_r)\} = n\omega_{s_1}^0 p_1(d_2, r) + n\omega_{s_1}^1 \cdot 0. \tag{4.28}$$

We now consider the first term $\mathrm{Rank}((S_2 \oplus S_r) \cap (S_1 \oplus S_r))$ in the max operation in (4.27). By Lemma 3.1.2, we can rewrite $\mathrm{Rank}((S_1 \oplus S_r) \cap (S_2 \oplus S_r))$ by

$$\begin{aligned} &\mathrm{Rank}((S_1 \oplus S_r) \cap (S_2 \oplus S_r)) \\ &= \mathrm{Rank}(S_2 \oplus S_r) + \mathrm{Rank}(S_1 \oplus S_r) - \mathrm{Rank}(S_1 \oplus S_2 \oplus S_r). \end{aligned} \tag{4.29}$$

The value of $\mathrm{Rank}(S_2 \oplus S_r)$ is quantified in (4.28). Since any $\mathbf{v}^{(1)}$ in Policy $\Gamma_{s_1,0}$ does not belong to $S_1 \oplus S_r$ (see (3.1)) and any $\mathbf{v}^{(1)}$ in Policy $\Gamma_{s_1,1}$ does not belong to $S_1 \oplus S_r$ either (see (3.2)), every time one of d_1 and r receives a packet of $\Gamma_{s_1,0}$ or $\Gamma_{s_1,1}$, $\mathrm{Rank}(S_1 \oplus S_r)$ will increase by one. In the end of $\Gamma_{s_1,1}$ we thus have

$$\mathsf{E}\{\mathrm{Rank}(S_1 \oplus S_r)\} = n\omega_{s_1}^0 \cdot p_1(d_1, r) + n\omega_{s_1}^1 \cdot p_1(d_1, r). \tag{4.30}$$

Similarly, since any $\mathbf{v}^{(1)}$ in Policy $\Gamma_{s_1,0}$ does not belong to $S_1 \oplus S_2 \oplus S_r$ (see (3.1)) and any $\mathbf{v}^{(1)}$ in Policy $\Gamma_{s_1,1}$ belongs to $S_1 \oplus S_2 \oplus S_r$ (see (3.2)), every time one of d_1, d_2, and r receives a packet of $\Gamma_{s_1,0}$, $\mathrm{Rank}(S_1 \oplus S_2 \oplus S_r)$ will increase by one. In the end of $\Gamma_{s_1,1}$ we thus have

$$\mathsf{E}\{\mathrm{Rank}(S_1 \oplus S_2 \oplus S_r)\} = n\omega_{s_1}^0 p_1(d_1, d_2, r) + n\omega_{s_1}^1 \cdot 0. \tag{4.31}$$

By (4.28), (4.29), (4.30), and (4.31), we can verify that (4.13) implies that $\mathrm{Rank}(S_2 \oplus S_r) > \mathrm{Rank}((S_2 \oplus S_r) \cap (S_1 \oplus S_r))$ in the end of Policy $\Gamma_{s_1,1}$. By swapping the roles of d_2 and r, symmetric arguments can be used to prove that (4.14) implies $\mathrm{Rank}(S_2 \oplus S_r) > \mathrm{Rank}((S_2 \oplus S_r) \cap (S_1 \oplus S_2))$ in the end of Policy $\Gamma_{s_1,1}$. Therefore, $\Gamma_{s_1,1}$ is feasible throughout its duration of $n\omega_{s_1}^1$ time slots.

Similar rank-comparison arguments can be used to complete the proof of (i) and (iii). The remaining derivation repeats similar steps described above, and hence is relegated to Appendix B. The proof of Proposition 4.2.3 is thus complete.

4.3.2 Capacity of COPE Principle 2-Flow Wireless Butterfly Network Without Opportunistic Routing

In this subsection we will prove that the capacity outer bound in Proposition 4.2.2 and the capacity inner bound in Proposition 4.2.3 coincide when destination d_i cannot directly hear from source s_i for $i = 1, 2$. Proposition 4.2.1 thus describes the exact COPE-principle 2-flow wireless butterfly network capacity region without opportunistic routing.

To complete the proof, we note that when $p_i(d_i) = 0$, for $i = 1, 2$, (4.12)–(4.20) of the inner bound in Proposition 4.2.3 is reduced to the following forms:[3]

$$\omega_{s_i}^0 p_i(d_j, r) \leq R_i, \tag{4.32}$$

$$\omega_{s_i}^1 p_i(r) \leq \omega_{s_i}^0 p_i(d_j\overline{r}), \tag{4.33}$$

$$\omega_{s_i}^1 p_i(d_j) \leq \omega_{s_i}^0 p_i(r\overline{d_j}), \tag{4.34}$$

$$\omega_{s_i}^2 p_i(r) \leq \omega_{s_i}^0 p_i(d_j\overline{r}) - \omega_{s_i}^1 p_i(r), \tag{4.35}$$

$$\omega_{s_i}^3 p_i(d_j) \leq \omega_{s_i}^0 p_i(r\overline{d_j}) - \omega_{s_i}^1 p_i(d_j), \tag{4.36}$$

$$\begin{aligned}\omega_{r,\mathrm{N}}^i p_r(d_i, d_j) \leq \omega_{s_i}^0 p_i(r\overline{d_j}) - \omega_{s_i}^1 p_i(d_j) \\ - \omega_{s_i}^3 p_i(d_j),\end{aligned} \tag{4.37}$$

$$\begin{aligned}\omega_{r,\mathrm{C}}^i p_r(d_i) \leq \omega_{s_i}^0 p_i(d_j r) + \omega_{s_i}^1 (p_i(d_j) + p_i(r)) \\ + \omega_{s_i}^2 p_i(r) + \omega_{s_i}^3 p_i(d_j) + \omega_{r,\mathrm{N}}^i p_r(d_j\overline{d_i}).\end{aligned} \tag{4.38}$$

and (4.20) becomes

$$p_r(d_i)(\omega_{r,\mathrm{N}}^i + \omega_{r,\mathrm{C}}^i) \geq R_i. \tag{4.39}$$

[3]Inequality (4.17) becomes trivial since the left-hand side of (4.17) becomes zero and the right-hand side of (4.17) is always non-negative.

The following lemma proves the tightness of the bounds when there is no 2-hop overhearing, i.e., $p_i(d_i) = 0$ for $i = 1, 2$.

Lemma 4.3.1 *For any 5-tuple $(R_1, R_2, t_1, t_2, t_r)$ satisfying the capacity outer bound (4.1)–(4.4), we can always find 14 companying variables $\omega^j_{s_i}, \omega^i_{r,N}, \omega^i_{r,C}$ for $i = 1, 2$ and $j = 0, 1, 2, 3, 4$, such that jointly the $14 + 5 = 19$ variables satisfy (4.9), (4.10), (4.32) to (4.39).*

Proof Given any $(R_1, R_2, t_{s_1}, t_{s_2}, t_r)$ satisfying (4.1)–(4.4), we construct $\left\{\omega^j_{s_i}, \omega^i_{r,\mathrm{N}}, \omega^i_{r,\mathrm{C}} : i \in \{1, 2\}, j \in \{0, 1, 2, 3, 4\}\right\}$ in the following way. For each pair $(i, j) = (1, 2)$ or $(2, 1)$, we define

$$
\begin{aligned}
\omega^0_{s_i} =& \frac{R_i}{p_i(d_j, r)}, \\
\omega^1_{s_i} =& R_i \left(\min\left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right), \\
\omega^2_{s_i} =& R_i \left(\frac{1}{p_i(r)} - \frac{1}{p_i(d_j)} \right)^+, \\
\omega^3_{s_i} =& \min\left\{ R_i \left(\frac{1}{p_i(d_j)} - \frac{1}{p_i(r)} \right)^+, t_{s_i} - \frac{R_i}{p_i(r)} \right\}, \\
\omega^4_{s_i} =& 0, \\
\omega^i_{r,\mathrm{N}} =& \frac{(R_i - t_{s_i} p_i(d_j))^+}{p_r(d_i, d_j)}, \\
\omega^i_{r,\mathrm{C}} =& \frac{R_i}{p_r(d_i)} - \frac{(R_i - t_{s_i} p_i(d_j))^+}{p_r(d_i, d_j)}.
\end{aligned}
$$

One can verify that the above assignment $\left\{R_1, R_2, t_{s_1}, t_{s_2}, t_r, \omega^j_{s_i}, \omega^i_{r,\mathrm{N}}, \omega^i_{r,\mathrm{C}} : i \in \{1, 2\}, j \in \{0, 1, 2, 3, 4\}\right\}$ satisfies (4.9), (4.10), (4.32) to (4.39). The detailed verification is relegated to Appendix C. The proof of Lemma 4.3.1 is thus complete. ■

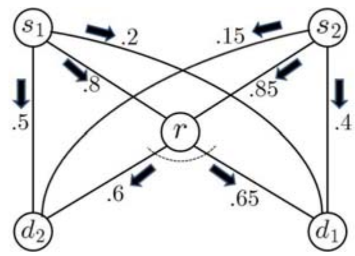


Fig. 4.2.: An instance of the 2-flow wireless butterfly network with the success probabilities being indicated next to the corresponding arrows. We also assume that the success events between different node pairs are independent.

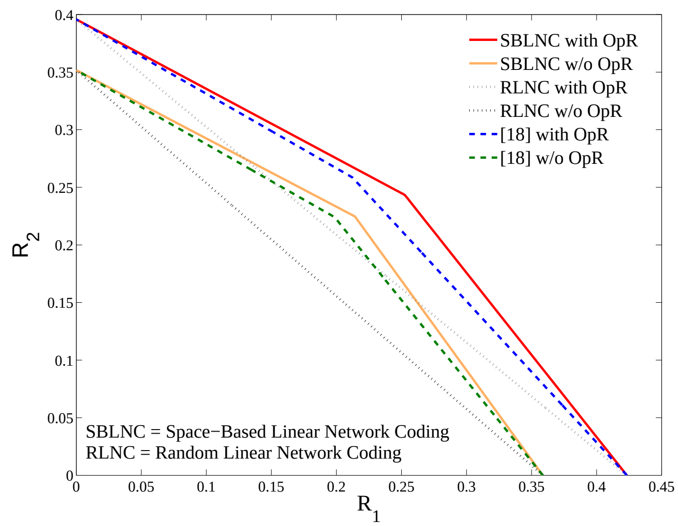


Fig. 4.3.: The capacity regions of the scenario in Fig. 4.2. The solid line indicates the throughput region of SBLNC. The dash line indicates the throughput region in [4]. The dot line indicates the throughput region of intra-session network coding (or random linear network coding).

4.4 Numerical Results

In this section, we apply the capacity results to some numerically generated scenarios so that we can explicitly quantify the throughput/capacity improvement of the COPE principle. The detailed simulation setting is described as follows.

Consider one specific setting of the 2-flow wireless butterfly network as depicted in Fig. 4.2. In Fig. 4.2, we specify the success transmission probability between each node pair as the number next to the corresponding arrow and we assume the success events between different node pairs are independent. For example, the probability that a packet sent by s_1 is heard by d_1 is $p_1(d_1) = .2$ and the probability that a packet sent by r is received by d_2 is $p_r(d_2) = .6$. We then compute 6 different capacity rate regions and plot them in Figure 4.3.

The solid line "SBLNC with OpR" represents the ultimate capacity region[4] of this network, for which relay r is allowed to perform inter-flow network coding across both flows and 2-hop overhearing directly from s_1 to d_1 (and from s_2 to d_2) is allowed. The curve "with COPE, w/o OpR" describes the capacity region when the relay r can perform inter-flow coding but there is no 2-hop overhearing. Both the curves "with COPE, with OpR" and "with COPE, w/o OpR" allow for optimal scheduling among s_1, s_2, and r. The dash line " [18] with (or w/o) OpR" represents the throughput region which can be achieved by the results in [18] with either opportunistic routing or not. The dot line "RLNC with (or w/o) OpR" represent the throughput region which can be achieved by simply the intra-session network coding (or random linear network coding [2]) with either opportunistic routing or not.

As can be seen, when there is only one flow in the network (say $R_2 = 0$), then OpR is optimal as was first established in [36]. However, when there are two coexisting flows (when both R_1 and $R_2 > 0$), the COPE principle alone sometimes outperforms OpR due to the strong overhearing between $s_2 \to d_1$ and $s_1 \to d_2$, $p_2(d_1) = 0.4$ and $p_1(d_2) = 0.5$, in this example. On the other hand, SBLNC provide the ultimate throughput when compared to the existing schemes.

We are also interested in quantifying the throughput benefits of COPE and OpR in a randomly placed network. To generate a typical XOR-in-the-air scenario, we first place the relay node in the center of a unit circle. Then we randomly place four nodes (s_1, s_2, d_1, d_2) inside the unit circle. To simulate the need of the relay for each session pair, we force the placement of each pair to be in the opposite 90 degree area. That is, d_i must be located

[4]Our main results provide a pair of outer and inner bounds for this capacity region. Since the gap between the inner and outer bounds is not visible in the figure (with relative gap less than 0.08%), we use the inner bound (the achievable rate) as the proxy of the capacity region.

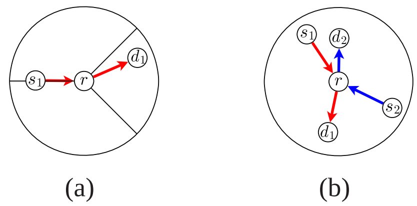


Fig. 4.4.: (a) The relative location of (s_i, d_i). (b) Topology of two (s_i, d_i) pairs.

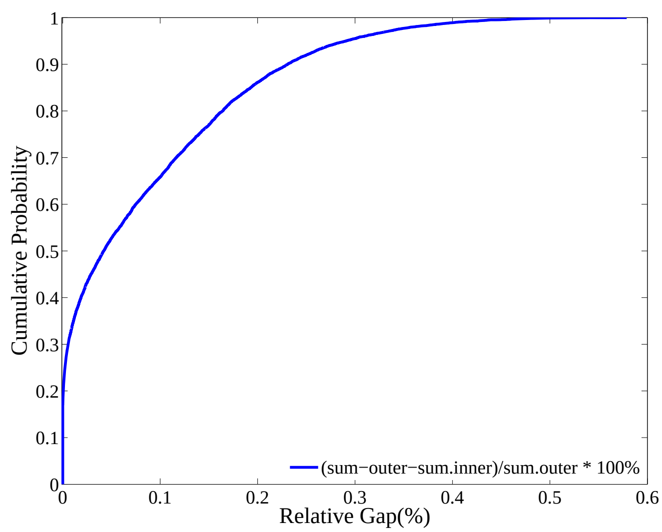


Fig. 4.5.: The cumulative distribution of the relative gap between the outer and the inner bounds when there is no fairness constraint. The outer and the inner bounds are described in Propositions 4.2.2 and 4.2.3, respectively.

Table 4.2: Average sum-rates over 10000 random node replacements.

Fairness Constraints	OpR	SBLNC	[18]	RLNC
No	allowed	.6599/.6594	.6472	.6180
	negligible	.4820	.4779	.4116
Proportional	allowed	.6294/.6286	.6101	.5484
	negligible	.4775	.4726	.3854
Min-cut	allowed	.6031/.6026	.5892	.5406
	negligible	.4671	.4626	.3856

in the opposite 90 degree area of s_i's location for $i = 1, 2$. See Fig. 4.4(a) for illustration. Fig. 4.4(b) illustrates one realization of our random node placement.

We use the Euclidean distance D between any two nodes to decide the overhearing probability when a packet is transmitted. More explicitly, we use the Rayleigh model

$$\mathsf{Prob}(\text{success}) = \int_{T^*}^{\infty} \frac{2x}{\gamma} e^{-\frac{x^2}{\gamma}} dx \quad \text{where } \gamma \triangleq \frac{1}{(4\pi)^2 D^{\alpha}},$$

where α is the path loss factor, and T^* is the decodable SNR threshold. To reflect the packet delivery ratio measured in practical environments, we choose $\alpha = 2.5$ and $T^* = 0.006$ so that the overhearing probability for a 1-hop neighbor is around 0.7–0.8 while overhearing probability for a 2-hop neighbor is around 0.2–0.3. If no direct overhearing is allowed, we simply hardwire the probability that d_i overhears s_i to be zero. We again assume that the success events between different node pairs are independent.

For practical concerns, we often enforce fairness constraint to avoid the situation that one of the flows occupies all of the allocated resource. Here we propose two kinds of fairness constraint: the min-cut fairness constraint, and the proportional fairness constraint. For the min-cut fairness constraint, we impose an additional constraint $R_i = \beta \min\left(p_i(d_i, r), p_i(d_i) + p_r(d_i)\right)$ for $i = 1, 2$ with a common β, which enforces the individual rate R_i being proportional to the min-cut value from s_i to d_i assuming no other sessions are transmitting and s_i and r are scheduled with the same frequency. For the proportional fairness constraint, we use $\log(R_1) + \log(R_2)$ as the objective function in the linear programming solver. When there is no fairness constraint, we simply maximize the sum rate $R_1 + R_2$ as the objective function in the linear programming solver. Combining the fairness constraint and the linear constraints in Proposition 4.2.2 or Proposition 4.2.3, we derive the optimal rates. We repeat the above experiment for 10000 times and lists the average sum rate $R_1 + R_2$ for each case in Table 4.2.

Table 4.2 lists the sum-rate averaged over 10000 simulations. When allowing 2-hop overhearing ($p_i(d_i) > 0$), then the inner and outer bounds do not always meet. Therefore, for the entry with both COPE and OpR, the number on the left is the average of the sum of the optimal rate $R_1 + R_2$ for the outer bound, denoted by $R_{\text{sum.outer}}$, while the number on the right is the average of the sum of the optimal rate $R_1 + R_2$ for the inner bound, denoted by $R_{\text{sum.inner}}$. When hardwiring the 2-hop overhearing probability to zero, as was proven Section 4.3.2, the sum-rate outer and inner bounds always coincide and hence only one number is shown in that entry. We also list the sum-rate inner bound results in [18]. The capacity of pure routing and pure OpR [36] can be explicitly computed and therefore there is only one number in those entries as well. We first note that in terms of the averaged throughput, the difference between the outer and the inner bounds is around 0.08%. Among all 10000 instances, the largest absolute difference is with $R_{\text{sum.outer}} = 0.6409$ and $R_{\text{sum.inner}} = 0.6375$. The proposed bounds thus effectively bracket the capacity when combining the XOR-in-the-air and the opportunistic routing principle. Jointly using COPE and OpR SBLNC scheme provides 60% throughput improvement over the classic pure routing scheme with optimal scheduling.

Fig. 4.5 focuses on the relative gap per experiment when allowing for both COPE and OpR. Specifically, we compute the relative gap per each experiment, $(R_{\text{sum.outer}} - R_{\text{sum.inner}})/R_{\text{sum.outer}}$ when there is no fairness constraint, and then plot the cumulative distribution function (cdf) for the relative gaps. We can see that with more than 80% of the experiments, the relative gap between the outer and inner bounds is smaller than 0.2%.

4.5 Chapter Summary

In this chapter, we discuss the capacity region of the local network formulated in Section 2.2. In Section 4.1, we compare the proposed model with existing results and demonstrate that the proposed model includes all the important features of the broadcast packet erasure channels, the COPE principle, and the opportunistic routing. In Section 4.2, we then characterize the full capacity region of the 2-flow wireless butterfly network without opportunistic routing, and propose the outer bound and the inner bound for the case with opportunistic routing. The SBLNC scheme is used to achieve the inner bound in Section 4.3. In Section 4.4, we use the numerical results to demonstrate how close the SBLNC scheme can approach the outer bound (and hence the optimal throughput) and the throughput provided by the SBLNC scheme strictly outperforms existing results.

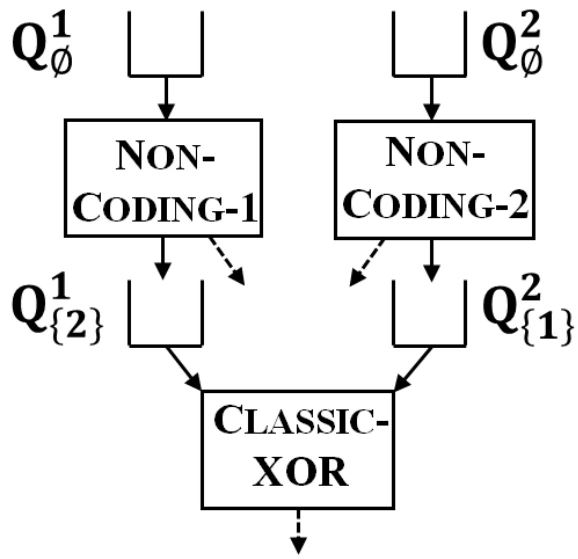
5. LINEAR NETWORK CODING SCHEDULING FOR 2-FLOW DOWNLINK TIME-VARYING BROADCAST PEC

Starting from this chapter, we turn our attention to the next step further, the stability region when considering the dynamic packet arrivals. The results in this and next chapters thus help us to establish closer relationship between the inter-session network coding solutions and the practical implementations. More specifically, we consider the downlink traffic from a base station to two different clients in the next two chapters.

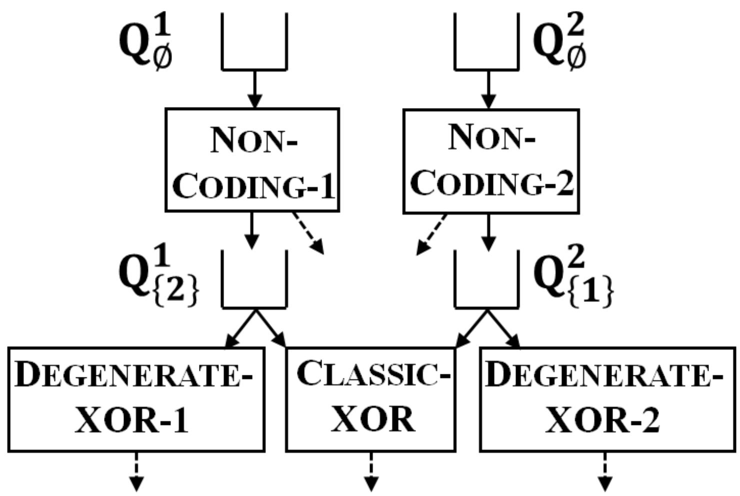
5.1 The Problem of Inter-Session NC Scheduling

Since 2000, NC has emerged as a promising technique in communication networks. The seminal work by [2] shows linear intra-session NC achieves the min-cut/max-flow capacity of single-session multi-cast networks. The natural connection of intra-session NC and the *maximum flow* allows the use of back-pressure (BP) algorithms to stabilize intra-session NC traffic, see [37] and the references therein.

However, when there are multiple coexisting sessions, the benefits of *inter-session* network coding (INC) are not fully utilized. The COPE architecture [5] demonstrated that a simple INC scheme can provide 40%–200% throughput improvement in a testbed environment. In the previous chapters, we also characterize the corresponding Shannon capacity. However, unlike the case of intra-session NC, there is no direct analogy from INC to the commodity flow. As a result, it is much more challenging to derive BP-based scheduling for INC traffic.



(a) INC using only 3 operations



(b) INC using only 5 operations

Fig. 5.1.: The virtual networks of two INC schemes.

5.1.1 An Illustration of The Issues And Challenges

We use the following example to illustrate this point. Consider a single source s and two destinations d_1 and d_2. Source s would like to send to d_1 the X_i packets, $i = 1, 2, \cdots$; and send to d_2 the Y_j packets, $j = 1, 2, \cdots$. The simplest INC scheme consists of three operations. OP1: Send uncodedly those X_i that have not been heard by any of $\{d_1, d_2\}$. OP2: Send uncodedly those Y_j that have not been heard by any of $\{d_1, d_2\}$. OP3: Send a linear sum $[X_i + Y_j]$ where X_i has been overheard by d_2 but not by d_1 and Y_j has been overheard by d_1 but not by d_2. For future reference, we denote OP1 to OP3 by NON-CODING-1, NON-CODING-2, and CLASSIC-XOR, respectively.

OP1 to OP3 can also be represented by the virtual network (vr-network) in Fig. 5.1(a). Namely, any newly arrived X_i and Y_j virtual packets[1] (vr-packets) that have not been heard by any of $\{d_1, d_2\}$ are stored in queues $Q^1_\emptyset$ and $Q^2_\emptyset$, respectively. The superscript $k \in \{1, 2\}$ indicates that the queue is for the session-k packets. The subscript $\emptyset$ indicates that those packets have not been heard by any of $\{d_1, d_2\}$. NON-CODING-1 then takes one X_i vr-packet from $Q^1_\emptyset$ and send it uncodedly. If such X_i is heard by d_1, then the vr-packet leaves the vr-network, which is described by the dotted arrow emanating from the NON-CODING-1 block. If X_i is overheard by d_2 but not d_1, then we place it in queue $Q^1_{\{2\}}$, the queue for the overheard session-1 packets. NON-CODING-2 in Fig. 5.1(a) can be interpreted symmetrically. CLASSIC-XOR operation takes an X_i from $Q^1_{\{2\}}$ and a Y_j from $Q^2_{\{1\}}$ and sends $[X_i + Y_j]$. If d_1 receives $[X_i + Y_j]$, then X_i is removed from $Q^1_{\{2\}}$ and leaves the vr-network. If d_2 receives $[X_i + Y_j]$, then Y_j is removed from $Q^2_{\{1\}}$ and leaves the vr-network.

It is known [30] that with dynamic packet arrivals, any INC scheme that (i) uses only these three operations and (ii) attains bounded decoding delay with rates (R_1, R_2) can be converted to a scheduling solution that stabilizes the vr-network with rates (R_1, R_2), and vice versa. *The INC design problem is thus converted to a vr-network scheduling problem.* To distinguish the above INC design for dynamical arrivals (the concept of stability regions) from the INC design assuming infinite backlog and decoding delay (the concept of the

[1]We denote the packets (jobs) inside the vr-network by “virtual packets.”

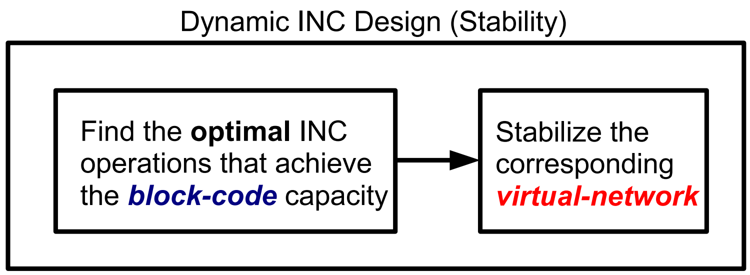


Fig. 5.2.: The two components of optimal dynamic INC design.

Shannon capacity), we term the former *the dynamic INC* design problem and the latter *the block-code INC* design problem.

The above vr-network representation also allows us to divide the optimal dynamic INC design problem into solving the following two major challenges separately. **Challenge 1**: The example in Fig. 5.1(a) focuses on dynamic INC schemes using only 3 possible operations. Obviously, the more INC operations one can choose from, the larger the degree of design freedom, and the higher the achievable throughput. *The goal is thus to find a (small) finite set of INC operations that can provably maximize the "block-code" achievable throughput*. **Challenge 2**: Suppose that we have found a set of INC operations that achieves the block-code capacity. However, it does not mean that such a set of INC operations always leads to a dynamic INC design since we still need to consider the delay/stability requirements. Specifically, once the optimal set of INC operations is decided, we can derive the corresponding vr-network. *The goal is then to devise a stabilizing scheduling policy for the vr-network, which leads to an equivalent representation of the optimal dynamic INC solution.* See Fig. 5.2 for the illustration of these two separate tasks.

Both tasks turn out to be highly non-trivial and optimal dynamic INC solution [30, 38, 39] has been designed only for the scenario of fixed channel quality. Specifically, [3] answers Challenge 1 and shows that for fixed channel quality, the 3 INC operations in Fig. 5.1(a) plus 2 additional DEGENERATE-XOR operations, see Fig. 5.1(b) and Section 5.3.1, can achieve the block-code INC capacity. One difficulty of resolving Challenge 2 is that an INC operation may involve multiple queues simultaneously, e.g., CLASSIC-XOR can only be scheduled when *both* $Q^1_{\{2\}}$ and $Q^2_{\{1\}}$ are non-empty. This is in sharp contrast with the traditional BP solutions in which each queue can act independently.[2] For the vr-network in Fig. 5.1(b), [38] circumvents this problem by designing a fixed priority rule that gives strict precedence to the CLASSIC-XOR operation. Alternatively, [30] derives a BP scheduling scheme by noticing that the vr-network in Fig. 5.1(b) can be decoupled into two vr-subnetworks (one for each data session) so that the queues in each of the vr-subnetworks can be activated independently and the traditional BP results follow.

5.1.2 Channel Quality Varies Over Time In Wireless Scenarios

However, the channel quality varies over time for practical wireless downlink scenarios. Therefore, one should opportunistically choose the most favorable users as receivers, the so-called *opportunistic scheduling* technique. Nonetheless, recently [40] shows that when allowing opportunistic coding+scheduling for time-varying channels, the 5 operations in Fig. 5.1(b) no longer achieve the block-code capacity. The existing dynamic INC design in [30, 38] are thus strictly suboptimal for time-varying channels since they are based on a suboptimal set of INC operations (recall Fig. 5.2).

5.1.3 Contribution Overview

In the next two chapters, we propose a new optimal dynamic INC design for 2-flow downlink traffic with time-varying packet erasure channels. Our detailed contributions are summarized as follows.

Contribution 1: We introduce a new pair of INC operations such that (i) The underlying concept is distinctly different from the traditional wisdom of XORing two overheard packets; (ii) The overall scheme uses only the ultra-low-complexity binary XOR operation; and

[2]Specifically, a critical assumption in [Section II C.1 [28]] is that if two queues Q_1 and Q_2 can be *activated* at the same time, then we can also choose to activate exactly one of the queues if desired. This is not the case in the vr-network. E.g., CLASSIC-XOR activates both $Q^1_{\{2\}}$ and $Q^2_{\{1\}}$ but no coding operation in Fig. 5.1(a) activates only one of $Q^1_{\{2\}}$ and $Q^2_{\{1\}}$.

(iii) The new set of INC operations is guaranteed to achieve the block-code-based Shannon capacity.

Contribution 2: The introduction of new INC operations leads to a new vr-network that is different from Fig. 5.1(b) and for which the existing "*vr-network decoupling* + *BP*" approach in [30] no longer holds. To answer Challenge 2, we generalize the results of Stochastic Processing Networks (SPNs) [31, 32] and apply it to the new vr-network. The end result is an opportunistic, dynamic INC solution that is completely *queue-length-based* and can robustly adapt to time-varying channels while achieving the largest possible stability region.

Contribution 3: The proposed solution can also be readily generalized for rate-adaptation. Through numerical experiments, we have shown that a simple extension of the proposed scheme can opportunistically and optimally choose the order of modulation and the rate of the error correcting codes used for each packet transmission while achieving the optimal stability region, i.e., equal to the Shannon capacity.

Contribution 4: A byproduct of our results is a scheduling scheme for SPNs with *random departure* instead of *deterministic departure*, which relaxes a major limitation of the existing SPN model. The results could thus further broaden the applications of SPN scheduling to other real-world scenarios.

5.1.4 Related Works

The most related existing works are [3, 30, 38, 39], which provides either a policy-based or a BP-based scheduling scheme for 2-user downlink networks and sometimes for more than 2 users. While they all achieve the 2-flow Shannon capacity of fixed channel quality, they are strictly suboptimal for time-varying channels and for rate-adaptation. Other works [41, 42] study the benefits of external side information while assuming the same setting of fixed-channel quality with no rate-adaptation.

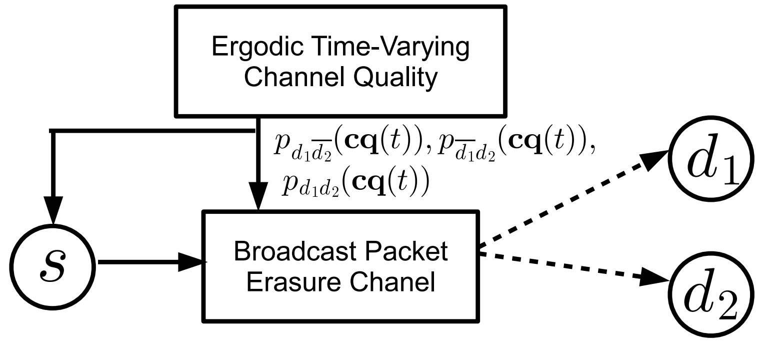


Fig. 5.3.: The time-varying broadcast packet erasure channel.

5.2 Problem Formulation

In this section, we formally introduce the problem setting for 2-flow downlink time-varying channels and the corresponding scheduling problem.

5.2.1 The Broadcast Erasure Channel With Dynamic Packet Arrivals

We model the 1-base-station/2-client downlink traffic as a broadcast packet erasure channel. See Fig. 5.3 for illustration. The detailed model description is as follows. Consider the following slotted transmission system.

Dynamic Arrival: In the beginning of every time t, there are $A_1(t)$ session-1 packets and $A_2(t)$ session-2 packets arriving at source s. We assume that $A_1(t)$ and $A_2(t)$ are i.i.d. integer-valued random variables with mean $(\mathsf{E}\{A_1(t)\}, \mathsf{E}\{A_2(t)\}) = (R_1, R_2)$ and bounded support. Recall that X_i and Y_j, $i, j \in \mathbb{N}$, denote the session-1 and session-2 packets, respectively.

Time-Varying Channel: We model the time-varying channel quality by a random process $\mathsf{cq}(t)$, which decides the reception probability of the *broadcast packet erasure channel.* In our proofs, we assume $\mathsf{cq}(t)$ is i.i.d. On the other hand, our numerical experiments show that the proposed scheme achieves the optimal stability region for any ergodic $\mathsf{cq}(t)$, say $\mathsf{cq}(t)$ being periodic.

Let CQ denote the support of $\mathsf{cq}(t)$ and we assume $|\mathsf{CQ}|$ is finite. For any $c \in \mathsf{CQ}$, we use f_c to denote the expected frequency of $\mathsf{cq}(t) = c$. We assume $f_c > 0$ for all $c \in \mathsf{CQ}$. Obviously $\sum_{c \in \mathsf{CQ}} f_c = 1$ since the total frequency is 1.

Broadcast Packet Erasure Channel: For each time slot t, source s can transmit one packet, which will be received by a random subset of destinations $\{d_1, d_2\}$. Specifically, there are 4 possible *reception status* $\{\overline{d_1 d_2}, d_1\overline{d_2}, \overline{d_1}d_2, d_1d_2\}$, e.g., the reception status $\mathsf{rcpt} = d_1\overline{d_2}$ means that the packet is received by d_1 but not d_2. The reception status probabilities can be described jointly by a vector $\vec{p} \triangleq (p_{\overline{d_1 d_2}}, p_{d_1\overline{d_2}}, p_{\overline{d_1}d_2}, p_{d_1d_2})$. For example, $\vec{p} = (0, 0.5, 0.5, 0)$ means that every time we transmit a packet, with 0.5 probability it will be received by d_1 only and with 0.5 probability it will be received by d_2 only. In contrast, if we have $\vec{p} = (0, 0, 0, 1)$, then it means that the packet is always received by d_1 and d_2 simultaneously. Since our model allows arbitrary joint probability vector $\vec{p}$, it captures the scenarios in which the erasure events of d_1 and d_2 are dependent, e.g., when the erasures at d_1 and d_2 are caused by a common (random) external interference source.

Opportunistic INC: Since the reception probability is decided by the channel quality, we write $\vec{p}(\mathsf{cq}(t))$ as a function of $\mathsf{cq}(t)$ at time t. In the beginning of time t, we assume that s is aware of the channel quality $\mathsf{cq}(t)$ (and thus knows $\vec{p}(\mathsf{cq}(t))$) so that s can opportunistically decide how to encode the packet for the current time slot. See Fig. 5.3. This opportunistic setting thus models the use of cognitive radio at source s.

ACKnowledgement: In the end of time t, both d_1 and d_2 will report back to s whether they have received the transmitted packet or not. This models the use of ACK.

5.3 Existing Results And The Issues

In this section, we introduce the important related results, and explain why these results could not be directly applied to our problem.

5.3.1 Existing Results on Block INC Design

References [40, 43] focus on the above setting but consider the infinite backlog block-code design instead of dynamic arrivals. Two findings of [40, 43] are summarized here.

The 5 INC operations in Fig. 5.1(b) are no longer optimal for time-varying channels

In Section 5.1, we have detailed 3 INC operations: NON-CODING-1, NON-CODING-2, and CLASSIC-XOR. Two additional INC operations are introduced in [3]: DEGENERATE-XOR-1 and DEGENERATE-XOR-2 as illustrated in Fig. 5.1(b). Specifically, DEGENERATE-XOR-1 is designed to handle the degenerate case in which $Q^1_{\{2\}}$ is non empty but $Q^2_{\{1\}} = \emptyset$. Namely, there is at least one X_i packet overheard by d_2 but there is no Y_j packet overheard by d_1. Not having such Y_j implies that one cannot send $[X_i + Y_j]$ (the CLASSIC-XOR operation). An alternative is thus to send the overheard X_i uncodedly (as if sending $[X_i + 0]$). We term this operation DEGENERATE-XOR-1. One can see from Fig. 5.1(b) that DEGENERATE-XOR-1 takes a vr-packet from $Q^1_{\{2\}}$ as input. If d_1 receives it, the vr-packet will leave the vr-network. DEGENERATE-XOR-2 is the symmetric version of DEGENERATE-XOR-1.

We use the following example to illustrate the sub-optimality of the above 5 operations. Suppose s has an X packet for d_1 and a Y packet for d_2 and consider a duration of 2 time slots. Also suppose that s knows beforehand that the time-varying channel will have (i) $\vec{p} = (0, 0.5, 0.5, 0)$ for slot 1; and (ii) $\vec{p} = (0, 0, 0, 1)$ for slot 2. The goal is to transmit as many packets in 2 time slots as possible.

Solution 1: INC based on the 5 operations in Fig. 5.1(b). In the beginning of time 1, both $Q^1_{\{2\}}$ and $Q^2_{\{1\}}$ are empty. Therefore, we can only choose either NON-CODING-1 or NON-CODING-2. Without loss of generality we choose NON-CODING-1 and thus send X uncodedly. Since $\vec{p} = (0, 0.5, 0.5, 0)$ in slot 1, there are only two cases to consider. Case 1: X is received only by d_1. In this case, we can send Y in the second time slot, which is guaranteed to arrive at d_2 since $\vec{p} = (0, 0, 0, 1)$ in slot 2. The total sum rate is sending 2 packets (X and Y) in 2 time slots. Case 2: X is received only by d_2. In this case, $Q^1_{\{2\}}$ contains one packet X, and $Q^2_{\emptyset}$ contains one packet Y, and all the other queues in Fig. 5.1(b) are empty. We can thus choose either NON-CODING-2 or DEGENERATE-XOR-1 for slot 2. Regardless of which coding operation we choose, slot 2 will then deliver 1 packet to either d_2 or d_1, depending on the INC operation we choose. Since no packet is delivered in slot 1, the total sum rate is 1 packet in 2 time slots. Since both cases have probability 0.5, the expected sum rate is $2 \cdot 0.5 + 1 \cdot 0.5 = 1.5$ packets in 2 time slots.

An optimal solution: We can achieve strictly better throughput by introducing new INC operations. Specifically, in slot 1, we send the linear sum $[X + Y]$ *even though neither X nor Y has ever been transmitted*, a distinct departure from the existing 5-operation-based solutions.

Again consider two cases: Case 1: $[X + Y]$ is received only by d_1. In this case, we let s send Y uncodedly in slot 2. Since $\vec{p} = (0, 0, 0, 1)$ in slot 2, the packet Y will be received by both d_1 and d_2. d_2 is happy since it has now received the desired Y packet. d_1 can use Y together with the $[X + Y]$ packet received in slot 1 to decode its desired X packet. Therefore, we deliver 2 packets (X and Y) in 2 time slots. Case 2: $[X + Y]$ is received only by d_2. In this case, we let s send X uncodedly in slot 2. By the symmetric argument of Case 1, we deliver 2 packets (X and Y) in 2 time slots. As a result, the sum-rate of the new solution is 2 packets in 2 slots, a 33% improvement over the existing solution.

Remark: This example focuses on a 2-time-slot duration due to the simplicity of the analysis. It is worth noting that the throughput improvement persists even for infinitely many time slots. See the simulations results in Section 6.4.

[40, 43] also derive the block-code capacity region

We summarize the high-level description of [43]:

Proposition 5.3.1 *[Propositions 1 and 3, [43]] For the block-code setting, a rate vector (R_1, R_2) can be achieved if and only if the corresponding linear programming (LP) problem is feasible. Given any (R_1, R_2), the LP problem of interest involves $18 \cdot |\mathsf{CQ}| + 7$ non-negative variables and $|\mathsf{CQ}| + 16$ (in-)equalities and can be explicitly computed.*

Our goal is to design a dynamic INC scheme, of which the stability region matches the block-code capacity region in Proposition 5.3.1.

5.3.2 Stochastic Processing Networks (SPNs)

The main tool that we use to stabilize the vr-network is stochastic processing networks (SPNs). In the following, we will discuss the basic definitions and existing results on SPNs.

The Main Feature of SPNs

The SPN is a generalization of the store-and-forward networks. In an SPN, a packet can not be transmitted directly from one queue to another queue through links. Instead, it must first be processed by a unit called "Service Activity" (SA). The SA first collects a certain amount of packets from one or more queues (named the *input queues*), jointly processes/consumes these packets, generates a new set of packets, and finally redistributes them to another set of queues (named the *output queues*). The number of consumed packets may be different than the number of generated packets. There is one critical rule for an SPN: *An SA can be activated only when all its input queues can provide enough amount of packets for the SA to process.* This rule captures directly the INC behavior and thus makes INC a natural application of SPNs. Other applications of SPNs include the video streaming problem [44] and the Map-&-Reduce scheduling problem [45].

SPNs with Deterministic Departure

All the existing SPN scheduling solutions [31, 32] assume a special class of SPNs, which we call SPNs with deterministic departure. We elaborate the detailed definition in the following.

Consider a time-slotted system with i.i.d. channel quality $\mathsf{cq}(t)$. An SPN consists of three components: the input activities (IAs), the service activities (SAs), and the queues. We suppose that there are K queues, M IAs, and N SAs.

Input Activities: Each IA represents a session (or a flow) of packets. Specifically, when an IA m is activated, it injects a deterministic number of $\alpha_{k,m}$ packets to queue k for a group of different k. Let $\mathcal{A} \in \mathbb{R}^{K*M}$ be the "input matrix" with the (k, m)-th entry equals to $\alpha_{k,m}$, for all m and k. At each time t, a random subset of IAs will be activated. Equivalently, we define $\mathbf{a}(t) \triangleq (a_1(t), a_2(t), \cdots, a_M(t)) \in \{0, 1\}^M$ as the random "arrival vector" at time t. If $a_m(t) = 1$, then IA m is activated at time t. We assume that the random vector $\mathbf{a}(t)$ is i.i.d. over time with the average rate vector $\mathbf{R} = \mathsf{E}\{\mathbf{a}(t)\}$. In our setting, the $\mathcal{A}$ matrix is a fixed (deterministic) system parameter and all the randomness of IAs lies in $\mathbf{a}(t)$.

Service Activities: For each service activity SA n, we define the *input queues* of SA n as the queues which are required to provide specified amounts of packets when SA n is activated. Let $\mathcal{I}_n$ denote the collection of the input queues of SA n. Similarly, we define the *output queues* of SA n as the queues which will possibly receive packets when SA n is activated, and let $\mathcal{O}_n$ be the collection of the output queues of SA n. That is, when SA n is activated, it takes packets from queues in $\mathcal{I}_n$, and sends packets to queues in $\mathcal{O}_n$. We assume that $\mathsf{cq}(t)$ does not change $\mathcal{I}_n$ and $\mathcal{O}_n$.

Let $\beta^{\text{in}}_{k,n}(c)$ be the number of packets from queue $k \in \mathcal{I}_n$ that will be consumed by SA n if SA n is activated under channel quality $\mathsf{cq}(t) = c$. Specifically, $\beta^{\text{in}}_{k,n}(c) \geq 0$ if queue k is the input queue of SA n (i.e. $k \in \mathcal{I}_n$), and we set $\beta^{\text{in}}_{k,n}(c) = 0$ otherwise. Similarly, let $\beta^{\text{out}}_{k,n}(c)$ be the number of packets received by queue k if SA n is activated under channel quality $\mathsf{cq}(t) = c$. Specifically, $\beta^{\text{out}}_{k,n}(c) \geq 0$ if queue $k \in \mathcal{O}_n$, and $\beta^{\text{out}}_{k,n}(c) = 0$ otherwise. Let $\mathcal{B}^{\text{in}}(c) \in \mathbb{R}^{K*N}$ be the *input service matrix* under channel quality c with the (k, n)-entry equals to $\beta^{\text{in}}_{k,n}(c)$, and let $\mathcal{B}^{\text{out}}(c) \in \mathbb{R}^{K*N}$ be the *output service matrix* under channel quality c with the (k, n)-entry equals to $\beta^{\text{out}}_{k,n}(c)$. For simplicity, we sometimes write $\mathcal{B}^{\text{in}}$ and $\mathcal{B}^{\text{out}}$ instead of $\mathcal{B}^{\text{in}}(c)$ and $\mathcal{B}^{\text{out}}(c)$. In the deterministic SPN setting, the matrices $\mathcal{B}^{\text{in}}(c)$ and $\mathcal{B}^{\text{out}}(c)$ are deterministic. The only random part is the arrival vector $\mathbf{a}(t)$ and the channel quality $\mathsf{cq}(t)$.

At the beginning of each time t, the SPN scheduler is made aware of the current channel quality $\mathsf{cq}(t)$ and can choose to "activate" a subset of the SAs. Let $\mathbf{x}(t) \in \{0, 1\}^N$ be the "service vector" at time t. If the n-th coordinate $\mathbf{x}_n(t) = 1$, then it implies that we choose to activate SA n at time t. To model the *interference constraint*, we require $\mathbf{x}(t)$ to be chosen from a pre-defined set of binary vectors $\mathfrak{X}$. Define Λ to be the convex hull of $\mathfrak{X}$ and let Λ° be the interior of Λ.

Acyclicness of The Underlying SPN: The input/outuput queues $\mathcal{I}_n$ and $\mathcal{O}_n$ of the SAs can be used to plot the corresponding SPN. We assume that the SPN is acyclic.

Existing results on the stability region of deterministic SPNs: We first introduce the following definitions.

Definition 5.3.1 *For the deterministic SPNs, an arrival rate vector* $\mathbf{R}$ *is "feasible" if there exist* $\mathbf{s}_c \in \Lambda$ *for all* $c \in \mathsf{CQ}$ *such that*

$$\mathcal{A} \cdot \mathbf{R}^{\mathsf{T}} + \sum_{c \in \mathsf{CQ}} f_c \cdot \mathcal{B}^{out}(c) \cdot \mathbf{s}_c^{\mathsf{T}} = \sum_{c \in \mathsf{CQ}} f_c \cdot \mathcal{B}^{in}(c) \cdot \mathbf{s}_c^{\mathsf{T}} \tag{5.1}$$

where $(\vec{v})^{\mathsf{T}}$ *is the transpose of the row vector* $\vec{v}$*. A rate vector* $\mathbf{R}$ *is "strictly feasible" if there exist* $\mathbf{s}_c \in \Lambda^\circ$ *for all* $c \in \mathsf{CQ}$ *such that (5.1) holds.*

Eq. (5.1) can be viewed as a flow conservation law of the deterministic SPN, for which the left-hand side describes the packets injected to queues 1 to k and the right-hand side corresponds to the packets leaving the queues.

Proposition 5.3.2 *[A combination of [31,32]] For deterministic SPNs, only feasible* $\mathbf{R}$ *can possibly be stabilized. Moreover, there exists an SPN scheduler that can stabilize all* $\mathbf{R}$ *that are strictly feasible.*

The achievability part for SPNs with deterministic departure (Proposition 5.3.2) is proven by the Deficit Max-Weight (DMW) algorithm in [31] and by the Perturb Max-Weight (PMW) algorithm in [32]. In the following, we briefly explain the existing DMW algorithm [31].

The Deficit Maximum Weight (DMW) Scheduling

In the DMW algorithm [31] for SPNs with deterministic departure, each queue k maintains a real-valued counter $q_k(t)$, called the virtual queue length. Initially, $q_k(1)$ is set to 0. For comparison, the actual queue length is denoted by $Q_k(t)$.

The key feature of a DMW algorithm is that it makes its decision based on $q_k(t)$ instead of $Q_k(t)$. Specifically, for each time t, we compute the "preferred[3] service vector" by

$$\mathbf{x}^*(t) = \arg\max_{\mathbf{x}\in\mathfrak{X}} \mathbf{d}^\mathsf{T}(t) \cdot \mathbf{x}, \tag{5.2}$$

where $\mathbf{d}(t)$ is the back pressure vector defined as $\mathbf{d}(t) = \left(\mathcal{B}^{\text{in}}(\mathsf{cq}(t)) - \mathcal{B}^{\text{out}}(\mathsf{cq}(t)\right)^\mathsf{T} \mathbf{q}(t)$, and $\mathbf{q}(t)$ is the vector of the virtual queue lengths. After computing $\mathbf{x}^*(t)$, we update $\mathbf{q}(t)$ according to the following flow conservation law:

$$\begin{aligned}\mathbf{q}(t+1) =& \mathbf{q}(t) + \mathcal{A}\cdot\mathbf{a}(t)\\ &+ \left(\mathcal{B}^{\text{out}}(\mathsf{cq}(t)) - \mathcal{B}^{\text{in}}(\mathsf{cq}(t))\right)\cdot\mathbf{x}^*(t).\end{aligned} \tag{5.3}$$

Unlike the actual queue lengths $Q_k(t)$, which is always ≥ 0, the virtual queue length $\mathbf{q}(t)$ can be smaller than 0 when updated via (5.3). That is, we do not need to take the projection to positive numbers when computing $\mathbf{q}(t)$.

It is worth emphasizing that the actual queue length still has to follow the SPN rule. That is, suppose SA n is the preferred service activity according to (5.2) but for at least one of its input queues, say queue k, the actual queue length $Q_k(t)$ is smaller than $\beta^{\text{in}}_{k,n}(\mathsf{cq}(t))$, the number of packets that are supposed to leave queue k. According to the model of SPN, we cannot schedule the preferred SA n due to the lack of enough packets in queue k. When this scenario happens, DMW simply skips activating SA n for this particular time slot, the system remains idle, and the actual queue length $Q_k(t+1) = Q_k(t)$. On the other hand,

[3]As we can see later, sometimes we may not be able to execute/schedule the preferred service activities chosen by (5.2). This is the reason why we only call the $\mathbf{x}^*(t)$ vector in (5.2) a preferred choice, instead of a scheduling choice.

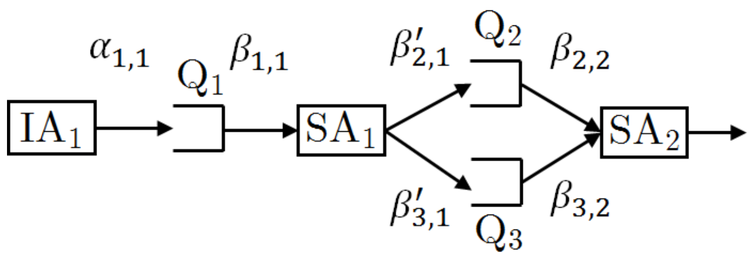


Fig. 5.4.: An SPN with random departure.

even though the system stays idle, the virtual queue length $\mathbf{q}(t)$ is still updated by (5.3). The above DMW algorithm is used to prove Propisition 5.3.2 in [31].

Open Problems for SPNs with Random Departure

Although the SPN with deterministic departure is relatively well understood, those SPN scheduling results cannot be applied to the INC vr-network. The reason is as follows. When a packet is broadcast by the base station, it can arrive at a random subset of receivers with certain probability distributions. Therefore, the vr-packets move among the vr-queues according to some probability distribution. This is not compatible with the deterministic departure SPN model, in which when an SA is activated we know *deterministically* $\beta^{\text{in}}_{k,n}(c)$ and $\beta^{\text{out}}_{k,n}(c)$, the service rates when the channel quality is $\mathsf{cq}(t) = c$. We call the SPN model that allows random $\beta^{\text{in}}_{k,n}(c)$ and $\beta^{\text{out}}_{k,n}(c)$ the SPN with random departure.

SPNs with random departure provide a unique challenge for the scheduling design. [31] provides the following example illustrating this issue. Fig. 5.4 describes an SPN with 6 transition edges. We assume IA1 is activated at every time slot and $\alpha_{1,1} = \beta^{\text{in}}_{1,1} = \beta^{\text{in}}_{2,2} = \beta^{\text{in}}_{3,2} = 1$. Namely, for every time t, $\alpha_{1,1} = 1$ packet will enter Q_1; in every time slot if we activate SA1, $\beta^{\text{in}}_{1,1} = 1$ packet will leave Q_1; if we activate SA2, $\beta^{\text{in}}_{2,2} = 1$ packet will leave Q_2 and $\beta^{\text{in}}_{3,2} = 1$ packet will leave Q_3. We assume these 4 transitions are deterministic but the two transitions SA1 $\rightarrow Q_2$ and SA1 $\rightarrow Q_3$ are random. Specifically, whenever SA1 is activated, it always takes a packet from Q_1. However, we flip a fair coin to decide whether the packet (generated from SA1) will go to Q_2 or Q_3. The random departure of SA1 implies that the queue length difference $|Q_2| - |Q_3|$ forms a binary random walk.

Note that SA2 has no impact on $|Q_2| - |Q_3|$ since it always takes 1 packet from each of the queues. The analysis of the random walk shows that $|Q_2| - |Q_3|$ goes unbounded with rate $\sqrt{t}$. And hence there is no scheduling algorithm which can stabilize both $|Q_2|$ and $|Q_3|$ simultaneously even though this example satisfies the flow-conservation law in (5.1) in the sense of expectation.

5.4 Chapter Summary

In this chapter, we discuss the INC scheduling problem for 2-flow downlink time-varying broadcast PEC. We first use an example to illustrate the issues and challenges. We then formally formulate the scheduling problem of 2-flow downlink time-varying broadcast PEC. We close this chapter by introducing the existing results and discuss why these results could not be applied to our problem.

6. ROBUST AND OPTIMAL OPPORTUNISTIC SCHEDULING FOR DOWNLINK 2-FLOW NETWORK CODING WITH VARYING CHANNEL QUALITY AND RATE ADAPTION

In this chapter, we propose the first opportunistic INC + scheduling solution that is provably optimal for time-varying channels, i.e., the corresponding stability region matches the optimal Shannon capacity. Specifically, we first introduce a new *binary INC* operation, which is distinctly different from the traditional wisdom of XORing two overheard packets. We then develop a *queue-length-based* scheduling scheme, which, with the help of the new INC operation, can robustly and optimally adapt to time-varying channel quality. We then show that the proposed algorithm can be easily extended for rate adaptation and it again robustly achieves the optimal throughput.

A byproduct of our results is a scheduling scheme for stochastic processing networks (SPNs) with *random departure*, which relaxes the assumption of *deterministic departure* in the existing results. The new SPN scheduler could thus further broaden the applications of SPN scheduling to other real-world scenarios.

6.1 The Proposed New INC Solution

In Section 5.2, we discuss the limitations of the existing works on the INC block code design and on the schedulers for SPNs, separately. In this section, we describe our new low-complexity binary INC scheme that achieves the block code capacity. In Section 6.2, we present our new scheduler design for the SPN with random departure. In Section 6.3, we will combine the proposed solutions to form the optimal dynamic INC design, see Fig. 5.2. For the new block code design in this section, we first describe the encoding steps and then discuss the decoding steps and buffer management.

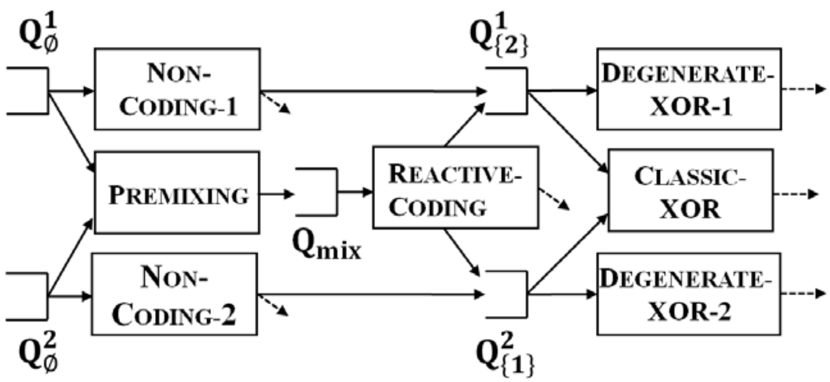


Fig. 6.1.: The virtual network of the proposed new INC solution.

6.1.1 Encoding

The proposed new INC solution is described as follows. We build upon the existing 5 operations, NON-CODING-1, NON-CODING-2, CLASSIC-XOR, DEGENERATE-XOR-1, and DEGENERATE-XOR-2. See Fig. 5.1(b) and the discussion in Sections 5.1 and 5.3.1. In addition, we add 2 more operations, termed PREMIXING and REACTIVE-CODING, respectively, and 1 new virtual queue, termed Q_{mix}. We plot the vr-network of the new scheme in Fig. 6.1. From Fig. 6.1, we can clearly see that PREMIXING involves both $Q^1_\emptyset$ and $Q^2_\emptyset$ as input and outputs to Q_{mix}. REACTIVE-CODING involves Q_{mix} as input and outputs to $Q^1_{\{2\}}$ or $Q^2_{\{1\}}$ or simply lets the vr-packet leave the vr-network (described by the dotted arrow). For every time instant, we can choose one of the 7 operations and the goal is to stabilize the vr-network. In the following, we describe in details how these two INC operations work and how to integrate them with the other 5 operations. Our description contains 4 parts.

Part I: The two operations, NON-CODING-1 and NON-CODING-2, remain the same.

Part II: We now describe the new operation PREMIXING. We can choose PREMIXING only if both $Q^1_\emptyset$ and $Q^2_\emptyset$ are non-empty. Namely, there are X_i packets and Y_j packets that have not been heard by any of d_1 and d_2. Whenever we schedule PREMIXING, we choose one X_i from $Q^1_\emptyset$ and one Y_j from $Q^2_\emptyset$ and send $[X_i + Y_j]$. If neither d_1 nor d_2 receives it, both X_i and Y_j remain in their original queues.

If at least one of $\{d_1, d_2\}$ receives it, we do the following. We remove *both* X_i and Y_j from their individual queues. We insert a tuple $(\mathsf{rcpt}; X_i, Y_j)$ into Q_{mix}. That is, unlike the other queues for which each entry is a single vr-packet, each entry of Q_{mix} is a tuple.

The first coordinate of $(\mathsf{rcpt}; X_i, Y_j)$ is rcpt, the reception status of $[X_i + Y_j]$. For example, if $[X_i + Y_j]$ was received by d_2 but not by d_1, then we set/record $\mathsf{rcpt} = \overline{d_1}d_2$; If $[X_i + Y_j]$ was received by both d_1 and d_2, then $\mathsf{rcpt} = d_1d_2$. The second and third coordinates store the participating packets X_i and Y_j separately. The reason why we do not store the linear sum directly is due to the new REACTIVE-CODING operation.

Part III: We now describe the new operation REACTIVE-CODING. For any time t, we can choose REACTIVE-CODING only if there is at least one tuple $(\mathsf{rcpt}; X_i, Y_j)$ in Q_{mix}.

Table 6.1: A summary of the REACTIVE-CODING operation

Current Reception Status $\mathbf{rcpt}(t)$	Departure	Insertion		
		$\mathbf{rcpt}^*{=}\,d_1\overline{d_2}$ (Send Y_j^*)	$\mathbf{rcpt}^*{=}\,\overline{d_1}d_2$ (Send X_i^*)	$\mathbf{rcpt}^*{=}\,d_1d_2$ (Send X_i^*)
$d_1\overline{d_2}$	Remove the tuple from Q_{mix}	Insert Y_j^* to $Q^2_{\{1\}}$	Insert X_i^* to $Q^2_{\{1\}}$	Insert Y_j^* to $Q^2_{\{1\}}$
$\overline{d_1}d_2$		Insert Y_j^* to $Q^1_{\{2\}}$	Insert X_i^* to $Q^1_{\{2\}}$	Insert X_i^* to $Q^1_{\{2\}}$
d_1d_2		Do not insert any packet back into the virtual network.		
$\overline{d_1d_2}$	Do nothing			

Choose one tuple from Q_{mix} and denote it by $(\mathsf{rcpt}^*; X_i^*, Y_j^*)$. We now describe the encoding part of REACTIVE-CODING.

Whenever we schedule REACTIVE-CODING, if $\mathsf{rcpt}^* = d_1\overline{d_2}$, send Y_j^*. If $\mathsf{rcpt}^* = \overline{d_1}d_2$, send X_i^*. If $\mathsf{rcpt}^* = d_1d_2$, send X_i^*. One can see that the coding operation depends on the reception status rcpt^* when $[X_i^* + Y_j^*]$ was first transmitted. This is why it is named REACTIVE-CODING.

The movement of the vr-packets depends on the current reception status of time t, denoted by $\mathsf{rcpt}(t)$, and also on the old reception status rcpt^* when the sum $[X_i^* + Y_j^*]$ was originally transmitted. The detailed movement rules are described in Table 6.1. The way to interpret the table is as follows. For example, when $\mathsf{rcpt}(t) = \overline{d_1d_2}$, i.e., neither d_1 nor d_2 receives the current transmission, then we do nothing, i.e., keep the tuple inside Q_{mix}. On the other hand, we remove the tuple from Q_{mix} whenever $\mathsf{rcpt}(t) \in \{d_1\overline{d_2}, \overline{d_1}d_2, d_1d_2\}$. If $\mathsf{rcpt}(t) = d_1d_2$, then we remove the tuple but do not insert any vr-packet back to the vr-network, see the second last row of Table 6.1. The tuple essentially leaves the vr-network in this case. If $\mathsf{rcpt}(t) = d_1\overline{d_2}$ and $\mathsf{rcpt}^* = d_1d_2$, then we remove the tuple from Q_{mix} and insert Y_j^* to $Q^2_{\{1\}}$. The rest of the combinations can be read from Table 6.1 in the same way. One can verify that the optimal INC example introduced in Section 5.3.1 is a direct application of the PREMIXING and REACTIVE-CODING operations.

Before we continue describing the slight modification to CLASSIC-XOR, DEGENERATE-XOR-1, and DEGENERATE-XOR-2, we briefly explain why the combination of PREMIXING and REACTIVE-CODING works. To facilitate discussion, we call the time slot in which we use PREMIXING to transmit $[X_i^* + Y_j^*]$ "slot 1" and the time slot in which we use REACTIVE-CODING "slot 2," even though the coding operations PREMIXING and REACTIVE-CODING may not be scheduled in two adjacent time slots. Using this notation, if $\mathsf{rcpt}^* = d_1\overline{d_2}$ and $\mathsf{rcpt}(t) = d_1d_2$, then it means that d_1 receives $[X_i^* + Y_j^*]$ and Y_j^* in slots 1 and 2, respectively and d_2 receives Y_j^* in slot 2. In this case, d_1 can decode the desired X_i^* and d_2 directly receives the desired Y_j^*. We now consider the perspective of the vr-network. Table 6.1 shows that the tuple will be removed from Q_{mix} and leave the vr-network. Therefore, no queue in the vr-network stores any of X_i^* and Y_j^*. This correctly reflects the fact that both X_i^* and Y_j^* have been received by their intended destinations.

Another example is when $\mathsf{rcpt}^* = \overline{d_1}d_2$ and $\mathsf{rcpt}(t) = d_1\overline{d_2}$. In this case, d_2 receives $[X_i^* + Y_j^*]$ in slot 1 and d_1 receives X_i^* in slot 2. From the vr-network's perspective, the movement rule (see Table 6.1) removes the tuple from Q_{mix} and insert an X_i^* packet to $Q^2_{\{1\}}$. Since a vr-packet is removed from a session-1 queue[1] Q_{mix} and inserted to a session-2 queue $Q^2_{\{1\}}$, the total number of vr-packets in the session-1 queue decreases by 1. This correctly reflects the fact that d_1 has received 1 desired packet X_i^* in slot 2.

An astute reader may wonder why in this example we can put X_i^*, a session-1 packet, into a session-2 queue $Q^2_{\{1\}}$. The reason is that whenever d_2 receives X_i^* in the future, it can recover its desired Y_j^* by subtracting X_i^* from the linear sum $[X_i^* + Y_j^*]$ it received in slot 1 (recall that $\mathsf{rcpt}^* = d_1\overline{d_2}$.) Therefore, X_i^* is now information-equivalent to Y_j^*, a session-2 packet. Moreover, d_1 has received X_i^*. Therefore, in terms of the information it carries, X_i^* is no different than a session-2 packet that has been overheard by d_1. As a result, it is fit to put X_i^* in $Q^2_{\{1\}}$.

Part IV: We now describe the slight modification to CLASSIC-XOR, DEGENERATE-XOR-1, and DEGENERATE-XOR-2. A unique feature of the new scheme is that some packets in $Q^2_{\{1\}}$ may be an X_i^* packet that is inserted by REACTIVE-CODING when $\mathsf{rcpt}^* = \overline{d_1}d_2$ and $\mathsf{rcpt}(t) = d_1\overline{d_2}$. (Also some $Q^1_{\{2\}}$ packets may be Y_j^*.) However, in our previous discussion, we have shown that those X_i^* in $Q^2_{\{1\}}$ is information-equivalent to a Y_j^* packet overheard by d_1. Therefore, in the CLASSIC-XOR operation, we should not insist on

[1]Q_{mix} is regarded as both a session-1 and a session-2 queue simultaneously.

Table 6.2: A summary of the transition probability of the virtual network in Fig. 6.1, where $p_{d_1 \vee d_2} \triangleq p_{d_1\overline{d_2}} + p_{\overline{d_1}d_2} + p_{d_1d_2}$; $p_{d_1} \triangleq p_{d_1\overline{d_2}} + p_{d_1d_2}$; NC1 stands for NON-CODING-1; CX stands for CLASSIC-XOR; DX1 stands for DEGENERATE-XOR-1; PM stands for PREMIXING; RC stands for REACTIVE-CODING.

Edge	Trans. Prob.	Edge	Trans. Prob.
$Q^1_\emptyset \to$NC1	$p_{d_1\vee d_2}$	$Q^1_\emptyset \to$PM	$p_{d_1\vee d_2}$
NC1$\to Q^1_{\{2\}}$	$p_{\overline{d_1}d_2}$	PM$\to Q_{\text{mix}}$	$p_{d_1\vee d_2}$
$Q^1_{\{2\}} \to$DX1	p_{d_1}	$Q_{\text{mix}} \to$RC	$p_{d_1\vee d_2}$
$Q^1_{\{2\}} \to$CX	p_{d_1}	RC$\to Q^1_{\{2\}}$	$p_{\overline{d_1}d_2}$

sending $[X_i + Y_j]$ but can also send $[P_1 + P_2]$ as long as P_1 is from $Q^1_{\{2\}}$ and P_2 is from $Q^2_{\{1\}}$. The same relaxation must be applied to DEGENERATE-XOR-1 and DEGENERATE-XOR-2 operations. Other than this slight relaxation, the three operations work in the same way as previously described in Sections 5.1 and 5.3.1.

As will be seen in Proposition 6.3.1 of Section 6.3, the two new operations PREMIXING and REACTIVE-CODING allow us to achieve the linear block-code capacity for any time-varying channels. We conclude this section by listing in Table 6.2 the transition probabilities of half of the edges of the vr-network of Fig. 6.1. For example, when we schedule PREMIXING, we remove a packet from $Q^1_\emptyset$ if at least one of $\{d_1, d_2\}$ receives it. As a result, the transition probability along the $Q^1_\emptyset \to$PREMIXING edge is $p_{d_1\vee d_2} \triangleq p_{d_1\overline{d_2}} + p_{\overline{d_1}d_2} + p_{d_1d_2}$. All the other transition probabilities in Table 6.2 can be derived similarly. The transition probability of the other half of the edges can be derived by symmetry.

6.1.2 Decoding and Buffer Management at Receivers

It is worth emphasizing that the vr-network is a conceptual tool used by the source s to decide what to transmit in each time slot. As a result, for the encoding purposes s only needs to store in its memory/buffer all the packets that currently participate in the vr-network. This automatically implies that as long as the queues in the vr-network are stabilized, the actual memory usage at the source is also stabilized. However, for the 1-to-2 access point network to be stable, one needs to ensure that the memory usage for the two receivers is stabilized as well. In this subsection we discuss the decoding operations and the memory usage at the receivers.

It is clear that each receiver needs to store some packets for the decoding purposes. A very commonly used assumption in the Shannon-capacity literature is to assume that the receivers store all the overheard packets in order to decode the possible XORed packets sent from the source. No packets will ever be removed from the buffer under such a policy. Obviously, such an infinite-buffer scheme is highly impractical.

In the existing INC scheduling works [5, 30, 38, 39], another commonly used buffer management scheme is the following. For any time t, define i^* (resp. j^*) as the smallest i (resp. j) such that d_1 (resp. d_2) has not decoded X_i (resp. Y_j) in the end of time t. Then each receiver can simply remove any X_i and Y_j in the buffer for those $i < i^*$ and $j < j^*$. The reason is that those X_i and Y_j has already been known by their intended receivers, will not participate in any future transmission, and thus can be removed from the receive buffer without any impact to future decoding.

On the other hand, under such a buffer management scheme, the receivers may use significantly more memory than that of the source, which was observed in our numerical experiments. The reason is as follows. Suppose d_1 has decoded X_1, X_3, X_4,..., X_8, and X_{10} and suppose d_2 has decoded Y_1 to Y_4 and Y_6 to Y_{10}. In this case $i^* = 2$ and $j^* = 5$. The aforementioned scheme will keep all X_2 to X_{10} in the buffer of d_2 and all Y_5 to Y_{10} in the buffer of d_1. But it turns out that the source is interested in only sending 3 more packets X_2, X_9, and Y_5. This apparent waste of memory is due to the fact that having 3 more packets to send does not mean that we only need to store X_2, X_9 and Y_5 in the buffer of the *receivers*. For the decoding purposes, we need to store extra “overheard” packets that can facilitate decoding in the future. But on the other hand, the above buffer management scheme is too conservative and very inefficient since it does not trace the actual overhearing status of each packet and only use the simplest i^* and j^* pair to decide whether to prune the packets in the buffers of the receivers.

In contrast with the above buffer management scheme used in [5, 30, 38, 39], our vr-network scheme admits the following efficient decoding operations and buffer management solution. In the following, we describe the decoding and buffer management at d_1. The operations at d_2 can be done symmetrically. Our description consists of two parts. We first describe how to perform decoding at d_1 and which packets need to be stored in d_1's buffer, while assuming that any packets that have been stored in the buffer will never be expunged. In the second part, we describe how to prune the memory usage without affecting the decoding operations.

Upon d_1 receiving a packet: Case 1: If the received packet is generated by NON-CODING-1, then such a packet must be X_i for some i. We thus pass such an X_i to the upper layer; Case 2: If the received packet is generated by NON-CODING-2, then such a packet must be Y_j for some j. We store Y_j in the buffer of d_1; Case 3: If the received packet is generated by PREMIXING, then such a packet must be $[X_i + Y_j]$. We store the linear sum $[X_i + Y_j]$ in the buffer. Case 4: If the received packet is generated by REACTIVE CODING , then such a packet can be either X_i^* or Y_j^*, see Table 6.1 for detailed descriptions of REACTIVE-CODING.

We have two sub-cases in this scenario. Case 4.1: If the packet is X_i^*, we pass such an X_i^* to the upper layer. Then d_1 examines whether it has stored $[X_i^* + Y_j^*]$ in its buffer. If so, use X_i^* to decode Y_j^* and insert Y_j^* to the buffer. If not, store a separate copy of X_i^* in the buffer even though one copy of X_i^* has already been passed to the upper layer. Case 4.2: If the packet is Y_j^*, then by Table 6.1, it is clear that d_1 must have received the linear sum $[X_i^* + Y_j^*]$ in the corresponding PREMIXING operation in the past. Therefore, $[X_i^* + Y_j^*]$ must be in the buffer of d_1 already. We can thus use Y_j^* and $[X_i^* + Y_j^*]$ to decode the desired X_i^*. Receiver d_1 then passes the decoded X_i^* to the upper layer and stores Y_j^* in its buffer.

Case 5: If the received packet is generated by DEGENERATE XOR-1, then such a packet can be either X_i or Y_j, where Y_j are those packets in $Q^1_{\{2\}}$ but coming from REACTIVE CODING, see Fig. 6.1. Case 5.1: If the packet is X_i, we pass such an X_i to the upper layer. Case 5.2: If the packet is Y_j, then from Table 6.1, it must be corresponding to the intersection of the row of $\mathsf{rcpt} = \overline{d_1}d_2$ and the column of $\mathsf{rcpt}^* = d_1\overline{d_2}$. As a result, d_1 must have received the corresponding $[X_i + Y_j]$ in the PREMIXING operation. By Case 3, the linear sum has been stored in the buffer, and d_1 can thus use the received Y_j to decode the desired X_i. After decoding, X_i is passed to the upper layer.

Case 6: the received packet is generated by DEGENERATE XOR-2. Consider two subcases. Case 6.1: the received packet is X_i. It is clear from Fig. 6.1 that such X_i must come from REACTIVE-CODING since any packet from $Q^2_{\emptyset}$ to $Q^2_{\{1\}}$ must be a Y_j packet. By Table 6.1 and the row corresponding to $\mathsf{rcpt} = d_1\overline{d_2}$, any $X_i \in Q^2_{\{1\}}$ that came from REACTIVE-CODING must correspond to the column of $\mathsf{rcpt}^* = \overline{d_1}d_2$. By the second half of Case 4.1, such $X_i \in Q^2_{\{1\}}$ must be in the buffer of d_1. As a result, d_1 can simply ignore any X_i packet it receives from DEGENERATE XOR-2. Case 6.2: the received packet is Y_j. By the discussion of Case 2, if the $Y_j \in Q^2_{\{1\}}$ came from NON-CODING-2, then it must be in the buffer of d_1 already. As a result, d_1 can simply ignore those Y_j packets. If the $Y_j \in Q^2_{\{1\}}$ came from REACTIVE-CODING, then by Table 6.1 and the row corresponding to $\mathsf{rcpt} = d_1\overline{d_2}$, those $Y_j \in Q^2_{\{1\}}$ must correspond to the column of either $\mathsf{rcpt}^* = d_1\overline{d_2}$ or $\mathsf{rcpt}^* = d_1d_2$. By the first half of Case 4.1 and by Case 4.2, such $Y_j \in Q^2_{\{1\}}$ must be in the buffer of d_1 already. Again, d_1 can simply ignore those Y_j packets. From the discussion of Cases 6.1 and 6.2, *any packet generated by* DEGENERATE XOR-2 *is already known to* d_1*, and nothing needs to be done in this case.*[2]

Case 7: the received packet is generated by CLASSIC-XOR. Since we have shown in Case 6 that any packet in $Q^2_{\{1\}}$ is already known to d_1, receiver d_1 can simply subtract the $Q^2_{\{1\}}$ packet from the linear sum received in Case 7. As a result, from d_1's perspective, it is no different than directly receiving a $Q^1_{\{2\}}$ packet, i.e., Case 5. As a result, d_1 will repeat the decoding operation and buffer management in the same way as in Case 5.

Periodically pruning the memory: In the above discussion, we elaborate which packets d_1 should store in its buffer and how to use them for decoding, while assuming no packet will ever be removed from the buffer. In the following, we discuss how to remove packets from the buffer of d_1.

[2]The discussion of Cases 5 and 6 echoes our arguments in the end of [Section 6.1.1: Encoding] that any packet in $Q^2_{\{1\}}$ (which can be either X_i or Y_j) is information-equivalent to a session-2 packet that has been overheard by d_1.

We first notice that by the discussion of Cases 1 to 7, the uncoded packets in the buffer of d_1, i.e., those of the form of either X_i or Y_j, are used for decoding *only in the scenario of Case 7*. Namely, they are used to remove the $Q^2_{\{1\}}$ packet participating in the linear sum of CLASSIC-XOR. As a result, periodically we let the source s send to d_1 the list[3] of all packets in $Q^2_{\{1\}}$ of the vr-network. After receiving the list, d_1 simply removes from its buffer any uncoded packets X_i and/or Y_j that are no longer in $Q^2_{\{1\}}$.

We then notice that by the discussion of Cases 1 to 7, the linear sum $[X_i + Y_j]$ in the buffer of d_1 is only used in one of the following two scenarios: (i) To decode Y_j in Case 4.1 or to decode X_i in Case 4.2; and (ii) To decode X_i in Case 5.2. As a result, the $[X_i + Y_j]$ in the buffer is "useful" only if one of the following two conditions are satisfied: (a) The corresponding tuple $(\mathsf{rcpt}, X_i, Y_j)$ is still in the Q_{mix} of the vr-network, which corresponds to the scenarios of Cases 4.1 and 4.2; and (b) If the participating Y_j is still in the $Q^1_{\{2\}}$ of the vr-network. By the above observation, periodically we let the source s send to d_1 the list of all packets in $Q^1_{\{2\}}$ and Q_{mix} of the vr-network.[4] After receiving the list, d_1 simply removes from its buffer any linear sum $[X_i + Y_j]$ that satisfies neither (a) nor (b).

The above pruning mechanism ensures that only the packets useful for future decoding are kept in the buffer of d_1 and d_2. Furthermore, it also leads to the following lemma.

Lemma 6.1.1 *Assume the lists of packets in $Q^1_{\{2\}}$, $Q^2_{\{1\}}$, and Q_{mix} are sent to d_1 after every time slot. The number of packets in the buffer of d_1 is upper bounded by $|Q^1_{\{2\}}| + |Q^2_{\{1\}}| + |Q_{mix}|$.*

Proof: From our discussion, the total number of uncoded packets X_i or Y_j in the buffer of d_1 is upper bounded by $|Q^2_{\{1\}}|$. Also, the total number of linear sum $[X_i + Y_j]$ in the buffer of d_1 is upper bounded by $|Q_{\text{mix}}|$ plus the number of Y_j packets in $Q^1_{\{2\}}$, which is further bounded by $|Q_{\text{mix}}| + |Q^1_{\{2\}}|$. As a result, the total number of packets in the buffer of d_1 is upper bounded by $|Q^1_{\{2\}}| + |Q^2_{\{1\}}| + |Q_{\text{mix}}|$. ∎

[3]Only the packet IDs are sent, not the payload. Therefore the overhead of sending the list is small. Moreover, we only need to send the "incremental changes" of the list and d_1 can update the list by itself. In this way, the overhead of sending the list can be made negligible.

[4]One can see that both d_1 and d_2 need to receive the lists of packets in $Q^1_{\{2\}}$, $Q^2_{\{1\}}$, and Q_{mix}. Therefore, s can simply *broadcast* (the changes) of the three lists to both d_1 and d_2.

Lemma 6.1.1 implies that as long as the queues in the vr-network are stabilized, the actual memory usage at both the source and the destinations can be stabilized simultaneously. Moreover, the combined memory usage of the source and 2 receivers will be upper bounded by $Q^1_\emptyset + Q^2_\emptyset + 3|Q^1_{\{2\}}| + 3|Q^2_{\{1\}}| + 3|Q_{\text{mix}}|$ in the vr-network.

Remark: In addition to efficient decoding and buffer management, we notice that in the proposed INC scheme, only the binary XOR is used and each transmitted packet is either an uncoded packet or a linear sum of two packets. Therefore, during transmission we only need to store 1 or 2 packet sequence numbers in the header of the uncoded/coded packet, depending on whether we send an uncoded packet or a linear sum. As a result, the communication overhead of the proposed scheme is very small.

6.2 The Proposed Scheduling Solution

In this section, we first formalize the model of SPNs with random departure and then we propose a new scheme that achieves the optimal throughput region for SPNs with random departure. We conclude this section by providing the corresponding stability/throughput analysis.

6.2.1 A Simple SPN model with Random Departure

Although our solution applies to general SPNs with random departure, for illustration purposes we describe our scheme by focusing on a simple SPN model with random departure, which we termed the (0,1) random SPN. The (0,1) random SPN includes the INC vr-network in Section 6.1 as a special example and is thus sufficient for our discussion.

Recall the definitions in Section 5.3.2 for SPNs with deterministic departure (we use deterministic SPNs as shorthand). The differences between the (0,1) random SPN and the deterministic SPN are:

Difference 1: In a deterministic SPN, SA n can be activated only if for all k in the input queues $\mathcal{I}_n$, queue k has at least $\beta^{\text{in}}_{k,n}$ number of packets in the queue. For comparison, in a (0,1) random SPN, SA n can be activated only if for all $k \in \mathcal{I}_n$, queue k has at least 1 packet in the queue. For easier future reference, we say SA n is *feasible* at time t if at time t queue k has at least 1 packet for all $k \in \mathcal{I}_n$. Otherwise, we say SA n is infeasible at time t.

Difference 2: In a deterministic SPN, when SA n is activated with the channel quality c, exactly $\beta^{\text{in}}_{k,n}(c)$ number of packets will leave queue k for all $k \in \mathcal{I}_n$. In a (0,1) random SPN, when SA n is activated with the channel quality c (assuming SA n is feasible), the number of packets leaving queue k is a binary random variable, $\beta^{\text{in}}_{k,n}(c)$, with mean $\overline{\beta^{\text{in}}_{k,n}(c)}$ for all $k \in \mathcal{I}_n$. Namely, with probability $\overline{\beta^{\text{in}}_{k,n}(c)}$, 1 packet will leave queue k and with probability $1 - \overline{\beta^{\text{in}}_{k,n}(c)}$ no packet will leave queue k. Since the packet consumption is Bernoulli, in a (0,1) random SPN, it is possible that an SA consumes zero packet even after being activated. However, since we do not know how many packets will be consumed beforehand, the (0,1) random SPN imposes that all the input queues have at least 1 packet before we can activate an SA, even though when we actually activate the SA, it sometimes consumes zero packet. For comparison, in a deterministic SPN, an SA n is feasible if all its input queues have at least $\beta^{\text{in}}_{k,n}(\mathsf{cq}(t))$ packets and it will always consume exactly $\beta^{\text{in}}_{k,n}(\mathsf{cq}(t))$ packets from its input queues once activated (see Difference 1).

Difference 3: In a (0,1) random SPN, when SA n is activated with the channel quality c (assuming SA n is feasible), the number of packets entering queue k is a binary random variable with mean $\overline{\beta^{\text{out}}_{k,n}(c)}$ for all $k \in \mathcal{O}_n$.

We also use the following 3 technical assumptions for the (0,1) random SPN: Assumption 1: Given any channel quality $c \in \mathsf{CQ}$, both the input and output service matrix $\mathcal{B}^{\text{in}}$ and $\mathcal{B}^{\text{out}}$ are independently distributed over time. Assumption 2: Each vector in the set of possible service vectors $\mathfrak{X}$ can have at most 1 non-zero coordinate. Namely, we can activate at most one service activity (out of totally N SAs) at any given time. Assumption 3: For any $\mathsf{cq}(t)$, the expectation of $\beta^{\text{in}}_{k,n}(\mathsf{cq}(t))$ (resp. $\beta^{\text{out}}_{k,n}(\mathsf{cq}(t))$) with $k \in \mathcal{I}_n$ (resp. $k \in \mathcal{O}_n$) is always strictly in $(0, 1]$. Namely, we do not consider the limiting case in which the Bernoulli random variables are always 0. Assumption 1 is related to the practical scenarios. Assumptions 2 and 3 are for rigorously proving the stability region.

One can easily verify that the three INC vr-networks in Figs. 5.1(a), 5.1(b), and 6.1 are special examples of the (0,1) random SPN and they satisfy the 3 technical assumptions as well.

6.2.2 The Proposed Scheduler For (0,1) Random SPNs

Similar to the DMW algorithm, each queue k maintains a real-valued counter $q_k(t)$, the virtual queue length. Initially, $q_k(1)$ is set to 0. For any time t, the realization of each entry in the input and output service matrices $\mathcal{B}^{\text{in}}$ and $\mathcal{B}^{\text{out}}$ takes values in either 0 or 1 since we are focusing on a (0,1) random SPN. We compute $\overline{\mathcal{B}^{\text{in}}(\mathsf{cq}(t))} \triangleq \mathsf{E}(\mathcal{B}^{\text{in}}|\mathsf{cq}(t))$ and $\overline{\mathcal{B}^{\text{out}}(\mathsf{cq}(t))} \triangleq \mathsf{E}(\mathcal{B}^{\text{out}}|\mathsf{cq}(t))$, the *expected* input and output service matrices, respectively, when the channel quality is $\mathsf{cq}(t)$. The entries of $\overline{\mathcal{B}^{\text{in}}(\mathsf{cq}(t))}$ and $\overline{\mathcal{B}^{\text{out}}(\mathsf{cq}(t))}$ are denoted by $\overline{\beta^{\text{in}}_{k,n}(\mathsf{cq}(t))}$ and $\overline{\beta^{\text{out}}_{k,n}(\mathsf{cq}(t))}$, respectively. Obviously, by definition, the expected input and output service rates are non-negative numbers. For each time t, we choose the preferred service vector by the back-pressure decision rule (5.2) except for that the back-pressure vector $\mathbf{d}(t)$ is now computed by

$$\mathbf{d}(t) = \left(\overline{\mathcal{B}^{\text{in}}(\mathsf{cq}(t))} - \overline{\mathcal{B}^{\text{out}}(\mathsf{cq}(t))}\right)^{\mathsf{T}} \mathbf{q}(t). \tag{6.1}$$

We use the new back-pressure vector $\mathbf{d}(t)$ plus (5.2) to find the preferred SA n^*, i.e., all the coordinates of $\mathbf{x}^*$ are zero except for the n^*-th coordinate being one. We then check whether the preferred SA n^* is feasible. If so, we officially schedule SA n^*. If not, we let the system to be idle,[5] i.e., the actually scheduled service vector $\mathbf{x}(t) = \mathbf{0}$ is now all-zero.

Regardless of whether the preferred SA n^* is feasible or not, we update $\mathbf{q}(t)$ by

$$\begin{aligned}\mathbf{q}(t+1) =& \mathbf{q}(t) + \mathcal{A} \cdot \mathbf{a}(t) \\ &+ \left(\overline{\mathcal{B}^{\text{out}}(\mathsf{cq}(t))} - \overline{\mathcal{B}^{\text{in}}(\mathsf{cq}(t))}\right) \cdot \mathbf{x}^*(t).\end{aligned} \tag{6.2}$$

[5]The reason of letting the system idle is to facilitate rigorous stability analysis. In practice, we can choose arbitrarily any other feasible SA at that moment.

Note that $\mathbf{q}(t)$ can sometimes take negative values since we do not project $\mathbf{q}(t)$ to positive reals.

In short, we borrow the wisdom of DMW so that we can make scheduling decisions based on the virtual queue lengths $q_k(t)$ that can take negative values. But then we update $q_k(t)$ only by the expected service rates rather than the actual service rates since we are dealing with a random SPN instead of a deterministic SPN. For notation simplicity, we denote the proposed scheduler for (0,1) random SPNs by $\mathsf{SCH}_{\text{avg}}$.

6.2.3 Performance Analysis

The example in Section 5.3.2 shows that one challenge of the SPN with random departure is that $Q_k(t)$ may grow unboundedly (sublinearly) even when the expected flow-conservation law in (5.1) is satisfied. In this work, we prove that the sublinearly growing queues in the example of Section 5.3.2 are actually the worst possible case that could happen. Namely, for SPNs with random departure, we can always find an algorithm such that all queue lengths grow sublinearly when the input rates are within the optimal stability region.

Note that from a throughput perspective, sublinear growth means that the throughput penalty incurred by the growing queues is negligible since the throughput is the average number of the packet arrivals per second and only the linear terms matter in the long run. Moreover, for any scheme A that achieves sublinearly growing queues, it is likely (without any rigorous proof) that we can convert it to a bounded queue scheme by (i) Run scheme A until any of the sublinearly growing queue length hits some pre-defined threshold; (ii) Stop scheme A and run a naive scheme B that focuses on "draining" the queues of the network; (iii) When running scheme B, put any new arrival packets into a separate buffer Q; (iv) After scheme B successfully drains out all the queues, we start to run scheme A again and we inject the packets collected in Q gradually back to the system. The above 4 steps guarantee that the queue lengths are bounded. Heuristically, they also approach the optimal throughput since the queues grow sublinearly, the penalty of running the "drainingstage scheme B" should also be negligible when choosing a sufficiently large threshold in Step (i).

From the above reasonings, we believe that sublinearly growing queues are as good as the bounded queues from a practical perspective. The following analysis is based on the concept of sublinearly growing queue lengths.

Definition 6.2.1 *A queue length $q(t)$* grows sublinearly *if for any $\epsilon > 0$ and $\delta > 0$, there exists t_0 such that*

$$\mathsf{Prob}(|q(t)| > \epsilon t) < \delta,\ \forall t > t_0. \tag{6.3}$$

Since we assume that the input activities $\mathbf{a}(t)$ have bounded support, an equivalent definition of sublinear growth is: $q(t)$ grows sublinearly if for any $\rho > 0$ there exists t_0 such that

$$\mathsf{E}\{|q(t)|\} < \rho t,\ \forall t > t_0. \tag{6.4}$$

An SPN is sublinearly stable *if all the queues grow sublinearly.*

Remark: As a result of the above definition, one can observe that the summation of finitely many sublinearly-growing queues is still sublinearly-growing.

The following two propositions characterize the sublinear stability region of any (0,1) random SPN. Proposition 6.2.1 specifies the outer bound of the stability region, and Proposition 6.2.2 specifies an inner bound.

Proposition 6.2.1 *Consider any (0,1) random SPN. A rate vector $\mathbf{R}$ can be sublinearly stabilized only if there exist $\mathbf{s}_c \in \Lambda$ for all $c \in \mathsf{CQ}$ such that*

$$\mathcal{A} \cdot \mathbf{R} + \sum_{c \in \mathsf{CQ}} f_c \cdot \overline{\mathcal{B}^{out}(c)} \cdot \mathbf{s}_c = \sum_{c \in \mathsf{CQ}} f_c \cdot \overline{\mathcal{B}^{in}(c)} \cdot \mathbf{s}_c. \tag{6.5}$$

Proposition 6.2.1 can be derived by conventional flow conservation arguments as in [31] and the proof is thus omitted.

Proposition 6.2.2 *For any SPN that satisfies the three assumptions in Section 6.2.1 and any rate vector* $\mathbf{R}$*, if there exist* $\mathbf{s}_c \in \Lambda^\circ$ *for all* $c \in \mathsf{CQ}$ *such that (6.5) holds, then the proposed scheme* SCH_{avg} *in Section 6.2.2 can sublinearly stabilize the SPN with arrival rate* $\mathbf{R}$.

Outline of the proof of Proposition 6.2.2: Let each queue k keep another two real-valued counters $q_k^{\text{inter}}(t)$ and $Q_k^{\text{inter}}(t)$, termed the *intermediate virtual queue length* and *intermediate actual queue length*. Recall that $q_k(t)$ is the virtual queue length and $Q_k(t)$ is the actual queue length. There are thus 4 different queue length values[6] for each queue k. To prove $\mathbf{Q}(t)$ can be sublinearly stabilized by $\mathsf{SCH}_{\text{avg}}$, we will show that both $Q_k^{\text{inter}}(t)$ and the absolute difference $|Q_k(t) - Q_k^{\text{inter}}(t)|$ can be sublinearly stabilized by $\mathsf{SCH}_{\text{avg}}$ for all k. Since the summation of sublinearly-growing random processes is still sublinearly-growing, $\mathbf{Q}(t)$ can be sublinearly stabilized by $\mathsf{SCH}_{\text{avg}}$, and we have thus proven Proposition 6.2.2.

To that end, we first specify the update rules for $q_k^{\text{inter}}(t)$ and $Q_k^{\text{inter}}(t)$. Initially, $q_k^{\text{inter}}(1)$ and $Q_k^{\text{inter}}(1)$ are set to 0 for all k. In the end of each time t, we compute $\mathbf{q}^{\text{inter}}(t+1)$ using the preferred schedule $\mathbf{x}^*(t)$ chosen by $\mathsf{SCH}_{\text{avg}}$:

$$\begin{aligned}\mathbf{q}^{\text{inter}}(t+1) =& \mathbf{q}^{\text{inter}}(t) + \mathcal{A} \cdot \mathbf{a}(t) \\ &+ \left(\mathcal{B}^{\text{out}}(\mathsf{cq}(t)) - \mathcal{B}^{\text{in}}(\mathsf{cq}(t))\right) \cdot \mathbf{x}^*(t).\end{aligned} \tag{6.6}$$

If we compare (6.6) with the computation of $\mathbf{q}(t)$ in (6.2), $\mathbf{q}^{\text{inter}}(t)$ is updated based on the *realization* of the input and output service matrices while $\mathbf{q}(t)$ is updated based on the *expected* input and output service matrices. Equivalently, we can rewrite (6.6) as

$$q_k^{\text{inter}}(t+1) = q_k^{\text{inter}}(t) - \mu_{\text{out},k}(t) + \mu_{\text{in},k}(t), \ \forall k, \tag{6.7}$$

[6] $q_k^{\text{inter}}(t)$ and $Q_k^{\text{inter}}(t)$ are used only for the proof and are not needed when running the scheduling algorithm.

where

$$\mu_{\text{out},k}(t) = \sum_{n=1}^{N} \left(\beta_{k,n}^{\text{in}}(\mathsf{cq}(t)) \cdot x_n^*(t)\right), \tag{6.8}$$

$$\mu_{\text{in},k}(t) = \sum_{m=1}^{M} \left(\alpha_{k,m} \cdot a_m(t)\right) + \sum_{n=1}^{N} \left(\beta_{k,n}^{\text{out}}(\mathsf{cq}(t)) \cdot x_n^*(t)\right). \tag{6.9}$$

Here, $\mu_{\text{out},k}$ is the amount of packets coming "out of queue k", which is decided by the "input rates of SA n". Similarly, $\mu_{\text{in},k}$ is the amount of packets "entering queue k", which is decided by the "output rates of SA n". We also update $\mathbf{Q}^{\text{inter}}(t+1)$ by

$$Q_k^{\text{inter}}(t+1) = \left(Q_k^{\text{inter}}(t) - \mu_{\text{out},k}(t)\right)^+ + \mu_{\text{in},k}(t),\ \forall k, \tag{6.10}$$

where $(v)^+ = \max\{0, v\}$.

The difference between $q_k^{\text{inter}}(t)$ and $Q_k^{\text{inter}}(t)$ is that the former can be still be strictly negative when updated via (6.7) while we enforce the latter to be non-negative.

To compare $Q_k^{\text{inter}}(t)$ and $Q_k(t)$, we observe that by (6.10), $Q_k^{\text{inter}}(t)$ is purely updated by the preferred service vector $\mathbf{x}^*(t)$ without considering whether the preferred SA n^* is feasible or not (see Difference 1 in Section 6.2.1). That is, in the case that SA n^* is infeasible, then SA n^* cannot be carried out successfully. Therefore, the system remains idle and the actual queue length $Q_k(t+1) = Q_k(t)$ for all $k = 1$ to K or $Q_k(t)$ increases if there is external arrival at queue k. In contrast, even though SA n^* cannot be carried out successfully, we still update $Q_k^{\text{inter}}(t+1)$ by (6.8) to (6.10) for all queue k. As a result, the $Q_k^{\text{inter}}(t)$ values will still change[7] for those $k \in \mathcal{I}_n \cup \mathcal{O}_n$.

To evaluate the absolute difference $|Q_k(t) - Q_k^{\text{inter}}(t)|$, for any time t and any queue k, we first define an event, which is called the *null activity* of queue k at time t. Since we assume at any time t, only one SA can be scheduled, we use $n(t)$ to denote the preferred SA suggested by the back-pressure scheduler in (5.2) and (6.1). As a result, at time t, we

[7] In the original DMW algorithm for deterministic SPNs [31], the quantity "actual queue length" is updated by (6.10). The "actual queue lengths in [31]" thus refer to a conceptual register value $Q_k^{\text{inter}}(t)$ rather than the number of physical packets in the buffer/queue. In this work, we rectify this inconsistency by renaming "the actual queue lengths in [31]" the "intermediate actual queue lengths $Q_k^{\text{inter}}(t)$."

say the null activity occurs at queue k if (i) $k \in \mathcal{I}_{n(t)}$ and (ii) $Q_k^{\text{inter}}(t) < \beta_{k,n}^{\text{in}}(\mathsf{cq}(t))$. That is, the null activity describes the event that the preferred SA shall consume the packets in queue k (since $k \in \mathcal{I}_{n(t)}$) but the intermediate actual queue length $Q_k^{\text{inter}}(t)$ is less than the realization $\beta_{k,n}^{\text{in}}(\mathsf{cq}(t))$. Note that the null activity is defined based on the intermediate actual queue length $Q_k^{\text{inter}}(t)$ and does not distinguish whether the actual queue length $Q_k(t)$ is larger or less than 1. Therefore the null activities are not directly related to the event that SA n is infeasible.

Let $N_{\mathsf{NA},k}(t)$ be the aggregate number of null activities occurred at queue k up to time t. Then we can write $N_{\mathsf{NA},k}(t)$ as

$$N_{\mathsf{NA},k}(t) \triangleq \sum_{\tau=1}^{t} I(k \in \mathcal{I}_{n(\tau)}) \cdot I(Q_k^{\text{inter}}(\tau) < \beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))),$$

where $I(\cdot)$ is the indicator function.

The following lemma upper bounds the difference of $Q_k(t)$ and $Q_k^{\text{inter}}(t)$ by the aggregate numbers of null activities.

Lemma 6.2.1 *For all $k = 1, 2, ..., K$, there exist K non-negative coefficients $\gamma_1, ..., \gamma_K$ such that*

$$\mathsf{E}(|Q_k(t) - Q_k^{inter}(t)|) \leq \sum_{\tilde{k}=1}^{K} \gamma_{\tilde{k}} N_{\mathsf{NA},\tilde{k}}(t). \tag{6.11}$$

for all $t = 1$ to ∞.

The proof of Lemma 6.2.1 is relegated to Appendix D. In Appendix G, we prove that both $Q_k^{\text{inter}}(t)$ and $N_{\mathsf{NA},k}(t)$ of (0,1) random SPN can be sublinearly stabilized by $\mathsf{SCH}_{\text{avg}}$

for all k.[8] Therefore, by Lemma 6.2.1, $Q_k^{\text{inter}}(t)$ and $|Q_k(t) - Q_k^{\text{inter}}(t)|$ can be sublinearly stabilized and so can $Q_k(t)$. Proposition 6.2.2 is thus proven.

6.3 The Combined Solution

We are now ready to combine the discussions in Sections 6.1 and 6.2. As discussed in Section 6.1, the 7 operations form a vr-network as described in Fig. 6.1 and both the source and the two receivers perform encoding and decoding according to the packet movements in the vr-network, respectively. Specifically, there are $K = 5$ queues, $M = 2$ IAs, and $N = 7$ SAs. The 5-by-2 input matrix $\mathcal{A}$ contains 2 ones, since the packets arrive at either $Q_\emptyset^1$ or $Q_\emptyset^2$. Given the channel quality $\mathsf{cq}(t) = c$, the expected input and output service matrices $\overline{\mathcal{B}^{\text{in}}(c)}$ and $\overline{\mathcal{B}^{\text{out}}(c)}$ can be derived from Table 6.2.

We use the following concrete example to illustrate our procedure. Suppose that the channel quality $\mathsf{cq}(t)$ is Bernoulli with parameter $1/2$ (i.e., flipping a perfect coin). Also suppose that when $\mathsf{cq}(t) = 0$, with probability 0.5 (resp. 0.7) destination d_1 (resp. destination d_2) can successfully receive a packet transmitted by source s; and when $\mathsf{cq}(t) = 1$, with probability $2/3$ (resp. $1/3$) destination d_1 (resp. destination d_2) can successfully receive a packet transmitted by source s. Further assume that all the success events of d_1 and d_2 are independent. Please also see Appendix H for further details on the matrix con-

[8] The DMW algorithm for SPNs were first introduced in [31]. However, in that paper, the authors rename the intermediate actual queue lengths defined in (6.10) of this paper as the actual queue length and prove that $Q_k^{\text{inter}}(t)$ can be stabilized for deterministic SPNs. However, proving $Q_k^{\text{inter}}(t)$ can be stabilized does not necessarily mean that $Q_k(t)$ can be stabilized, as discussed in the paragraphs after (6.10). The critical part of proving $|Q_k(t) - Q_k^{\text{inter}}(t)|$ is stabilized is unfortunately missing in [31]. One contribution of this work is to provide in Lemma 6.2.1 the first rigorous proof showing that $|Q_k(t) - Q_k^{\text{inter}}(t)|$ can be stabilized as well.

struction. If we order the 5 queues as $\left[\mathbf{Q}^1_{\emptyset}, \mathbf{Q}^2_{\emptyset}, \mathbf{Q}^1_{\{2\}}, \mathbf{Q}^2_{\{1\}}, \mathbf{Q}_{\text{mix}}\right]$, the 7 service activities as $[\text{NC1}, \text{NC2}, \text{DX1}, \text{DX2}, \text{PM}, \text{RC}, \text{CX}]$, then the matrices of the SPN become

$$\mathcal{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}},$$

$$\overline{\mathcal{B}^{\text{in}}(0)} = \begin{bmatrix} 0.85 & 0 & 0 & 0 & 0.85 & 0 & 0 \\ 0 & 0.85 & 0 & 0 & 0.85 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0.7 & 0 & 0 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0.85 & 0 \end{bmatrix},$$

$$\overline{\mathcal{B}^{\text{in}}(1)} = \begin{bmatrix} 7/9 & 0 & 0 & 0 & 7/9 & 0 & 0 \\ 0 & 7/9 & 0 & 0 & 7/9 & 0 & 0 \\ 0 & 0 & 2/3 & 0 & 0 & 0 & 2/3 \\ 0 & 0 & 0 & 1/3 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 7/9 & 0 \end{bmatrix},$$

$$\overline{\mathcal{B}^{\text{out}}(0)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.35 & 0 & 0 & 0 & 0 & 0.35 & 0 \\ 0 & 0.15 & 0 & 0 & 0 & 0.15 & 0 \\ 0 & 0 & 0 & 0 & 0.85 & 0 & 0 \end{bmatrix},$$

$$\overline{\mathcal{B}^{\text{out}}(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/9 & 0 & 0 & 0 & 0 & 1/9 & 0 \\ 0 & 4/9 & 0 & 0 & 0 & 4/9 & 0 \\ 0 & 0 & 0 & 0 & 7/9 & 0 & 0 \end{bmatrix},$$

$$\mathbf{s}_c = \left[x^{[c]}_{\text{NC1}}\ x^{[c]}_{\text{NC2}}\ x^{[c]}_{\text{DX1}}\ x^{[c]}_{\text{DX2}}\ x^{[c]}_{\text{PM}}\ x^{[c]}_{\text{RC}}\ x^{[c]}_{\text{CX}}\right]^{\mathsf{T}}.$$

For example, the seventh column of $\overline{\mathcal{B}^{\text{in}}(0)}$ indicates that when $\mathsf{cq}(t) = 0$ and the CLASSIC-XOR is activated, with probability 0.5 (resp. 0.7) 1 packet will be consumed from queue $\mathbf{Q}^1_{\{2\}}$ (resp. $\mathbf{Q}^2_{\{1\}}$). The third row of $\overline{\mathcal{B}^{\text{out}}(1)}$ indicates that when $\mathsf{cq}(t) = 1$, queue $\mathbf{Q}^1_{\{2\}}$ will increase by 1 with probability 1/9 (resp. 1/9) if coding choice NON-CODING-1 (resp. REACTIVE-CODING) is activated since it corresponds to the event that d_1 receives the transmitted packet but d_2 does not.

Since there are 7 coding operations (SAs), each vector in $\mathfrak{X}$ is a 7-dimensional binary vector. Since we are allowed to choose any one of the 7 operations or choose to transmit nothing, 7 of the 8 vectors are the Dirac delta vectors and the rest is an all-zero vector. We can now use the proposed DMW scheduler in (5.2), (6.1), and (6.2) to compute the preferred scheduling decision. We activate the preferred decision if it is feasible. If not, then the system remains idle.

For general channel parameters (including but not limited to this simple example), after computing the $\overline{\mathcal{B}^{\text{in}}(c)}$ and $\overline{\mathcal{B}^{\text{out}}(c)}$ of the vr-network in Fig. 6.1 with the help of Table 6.2, we can explicitly compare the sublinear stability region in Propositions 6.2.1 and 6.2.2 with the Shannon capacity region in [43]. In the end, we have the following proposition.

Proposition 6.3.1 *The sublinear stability region of the proposed INC-plus-SPN-scheduling scheme matches the block-code capacity of time-varying channels.*

The detailed proof of Proposition 6.3.1 is provided in Appendix H.

Remark: During numerical simulations, we notice that we can further revise the proposed scheme to reduce the actual queue lengths $Q_k(t)$ by $\approx 50\%$ even though we do not have any rigorous proofs/performance guarantees for the revised scheme. That is, when making the scheduling decision by (5.2), we can compute $\mathbf{d}(t)$ by

$$\mathbf{d}(t) = \left(\overline{\mathcal{B}^{\text{in}}(\mathsf{cq}(t))} - \overline{\mathcal{B}^{\text{out}}(\mathsf{cq}(t))}\right)^{\mathsf{T}} \mathbf{q}^{\text{inter}}(t). \tag{6.12}$$

where $\mathbf{q}^{\text{inter}}(t)$ is the intermediate virtual queue length defined in (6.7) of Section 6.2.3. The intuition behind is that the new back-pressure in (6.12) allows the scheme to directly control $q_k^{\text{inter}}(t)$, which, when compared to the virtual queue $\mathbf{q}(t)$ in (6.2), is more closely related to the actual queue length[9] $Q_k(t)$.

6.3.1 Extensions For Rate Adaption

We close this section by noting that the proposed solution can be naturally extended to the case of rate adaptation, which is also known as adaptive coding and modulation. For illustration purposes, we consider the following simple example of adaptive coding and modulation scheme.

Consider 2 possible error correcting rates (1/2 and 3/4); 2 possible modulation schemes QPSK and 16QAM; and jointly there are 4 possible combinations. The lowest throughput combination is rate-1/2 plus QPSK and the highest throughput combination is rate-3/4 plus 16QAM. Assuming the packet size is fixed. If the highest throughput combination takes 1-unit time to finish sending 1 packet, then the lowest throughput combination will take 3-unit time. For these 4 possible (rate,modulation) combinations, we denote the unit-time to finish transmitting 1 packet as T_1 to T_4, respectively.

For the i-th (rate,modulation) combination, $i = 1$ to 4, source s can measure the probability that d_1 and/or d_2 successfully hears the transmission, and denote the corresponding probability vector by $\vec{p}^{(i)}$. Source s then uses $\vec{p}^{(i)}$ to compute the $\overline{\mathcal{B}^{\text{in},(i)}(\mathsf{cq}(t))}$ and $\overline{\mathcal{B}^{\text{out},(i)}(\mathsf{cq}(t))}$ for the vr network. Then it computes the backpressure by

$$\mathbf{d}^{(i)}(t) = \left(\overline{\mathcal{B}^{\text{in},(i)}(\mathsf{cq}(t))} - \overline{\mathcal{B}^{\text{out},(i)}(\mathsf{cq}(t))}\right)^{\mathsf{T}} \mathbf{q}(t).$$

We can now compute the preferred scheduling choice by

$$\arg\max_{i \in \{1,2,3,4\}, \mathbf{x} \in \mathfrak{X}} \frac{\mathbf{d}^{(i)}(t)^{\mathsf{T}} \cdot \mathbf{x}}{T_i} \tag{6.13}$$

[9]There are four types of queue lengths in this work: $\mathbf{q}(t)$, $\mathbf{q}^{\text{inter}}(t)$, $\mathbf{Q}^{\text{inter}}(t)$, and $\mathbf{Q}(t)$ and they range from the most artificially-derived $\mathbf{q}(t)$ to the most realistic metric, the actual queue length $\mathbf{Q}(t)$.

and update the virtual queue length $\mathbf{q}(t)$ by (6.2). Namely, the backpressure $\mathbf{d}^{(i)}(t)^{\mathsf{T}} \cdot \mathbf{x}$ is scaled inverse proportionally with respect to T_i, the time it takes to finish the transmission of 1 packet. If the preferred SA n^* is feasible, then we use the i-th (rate,modulation) combination plus the coding choice n^* for the current transmission. If the preferred SA n^* is infeasible, then we either choose another (rate,modulation) combination plus coding choice arbitrarily or simply let the system idle.

One can see that the new scheduler (6.13) automatically balances the packet reception status (the $\mathbf{q}(t)$ terms), the success overhearing probability of different (rate,modulation) (the $\overline{\mathcal{B}^{\text{in},(i)}(\mathsf{cq}(t))}$ and $\overline{\mathcal{B}^{\text{out},(i)}(\mathsf{cq}(t))}$ terms), and different amount of time it takes to finish transmission of a coded/uncoded packet (the T_i term). In all the numerical experiments of Section 6.4, the new scheduler (6.13) robustly achieves the optimal throughput with adaptive coding and modulation.

6.4 Simulation Results

In this section, we simulate the proposed optimal 7-operation INC + scheduling solution and compare the results with the existing INC solutions and the (back-pressure) pure-routing solutions.

In Fig. 6.2, we simulate a simple time-varying channel situation first described in Section 5.3.1. Specifically, the channel quality $\mathsf{cq}(t)$ is i.i.d. distributed and for any t, $\mathsf{cq}(t)$ is uniformly distributed on $\{1, 2\}$. When $\mathsf{cq}(t) = 1$, the success probabilities are $\vec{p}^{(1)} = (0, 0.5, 0.5, 0)$ and when $\mathsf{cq}(t) = 2$, the success probabilities are $\vec{p}^{(2)} = (0, 0, 0, 1)$, respectively. We consider four different schemes: (i) Back-pressure (BP) + pure routing; (ii) BP + INC with 5 operations [27, 39]; (iii) The proposed DMW+INC with 7 operations, and (iv) The modified DMW+INC with 7 operations that use $q_k^{\text{inter}}(t)$ to compute the back pressure, see (6.12), instead of $q_k(t)$ in (6.1).

We choose perfectly fair $(R_1, R_2) = (\theta, \theta)$ and gradually increase the θ value and plot the stability region. For each experiment, i.e., each θ, we run the schemes for 10^5 time slots. The horizontal axis is the sum rate $R_1 + R_2 = 2\theta$ and the vertical axis is the

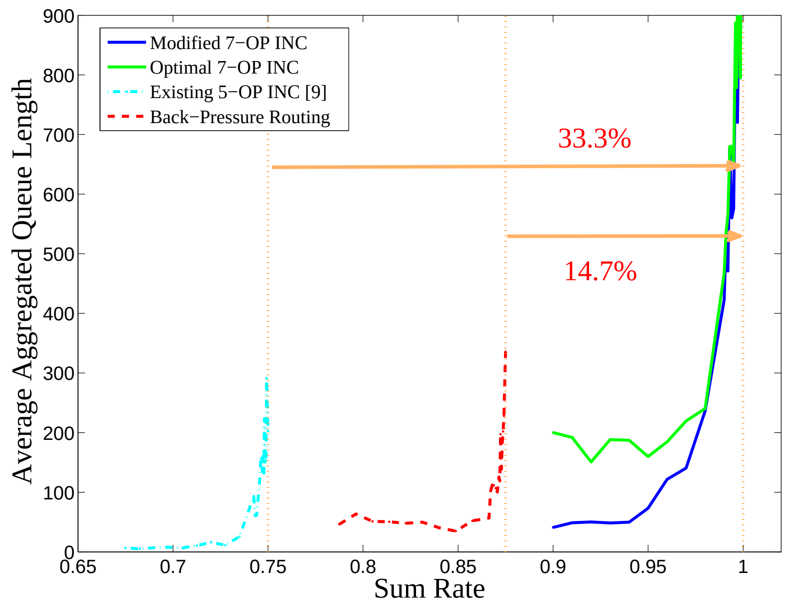
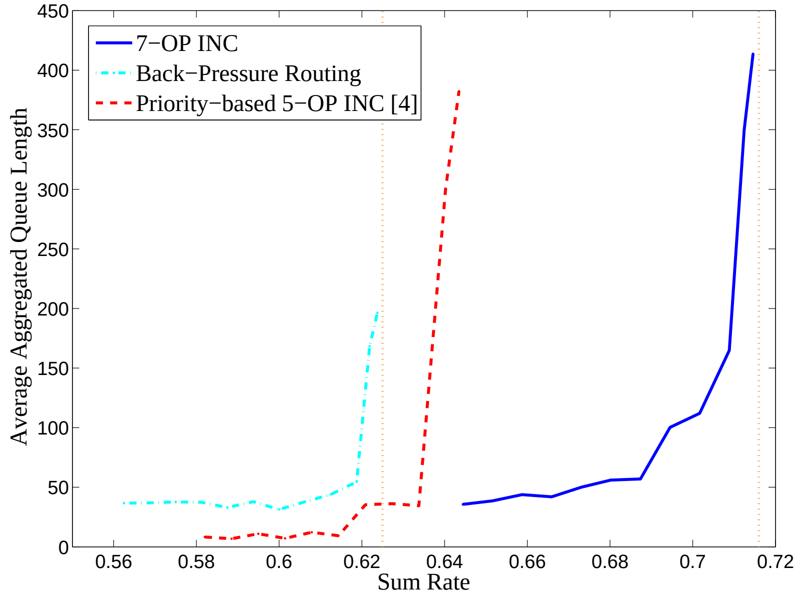
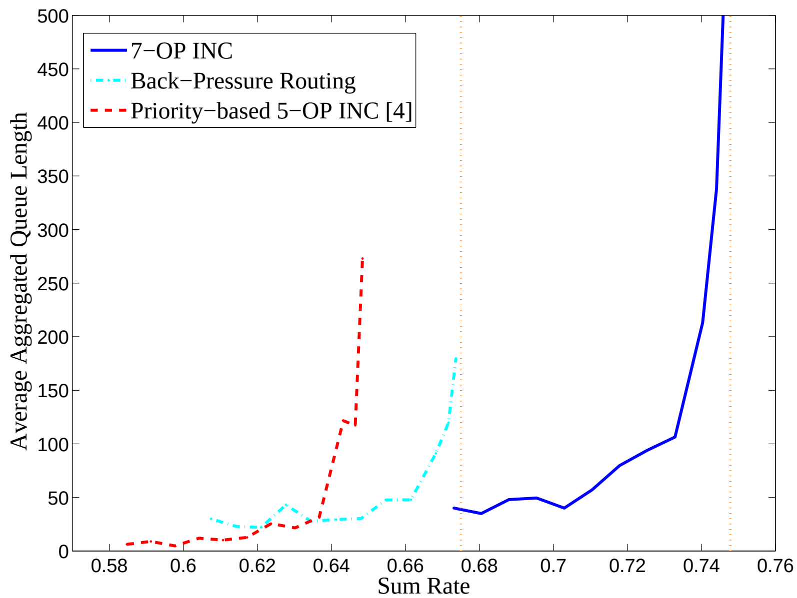


Fig. 6.2.: The backlog of four different schemes for a time-varying channel with $\mathsf{cq}(t)$ uniformly distributed on $\{1,2\}$, and the packet delivery probability being $\vec{p} = (0, 0.5, 0.5, 0)$ if $\mathsf{cq}(t) = 1$ and $\vec{p} = (0, 0, 0, 1)$ if $\mathsf{cq}(t) = 2$.



(a) $(f_1, f_2, f_3, f_4) = (0.15, 0.15, 0.35, 0.35)$.



(b) $(f_1, f_2, f_3, f_4) = (0.25, 0.25, 0.25, 0.25)$.

Fig. 6.3.: The backlog comparison with cq(t) chosen from $\{1, 2, 3, 4\}$ and the probability vectors are $\vec{p}^{(1)} = (0.14, 0.06, 0.56, 0.24)$, $\vec{p}^{(2)} = (0.14, 0.56, 0.06, 0.24)$, $\vec{p}^{(3)} = (0.04, 0.16, 0.16, 0.64)$, and $\vec{p}^{(4)} = (0.49, 0.21, 0.21, 0.09)$.

aggregate backlog (averaged over 10 trials) in the end of 10^5 slots. By the results in [43], the sum rate Shannon capacity is 1 packet/slot, the best possible rate for 5-OP INC is 0.875 packet/slot, and the best pure routing rate is 0.75 packet/slot, which are plotted as vertical lines in Fig. 6.2. The simulation results confirm our analysis. The proposed 7-operation dynamic INC has a stability region matching the Shannon block code capacity and provides 14.7% throughput improvement over the 5-operation INC, and 33.3% over the pure-routing solution.

Also, both our original proposed solution (using $q_k(t)$) and the modified solution (using $q_k^{\text{inter}}(t)$) can approach the stability region while the modified solution has smaller backlog. This phenomenon is observed throughout all our experiments. As a result, in the following experiments, we only report the results of the modified solution using $q_k^{\text{inter}}(t)$ to compute the backpressure.

Next we simulate the scenario of 4 different channel qualities: $\mathsf{CQ} = \{1, 2, 3, 4\}$. The varying channel qualities could model the situations like the different packet transmission rates and loss rates due to time-varying interference caused by the primary traffic in a cognitive radio environment. We assume four possible channel qualities with the corresponding probability distributions being $\vec{p}^{(1)} = (p^{(1)}_{\overline{d_1 d_2}}, p^{(1)}_{d_1\overline{d_2}}, p^{(1)}_{\overline{d_1}d_2}, p^{(1)}_{d_1 d_2}) = (0.14, 0.06, 0.56, 0.24)$, $\vec{p}^{(2)} = (0.14, 0.56, 0.06, 0.24)$, $\vec{p}^{(3)} = (0.04, 0.16, 0.16, 0.64)$, and $\vec{p}^{(4)} = (0.49, 0.21, 0.21, 0.09)$ in both Figs. 6.3(a) and 6.3(b). The difference is that in Fig. 6.3(a), the channel quality $\mathsf{cq}(t)$ is i.i.d. distributed with probability (frequency) (f_1, f_2, f_3, f_4) being $(0.15, 0.15, 0.35, 0.35)$. In Fig. 6.3(b) the $\mathsf{cq}(t)$ is again i.i.d. but with different frequency $(f_1, f_2, f_3, f_4) = (0.25, 0.25, 0.25, 0.25)$. Again, we assume perfect fairness $(R_1, R_2) = (\theta, \theta)$ and gradually increase the θ value. The sum-rate Shannon block-code capacity is $R_1 + R_2 = 0.716$ when $(f_1, f_2, f_3, f_4) = (0.15, 0.15, 0.35, 0.35)$ and $R_1 + R_2 = 0.7478$ when $(f_1, f_2, f_3, f_4) = (0.25, 0.25, 0.25, 0.25)$, and the pure routing sum-rate capacity is $R_1 + R_2 = 0.625$ when $(f_1, f_2, f_3, f_4) = (0.15, 0.15, 0.35, 0.35)$ and $R_1 + R_2 = 0.675$ when $(f_1, f_2, f_3, f_4) = (0.25, 0.25, 0.25, 0.25)$. We simulate our modified 7-OP INC, the priority-based solution in [38], and a standard back-pressure routing

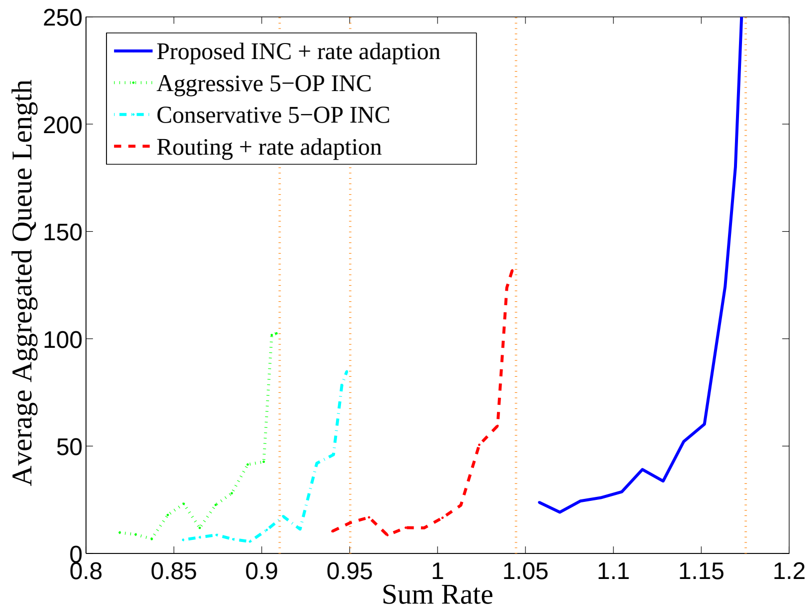


Fig. 6.4.: The backlog of four different schemes for rate adaptation with two possible (error-correcting-code rate,modulation) combinations. The back-pressure-based INC scheme in [39] is used in both aggressive and conservative 5-OP INC, where the former always chooses the high-throughput (rate,modulation) combination while the latter always chooses the low-throughput (rate,modulation) combination.

scheme [28]. Each point of the curves is the average of 10 trials and each trial lasts for 10^5 slots.

Although the priority-based scheduling solution is provably optimal for fixed channel quality, it is less robust and can sometimes be substantially suboptimal (see Fig. 6.3(b)) due to the ad-hoc nature of the priority-based policy. For example, as depicted by Figs. 6.3(a) and 6.3(b), the pure-routing solution outperforms the 5-operation scheme for one set of frequency (f_1, f_2, f_3, f_4) while the order is reversed for another set of frequency. On the other hand, the proposed 7-operation scheme consistently outperforms all the existing solutions and has a stabiliby region matching the Shannon block-code capacity. We have tried many other combinations of time-varying channels. In all our simulations, the proposed DMW scheme always achieves the block-code capacity in [43] and outperforms routing and any existing solutions [38, 39].

Our solution in Section 6.3.1 is the first dynamic INC design that is guaranteed to achieve the optimal linear INC capacity with rate-adaptation (adaptive coding and modulation) [43]. Fig. 6.4 compares its performance with existing routing-based rate-adaptation scheme and the existing INC schemes, the latter of which are designed without rate adaptation. We assume there are two available (error-correcting-code rate,modulation) combinations to be selected. We assume that the first combination takes 1 second to finish transmitting a single packet and the second combination takes 1/3 second to finish a single packet. That is, the transmission rate of the second combination is 3 times faster than the first combination.

We further assume the packet delivery probability is $\vec{p} = (0.1 \cdot 0.05, 0.95 \cdot 0.1, 0.05 \cdot 0.9, 0.95 \cdot 0.9)$ if the first combination is selected and $\vec{p} = (0.6 \cdot 0.8, 0.8 \cdot 0.4, 0.2 \cdot 0.6, 0.2 \cdot 0.4)$ if the second combination is selected. That is, the low-throughput combination is likely to be overheard by both destinations and the high-throughput combination has a much lower success transmission probability. We can compute the corresponding Shannon block-code capacity region by modifying the equations in [43]. We then use the proportional fairness objective function $\xi(R_1, R_2) = \log(R_1) + \log(R_2)$ and find the maximizing R_1^* and R_2^* over the Shannon capacity region, which are $R_1^* = 0.6508$ packets per second and $R_2^* = 0.5245$ packets per second in this example.

After computing the optimal block code capacity, we assume the following dynamic packet arrivals. We define $(R_1, R_2) = \theta \cdot (R_1^*, R_2^*)$ for any given $\theta \in (0, 1)$. For any experiment (i.e., for any given θ), the arrivals of session-i packets is a Poisson random process with rate R_i packets per second for $i = 1, 2$. That is, if the low-throughput combination 1 is selected to transmit 1 packet, then during the 1 second it takes to finish, the number of arrivals of session-i packets is a Possion random variable with mean $R_i \cdot 1$ packets. Similarly, if the high-throughput combination is selected to transmit 1 packet, then during the 1/3 second it takes to finish transmission, the number of arrivals of session-i packets is a Possion random variable with mean $R_i \cdot 1/3$ packets.

Each point of the curves of Fig. 6.4 consists of 10 trials and each trial lasts for 10^5 seconds. We compare the performance of our scheme in Section 6.3.1 with (i) Pure-routing with rate-adaptation; (ii) aggressive 5-OP INC, i.e., use the scheme in [39] and always choose combination 2; and (iii) conservative 5-OP INC, i.e., use the scheme in [39] and always choose combination 1. We also plot the optimal routing-based rate-adaptation rate and the optimal Shannon-block-code capacity rate as vertical lines.

We can observe that since our proposed scheme jointly decides which (rate,modulation) combination to use and which INC operation to encode the packet in an optimal way, see (6.13), the stability region of our scheme matches the block-code Shannon capacity with rate-adaptation. It provides 12.51% throughput improvement over the pure routing-based rate-adaptation solution (which is represented by the red dash line in Fig. 6.4).

Furthermore, we observe that if we perform INC but always choose the low-throughput (rate,modulation), as suggested in some existing works [46], then the largest sum-rate $R_1 + R_2 = \theta^*_{\text{cnsv. 5-OP}}(R_1^* + R_2^*) = 0.9503$, which is worse than pure routing with rate-adaptation $\theta^*_{\text{routing,RA}}(R_1^* + R_2^*) = 1.0446$. Even if we always choose the high-throughput (rate,modulation) with 5-OP INC, then the largest sum-rate $R_1+R_2 = \theta^*_{\text{aggr. 5-OP}}(R_1^*+R_2^*) = 0.9102$ is even worse than the conservative 5-OP INC capacity. We have tried many other rate-adaptation scenarios. In all our simulations, the proposed DMW scheme always achieves the block-code capacity and outperforms pure-routing, conservative 5-OP INC, and aggressive 5-OP INC.

It is worth emphasizing that in our simulation, for any fixed (rate,modulation) combination, the channel quality is also fixed. Therefore since 5-OP scheme is throughput optimal for fixed channel quality [3], it is guaranteed that the 5-OP scheme is throughput optimal when using a fixed (rate,modulation) combination. Our results thus show that using a fixed (rate,modulation) combination is the main reason of the suboptimal performance and the proposed scheme in (5.2), (6.2), and (6.13) can dynamically decide which (rate,modulation) combination to use for each transmission and achieve the largest possible stability region.

6.5 Chapter Summary

In this chapter, we have proposed a new 7-operation INC scheme together with the corresponding scheduling algorithm to achieve the optimal downlink throughput of the 2-flow access point network with time varying channels. Based on binary XOR operations, the proposed solution admits ultra-low encoding/decoding complexity with efficient buffer management and minimal communication and control overhead. The proposed algorithm has also been generalized for rate adaptation and it again robustly achieves the optimal throughput in all the numerical experiments. The proposed algorithm has also been generalized for rate adaptation and it again robustly achieves the optimal throughput in all the numerical experiments. A byproduct of this paper is a throughput-optimal scheduling solution for SPNs with random departure, which could further broaden the applications of SPNs to other real-world applications.

7. 802.11-BASED INTER-SESSION NETWORK CODING MAC PROTOCOL WITH RATE ADAPTION

In this chapter, we propose an 802.11-based inter-session network coding MAC protocol with rate adaption, which can be considered as an extension of the rate adaption solution proposed in the previous chapter.

7.1 A Brief On 802.11 MAC

We build the proposed practical protocol on top of CSMA-CA, which is widely adapted in the industry standards, e.g. IEEE 802.11, IEEE 802.15.4, and ITU-T G.hm. Furthermore, we are targeting building our protocol base on 802.11n MAC with minimum possible rule changes and the backward compatibility, and hence it provides a great implementation potential and a fair comparison with a wide range of existing wireless devices. We begin the discussion with an overview of 802.11 MAC protocol.

7.1.1 Network Topology

In 802.11, the access point (AP) is the basic unit of the wireless service provider. It allows the surrounding wireless devices, which are called as stations (STAs) conventionally, to access the Internet through it. Each AP then forms a basic service set (BSS) with an associated basic service set ID (BSSID) which is broadcasted periodically in the beacon frame. When one STA attempts to access to Internet through 802.11, it first tries to receive the surrounding beacon frames and retrieving the BSSID to the user. The user then selects the desired BSSID, and connect to the wireless service.

7.1.2 The Basic Mechanisms

The main functionality of 802.11 MAC is Distributed Coordination Function (DCF), which is later extended to Hybrid Coordination Function (HCF) after 802.11e. Under DCF rules, each wireless device, which attempts to send packets, first needs to detect the medium being idle for DCF InterFrame Space (DIFS) duration, and then wait for another *random backoff* duration, which is randomly drawn from a predefined contention window (CW). Only after the medium remains idle for the random backoff duration, should the device start transmitting the packets.

Once the device wins the medium access control and start to send packets, it reserves the right of accessing the medium for the duration of Transmit Opportunity (TXOP). The device then can send as many packets as possible during the available TXOP.

After 802.11e, DCF was modified to accommodate the concept of Quality of Service (QoS), and was called HCF afterward. HCF categorizes the incoming packets from up-layers into 4 different access categories (AC), called "Voice (AC_VO)," "Video (AC_VI)," "Best Effort (AC_BE)", and "Background (AC_BK)." Each category has the different durations of Arbitration InterFrame Space (AIFS), which replaces the role of DIFS, different contention window sizes, and different durations of transmit opportunity (TXOP), to address the priorities of each AC. Generally speaking, HCF assigns shorter duration of AIFS, smaller contention window, and longer TXOP to the higher priority AC.

The RTS-CTS hand-shaking mechanism is an optional feature in 802.11MAC. When RTS-CTS is on, any transmission-attempting device first need to broadcast an request-to-send (RTS) packet to the desired destination device. The destination device then replies a clear-to-send (CTS) packet back if it is idle and the medium is clear. Once CTS is received by the transmission-attempting device, it then starts to send the packets. However, in general, this feature is usually turned off in most of deployed 802.11 devices.

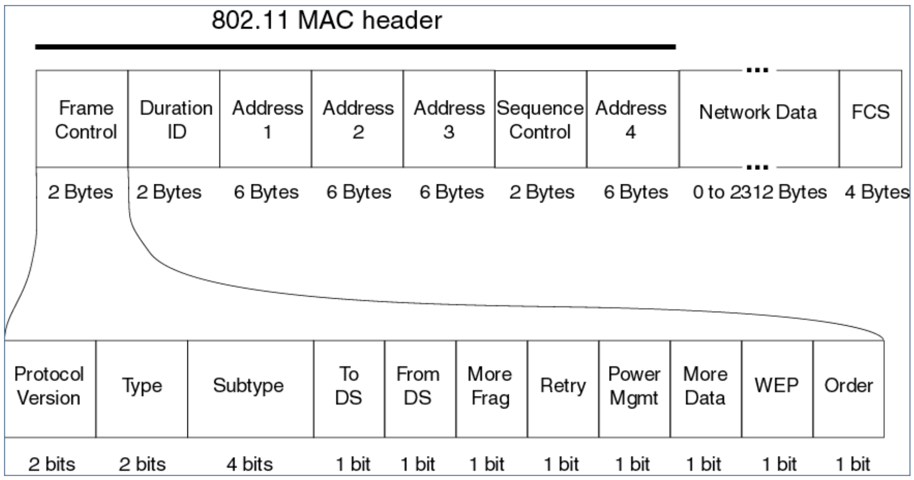


Fig. 7.1.: The MAC header in 802.11 [47].

Table 7.1: The address field in 802.11 MAC header.

To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	Destination	Source	BSSID	N/A
0	1	Destination	BSSID	Source	N/A
1	0	BSSID	Source	Destination	N/A
1	1	Receiver	Transmitter	Destination	Source

7.1.3 Frame Structure

In 802.11 MAC, the basic frame transmitted in MAC is called MAC Protocol Data Unit (MPDU). The frame structure of one MPDU is illustrated in Fig. 7.1.

The 802.11 MAC header consists of 30 bytes of header fields. The first two bytes are for Frame Control. It is worth mentioning the “To DS” and “From DS” fields. These two fields are designed for the scenarios the packet is sent to or from distributed systems (DS). And the contents of Address 1 to Address 4 will also change according to “To DS” and “From DS.” For example, in the case that both the source and the destination are not from distributed systems, Address 1 denotes the destination ID and Address 2 denotes the Source ID while Address 3 contains the BSSID. The detailed relationship between (To DS, From DS) pairs and the contents in Address 1 to Address 4 is illustrated in Table 7.1.

The other thing in the 802.11 MAC header that is worth mentioning is Sequence Control field. In Sequence Control field, the first 12 bits describe Sequence Number for that packet, and the last 4 bits describe Fraction Number for the case that the packet is partitioned into several pieces.

After 802.11e and 802.11n, the idea of aggregate-MPDU (A-MPDU) is introduced in the standard to reduce the medium contention overhead and ACK/SIFS overhead, and now has become a required feature. Specifically speaking, once the device wins the medium access, it reserves the right to transmit for a duration of TXOP. Within this duration of TXOP, it consecutively sending MPDUs without requiring any ACK/NACK. And at the end of the TXOP with properly reserved time slots for feedbacks, the transmitting device sends the request for Block-ACK to the receiver. The receiver then replies Block-ACK, which contains ACKs for each individual MPDU received during this TXOP, to the transmitter. This procedure is called *A-MPDU and Block-ACK* mechanism.

7.2 Practical MAC Protocol Based On 802.11n CSMA-CA

Before we explicitly describe the protocol, we first state the assumptions.

Assumption 1: In this protocol, we only focus on one category of packets, e.g. Best Effort.

That is, all the incoming packets from upper layers have the same priority and the similar packet lengths.

Assumption 2: We assume RTS-CTS feature is off.

Assumption 3: We consider only two fixed receiving devices (STAs) with one Access Point (AP). That is, the channel condition changes only according to the chosen rate by AP, and all the transmissions happen within one basic service set (BSS).

Assumption 4: We assume AP has the capability to store two separate queues of packets, one for each STA.

Assumption 5: We assume the features of aggregate-MPDU (A-MPDU) and Block-ACK are on. That is, each transmission consists of multiple consecutive MPDUs (packets) without either ACKs or IFS in between. And after the entire transmission is finished, we use Block-ACK to provide the feedback for each MPDU.

7.2.1 Additional Fields in 802.11 MAC Headers

Our MAC protocol is mainly based on CSMA-CA specified in 802.11n. Other than the MAC header specified in 802.11n, we also introduce one new field, called ”Cross-Session Indicator (CSI),” which is required to be included in the frame control of the MAC header. We also need to modify the contents of two existing fields (Address 4 and Sequence Control).

New Field: In the Frame Control of 802.11 MAC header, we need to introduce a one-bit field, Cross-Session Indicator. When CSI is set to be 1, it implies that the content in the following A-MPDU comes from both sessions. Otherwise, CSI is set to be 0.

Modified Field 1: As stated in Assumption 3, we only focus on the scenarios with transmissions happen within one BSS. As a result, the two bits ”To DS” and ”From DS,” which indicate whether the packet is sent to or from distributed systems, will always be set to be both zeros. Given such fact, we let Address 4 stores the second Destination if CSI is set to be 1. Otherwise, Address 4 remains N/A.

Modified Field 2: We need to increase the length of Sequence Control to 4 Bytes, which contains Sequence Number 1 (12 bits), Fragment Number 1 (4 bits), Sequence Number 2 (12 bits), and Fragment Number 2 (4 bits). The Sequence Number 1 and Fragment Number 1 correspond to the packets sent to Destination 1 specified in Address 1. Sequence Number 2 and Fragment Number 2 correspond to the packets sent to Destination 2 specified in Address 4 if CSI is set to be 1, and both of them are set to be N/A if CSI is 0.

In summary, we need to increase the original MAC header length by 17 bits, which includes one bit for CSI and 2 bytes for Sequence Control. We further notice that even though our protocol is based on 802.11n, but the changes can be readily extended to 802.11ac as well. The reason is because the MAC level changes from 802.11n to 802.11ac are mainly to deal with the bandwidth expansion without changing the fundamental CSMA/CA rules.

7.2.2 When AP Wins The Medium Access Control

We let each device contends the medium access control through the regular CSMA-CA mechanism described in 802.11. We then state the operations needed to be perform at AP when AP wins the medium access control.

AP maintains 5 values, which are initially zeros with names $q^1_\emptyset$, $q^2_\emptyset$, $q^1_{\{2\}}$, $q^2_{\{1\}}$, and q_{mix}. AP will make transmission decision base on these values before each transmission, and update them during the transmission and after receiving Block-ACK.

Mode Selection

In our protocol, there are three possible modes that are available to be chosen for each TXOP transmission. We first discuss how to choose the mode at the beginning of the TXOP transmission, and then we describe the transmission details with the chosen mode.

Let $T_{\text{1BA}} = T_{\text{TXOP}} - T_{\text{SIFS}} - T_{\text{BA}}$ and $T_{\text{2BA}} = T_{\text{TXOP}} - 2(T_{\text{SIFS}} - T_{\text{BA}})$, where T_{TXOP} is the time duration for the remaining available TXOP in the current transmission, T_{SIFS} is the time duration for SIFS, and T_{BA} is the time duration for Block-ACK. Equivalently speaking, T_{1BA} is the time duration that is available for transmitting MPDUs if we require only one Block-ACK, and T_{2BA} is the time duration that is available for transmitting MPDUs if we require two Block-ACKs. Let L be the length of one MPDU. Without loss of generality, we assume T_{1BA} can support at least one MPDU transmission, otherwise AP simply skips this transmission. We also assume there are 7 modulation and coding scheme (MCS) combinations are available for AP to send packets, and denote D_i as the data rate under the i-th MCS combination. After AP wins the medium access control, AP first calculates $N_{\text{MPDU},i}(m)$, the number of MPDUs can be transmitted under Mode m if i-th MCS is chosen. To be more specific,

$$N_{\text{MPDU},i}(m) = \begin{cases} \lfloor (T_{\text{1BA}} \cdot D_i)/L \rfloor & \text{if } m = 1,2 \\ \lfloor (T_{\text{2BA}} \cdot D_i)/L \rfloor & \text{if } m = 3 \end{cases} \tag{7.1}$$

For mode $m = 1$, each MCS $i = 1, 2, ..., 7$, and each MPDU transmission $j = 1$ to $N_{\text{MPDU},i}(m)$, AP computes

$$\text{BP}_{1,i,\text{NC1}}(j) = q^1_{\emptyset}(j) \cdot D_i \cdot (p_{d_1 d_2,i} + p_{\overline{d_1} d_2,i} + p_{d_1 \overline{d_2},i}) - q^1_{\{2\}}(j) \cdot D_i \cdot p_{\overline{d_1} d_2,i}, \tag{7.2}$$

$$\text{BP}_{1,i,\text{DX1}}(j) = q^1_{\{2\}}(j) \cdot D_i \cdot (p_{d_1 d_2,i} + p_{d_1 \overline{d_2},i}). \tag{7.3}$$

We then "virtually" choose the preferred SA under i-th MCS and j-th MPDU transmission based on the maximum of $\text{BP}_{1,i,\text{NC1}}(j)$ and $\text{BP}_{1,i,\text{DX1}}(j)$, denoted by $\text{BP}_{1,i}(j)$, and update the virtual queues $\mathbf{q}(j+1)$ by the average service matrix and the preferred SA.

Finally, AP computes

$$\text{BP}_{1,i} = \sum_{j=1}^{N_{\text{MPDU},i}(m)} \text{BP}_{1,i}(j) \tag{7.4}$$

for each MCS combination $i = 1, 2, ..., 7$, and

$$\text{BP}_1 = \max_{i=1,2,...,7} \text{BP}_{1,i} \tag{7.5}$$

Symmetric operations can be carried out for Mode $m = 2$ by replacing $q^1_{\emptyset}(j)$ by $q^2_{\emptyset}(j)$ and replacing $q^1_{\{2\}}(j)$ by $q^2_{\{1\}}(j)$. And thus derive BP_2 and its preliminary quantities.

For Mode $m = 3$, each MCS $i = 1, 2, ..., 7$, and each MPDU transmission $j = 1$ to $N_{\mathrm{MPDU},i}(m)$, AP computes

$$
\begin{aligned}
\mathrm{BP}_{3,i,\mathrm{NC1}} =& (q^1_{\emptyset}(j) \cdot D_i \cdot (p_{d_1 d_2,i} + p_{\overline{d_1} d_2,i} + p_{d_1 \overline{d_2},i}) - q^1_{\{2\}}(j) \cdot D_i \cdot p_{\overline{d_1} d_2,i}),\\
\mathrm{BP}_{3,i,\mathrm{NC2}} =& (q^2_{\emptyset}(j) \cdot D_i \cdot (p_{d_1 d_2,i} + p_{\overline{d_1} d_2,i} + p_{d_1 \overline{d_2},i}) - q^2_{\{1\}}(j) \cdot D_i \cdot p_{d_1 \overline{d_2},i}),\\
\mathrm{BP}_{3,i,\mathrm{DX1}} =& q^1_{\{2\}}(j) \cdot D_i \cdot (p_{d_1 d_2,i} + p_{d_1 \overline{d_2},i}),\\
\mathrm{BP}_{3,i,\mathrm{DX2}} =& q^2_{\{1\}}(j) \cdot D_i \cdot (p_{d_1 d_2,i} + p_{\overline{d_1} d_2,i}),\\
\mathrm{BP}_{3,i,\mathrm{PM}} =& (q^1_{\emptyset}(j) + q^2_{\emptyset}(j) - 2q_{\mathrm{mix}}(j)) \cdot D_i \cdot (p_{d_1 d_2,i} + p_{\overline{d_1} d_2,i} + p_{d_1 \overline{d_2},i}),\\
\mathrm{BP}_{3,i,\mathrm{RC}} =& (2q_{\mathrm{mix}}(j) \cdot D_i \cdot (p_{d_1 d_2,i} + p_{\overline{d_1} d_2,i} + p_{d_1 \overline{d_2},i}) - q^1_{\{2\}}(j) \cdot D_i \cdot p_{\overline{d_1} d_2,i}\\
& - q^2_{\{1\}}(j) \cdot D_i \cdot p_{d_1 \overline{d_2},i}),\\
\mathrm{BP}_{3,i,\mathrm{CX}} =& (q^1_{\{2\}}(j) \cdot (p_{d_1 d_2,i} + p_{d_1 \overline{d_2},i}) + q^2_{\{1\}}(j) \cdot (p_{d_1 d_2,i} + p_{\overline{d_1} d_2,i})) \cdot D_i.
\end{aligned}
$$

We then "virtually" choose the preferred SA under i-th MCS and j-th MPDU transmission based on the maximum of the above 7 quantities, denoted by $\mathrm{BP}_{3,i}(j)$, and update the virtual queues $\mathbf{q}(j+1)$ by the average service matrix and the preferred SA.

Finally, AP computes

$$\mathrm{BP}_{3,i} = \sum_{j=1}^{N_{\mathrm{MPDU},i}(m)} \mathrm{BP}_{3,i}(j) \tag{7.6}$$

for each MCS combination $i = 1, 2, ..., 7$, and

$$\mathrm{BP}_3 = \max_{i=1,2,...,7} \mathrm{BP}_{3,i} \tag{7.7}$$

Mode Transmission

Let $m^* = \arg\max_m \mathrm{BP}_m$. If $m^* = 1, 2$, then AP simply send the uncoded Session-m^* packets using the i-th coding and modulation combination which gives the largest BP_{m^*} during the computation in this TXOP transmission.

If $m^* = 3$, AP then sends the packet according to the computed schedule during the computation of BP_3 using the i-th coding and modulation combination which gives the largest BP_{m^*}.

At the end of this transmission, i.e. 1 T_{BA} left for $m^* = 1, 2$ and 2 T_{BA} left for $m^* = 3$, AP then sends the Block-ACK request along with the assigned Block-ACK schedule to the destination(s). Upon the reception of Block-ACKs, AP then update the virtual queues according to the rules specified in the previous chapter.

7.2.3 When STA Receives Packets

STA stores and drops the overhearing packets according to the rules specified in the previous chapter regardless whether CSI is 0 or 1.

If CSI is set to be 1 in the received transmission and either Address 1 or Address 4 matches the address of this STA, then this STA need to send Block-ACK according to the designate schedule sent by AP at the end of the transmission.

If CSI is set to be 0 and Address 1 does not match the address of this STA, then whenever STA has the opportunity to send packets to AP, STA first sends an overhearing report, which contains the Block-ACK with respect to the overhearing packets so far, to AP.

7.2.4 The Construction of The Event Probabilities Under Each Coding And Modulation Combination At AP

To calculate the quantities introduced in Section 7.2.2, one key ingredient is that AP must have the information of the event probabilities under each coding and modulation combination. Such information can be collected by AP distributively in two approaches, which should be applied jointly, without alternating any existing 802.11 transmission behaviors.

The first approach is through the channel estimation. Since 802.11n, MIMO has been standardized into the specification to increase the data throughput. To establish the MIMO transmission, the first step is to let the receiver and the transmitter exchange the channel estimation information to calculate the rank of the channel matrix. With such a channel estimation information, AP can also estimate the event probabilities by assuming the independence between multiple receivers and applying the modified Shannon equation with proper adjusted coefficients.

Of course it is only an estimation through the first approach, but we can further improve its accuracy through the second approach. In the second approach, AP iteratively adjusts the numbers derived in the first approach according to the feedbacks through transmissions. For example, if AP receives an ACK from the receiver, then AP can increment the corresponding event probability.

To reduce the computation complexity, we could also quantized the floating-point numbers representing probabilities into several levels of integers depending on how accurate the calculation needs to be.

7.3 Chapter Summary

In this chapter, we introduce a new 802.11-based inter-session network coding MAC protocol with rate adaption scheme. We begin this chapter by a brief review on the 802.11 MAC protocol, which includes the basic medium contention mechanism and the frame structure, in Section 7.1. We then formally describe the proposed inter-session network coding MAC protocol in Section 7.2. To be more specific, we introduce the required new headers in Section 7.2.1, and show that the new headers only cost only 17 bits. Then in Section 7.2.2 and Section 7.2.3, we detail the behaviors that need to be performed at AP and STAs. Finally, we discuss how to construct the event probability at AP in Section 7.2.4.

8. CONCLUSION AND FUTURE WORK

In this thesis, we first propose a space-based LNC scheme to characterize the Shannon capacity of the COPE principle 2-user wireless butterfly network with broadcast packet erasure channels, which incorporates the broadcast packet erasure channels with feedback, the COPE principle, and the opportunistic routing all together. In the second part, we further extend our attention to the inter-session network coding scheduling problems with dynamic packet arrivals, which gives a closer connection between our solutions to the practical implementations. And we propose a new 7-operation INC scheme together with the corresponding scheduling algorithm to achieve the optimal downlink throughput of the 2-flow access point network with time varying channels, and has been generalized for rate adaption scenarios. Last but not least, in the third part, we go one step more toward the practical implementation by proposing an 802.11-based inter-session network coding MAC protocol with rate adaption mechanism. The proposed solution delicately retain the CSMA-CA distributed contention mechanism with only 17 bits new header field changes. The new solution demonstrates concrete throughput improvement without alternating too much packet-by-packet traffic behavior. Such a feature is critical in practical implementation since it allows the network coding solution to be transparent to any arbitrary upper layer applications.

In Chapter 1, we discuss the network coding gain for three local wireless network topologies, including the broadcast packet erasure channel with feedback, the COPE principle wireless butterfly network, and the opportunistic routing. All of them exhibit significant end-to-end throughput improvement when network coding can be utilized. In Chapter 2, we propose a local network topology which incorporates all the three local wireless network together, name the “COPE principle 2-user wireless butterfly network with broadcast packet erasure channels.” The space-based LNC scheme established in Chapter 3 provides a intuitive and systematical approach to exploit the possible joint inter-session and inter-session network coding gain in the network. With the help of space-based LNC scheme, in Chapter 4, we characterize the Shannon capacity of the COPE principle 2-user wireless butterfly network with broadcast packet erasure channel and demonstrate significant throughput improvement compared with existing solutions. We further extend the block-code-based model to the dynamic packet arrival model, so called the stability analysis. This extension promotes the proposed solution one step closer to practical implementations. However, there exist crucial issues and challenging the inter-session network coding stability analysis, as discussed in Chapter 5. In Chapter 6, we robustly and optimally solve this problem for 2-flow downlink time-varying broadcast PEC, and extend the result to accommodate the rate adaption scenarios. Finally, in Chapter 7, we propose an 802.11-based inter-session network coding MAC protocol with rate adaption, and demonstrate concrete throughput improvement with significant implementation potential.

Even though we build up the analysis tool based on the local wireless network, those tools reveal great insight about the possible joint intra-session and inter-session linear network coding gain. Our next step is to verify our solution in software-defined-networks.

LIST OF REFERENCES

LIST OF REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, “Network information flow,” *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] S.-Y. Li, R. Yeung, and N. Cai, “Linear network coding,” *IEEE Trans. Inf. Theory*, vol. 49, pp. 371–381, Feb 2003.

[3] 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), pp. 54–61, June 2009.

[4] C.-C. Wang, “On the capacity of 1-to-K broadcast packet erasure channels with channel output feedback,” *IEEE Trans. Inf. Theory*, vol. 58, pp. 931–956, Feb 2012.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, “XORs in the air: Practical wireless network,” in *Proc. ACM Special Interest Group on Data Commun. (SIGCOMM)*, 2006.

[6] P. Chaporkar and A. Proutiere, “Adaptive network coding and scheduling for maximizing throughput in wireless networks,” in *Proc. Ann. ACM Int. Conf. on Mobile Computing & Networking (MobiCom)*, (Montreal, QC, Canada), pp. 135–146, Sept 2007.

[7] T. Cui, L. Chen, and T. Ho, “Energy efficient opportunistic network coding for wireless networks,” in *Proc. 27th IEEE Conf. Computer Communications (INFOCOM)*, (Phoenix, AZ), Apr 2008.

[8] W.-C. Kuo and C.-C. Wang, “On the capacity of 2-user 1-hop relay erasure networks—the union of feedback, scheduling, opportunistic routing, and network coding,” in *Proc. IEEE Int. Symp. Inform. Theory*, (Saint Petersburg, Russia), Aug 2011.

[9] B. Rankov and A. Wittneben, “Achievable rate regions for the two-way relay channel,” in *Proc. IEEE Int. Symp. Inform. Theory*, (Seattle, WA), 2006.

[10] K. Salamatian and R. Khalili, “An information theory for erasure channels,” in *Proc. 43rd Ann. Allerton Conf. on Comm.*, (Monticello, IL), Sept 2005.

[11] C. Suh and K. Ramachandran, “Exact-repair MDS code construction using interference alignment,” *IEEE Trans. Inf. Theory*, vol. 57, pp. 1425–1442, Mar 2011.

[12] J. Yoo, T. Liu, and F. Xue, “Gaussian broadcast channels with receiver message side information,” in *Proc. IEEE Int. Symp. Inform. Theory*, (Seoul, Korea), June 2009.

[13] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sengupta, "Loss-aware network coding for unicast wireless sessions: design, implementation, and performance evaluation," in *Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, (Annapolis, MD, USA), pp. 85–96, 2008.

[14] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: The case for coding-aware routing," in *Proc. 26th IEEE Conf. Computer Communications (INFOCOM)*, (Anchorage, Alaska, USA), pp. 1028–1036, May 2007.

[15] H. Seferoglu, A. Markopoulou, and K. K. Ramakrishnan, "I^2NC: Intra- and inter-session network coding for unicast flows in wireless networks," in *Proc. 30th IEEE Conf. Computer Communications (INFOCOM)*, (Shanghai, China), pp. 1035–1043, April 2011.

[16] L. Ozarow and S. Leung-Yan-Cheong, "An achievable region and outer bound for the Gaussian broadcast channel with feedback," *IEEE Trans. Inf. Theory*, vol. 30, July 1984.

[17] A. Eryilmaz, D. Lun, and B. Swapna, "Control of multi-hop communication networks for inter-session network coding," *IEEE Trans. Inf. Theory*, vol. 57, pp. 1092–1110, Feb. 2011.

[18] C.-C. Wang, "On the capacity of wireless 1-hop intersession network coding — a broadcast packet erasure channel approach," *IEEE Trans. on Information Theory*, vol. 58, pp. 957–988, Feb 2012.

[19] J. Korner and K. Marton, "General broadcast channels with degraded message sets," *IEEE Trans. Inf. Theory*, vol. 23, pp. 60–64, Jan 1997.

[20] G. Kramer and S. Shamai, "Capacity for classes of broadcast channels with receiver side information," in *Proc. IEEE Inform. Theory Workshop*, (Lake Tahoe, CA), pp. 313–318, Sept 2007.

[21] B. Smith and B. Hassibi, "Wireless erasure networks with feedback," in *Proc. IEEE Int. Symp. Inform. Theory*, pp. 339–343, July 2008.

[22] C.-C. Wang and N. Shroff, "Beyond the butterfly—a graph-theoretic characterization of the feasibility of network coding with two simple unicast sessions," in *Proc. IEEE Int. Symp. Inform. Theory*, (Nice, France), pp. 121–125, June 2007.

[23] F. Xue and X. Yang, "Network coding and packet-erasure broadcast channel," in *Proc. 5th IEEE Ann. Communications Society Conf. on Sensor, Mesh and Ad Hoc Communications and Networks Workshops (SECON)*, (San Francisco, CA), July 2008.

[24] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. ACM Special Interest Group on Data Commun. (SIGCOMM)*, (Kyoto, Japan), Aug 2007.

[25] D. Koutsonikolas, C.-C. Wang, and Y. Hu, "CCACK: Efficient network coding based opportunistic routing through cumulative coded acknowledgments," in *Proc. 29th IEEE Conf. Computer Communications (INFOCOM)*, (San Diego, CA), pp. 1–9, Mar 2010.

[26] X. Lin, N. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 1452–1463, Aug 2006.

[27] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource allocation and cross-layer control in wireless networks*. Now Pub, 2006.

[28] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *Automatic Control, IEEE Transactions on*, vol. 37, no. 12, pp. 1936–1948, 1992.

[29] Y. E. Sagduyu and A. Ephremides, "On broadcast stability region in random access through network coding," in *44th Allerton Annual Conference on Communication, Control and Computing*, p. 38, 2006.

[30] G. Paschos, L. Georgiadis, and L. Tassiulas, "Scheduling with pairwise xoring of packets under statistical overhearing information and feedback," *Queueing Systems*, vol. 72, no. 3-4, pp. 361–395, 2012.

[31] L. Jiang and J. Walrand, "Stable and utility-maximizing scheduling for stochastic processing networks," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pp. 1111–1119, IEEE, 2009.

[32] L. Huang and M. J. Neely, "Utility optimal scheduling in processing networks," *Performance Evaluation*, vol. 68, no. 11, pp. 1002–1021, 2011.

[33] V. Anantharam and S. Verdtu, "Bits through queues," *IEEE Trans. Inf. Theory*, vol. 42, pp. 4–18, 1996.

[34] T. Lutz, G. Kramer, and C. Hausl, "Capacity for half-duplex line networks with two sources," in *Proc. IEEE Int'l Symp. Inform. Theory*, (Austin, Texas, USA), June 2010.

[35] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[36] A. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Trans. Inf. Theory*, vol. 52, pp. 789–804, Mar. 2006.

[37] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," *Information Theory, IEEE Transactions on*, vol. 55, no. 2, pp. 797–815, 2009.

[38] Y. Sagduyu, L. Georgiadis, L. Tassiulas, and A. Ephremides, "Capacity and stable throughput regions for the broadcast erasure channel with feedback: An unusual union," *Information Theory, IEEE Transactions on*, vol. 59, no. 5, pp. 2841–2862, 2013.

[39] S. A. Athanasiadou, M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Stable and capacity achieving xor–based policies for the broadcast erasure channel with feedback," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, IEEE, 2013.

[40] C.-C. Wang and D. J. Love, "Linear network coding capacity region of 2-receiver mimo broadcast packet erasure channels with feedback," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pp. 2062–2066, IEEE, 2012.

[41] G. Paschos, C. Fragiadakis, L. Georgiadis, and L. Tassiulas, “Wireless network coding with partial overhearing information,” in *INFOCOM, 2013 Proceedings IEEE*, pp. 2337–2345, April 2013.

[42] W.-C. Kuo and C.-C. Wang, “Two-flow capacity region of the cope principle for wireless butterfly networks with broadcast erasure channels,” *Information Theory, IEEE Transactions on*, vol. 59, pp. 7553–7575, Nov 2013.

[43] C.-C. Wang and J. Han, “The capacity region of 2-receiver multiple-input broadcast packet erasure channels with channel output feedback,” *Information Theory, IEEE Transactions on*, 2014.

[44] L. Amini, N. Jain, A. Sehgal, J. Silber, and O. Verscheure, “Adaptive control of extreme-scale stream processing systems,” in *Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference on*, pp. 71–71, IEEE, 2006.

[45] M. Zaharia, A. Konwinski, A. D. Joseph, R. Katz, and I. Stoica, “Improving mapreduce performance in heterogeneous environments,” in *Proceedings of the 8th USENIX conference on Operating systems design and implementation*, pp. 29–42, 2008.

[46] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sengupta, “Loss-aware network coding for unicast wireless sessions: Design, implementation, and performance evaluation,” in *SIGMETRICS*, Annapolis, Maryland, USA, June 2008.

[47] WildPackets, “Packets and Protocols.” `http://www.wildpackets.com/resources/compendium/wireless_lan/wlan_packets`. [Online; accessed 3-March-2015].

[48] A. El Gamal, “The feedback capacity of degraded broadcast channels,” *IEEE Trans. Inf. Theory*, vol. 25, pp. 379–381, March 1978.

[49] R. Durrett, *Probability: Theory and Examples*. New York, NY, USA: Cambridge University Press, 4th ed., 2010.

APPENDICES

A. THE CONVERSE OF THE SHANNON CAPACITY

In this appendix, we prove Proposition 4.2.2. For any joint scheduling and NC scheme, we choose t_{s_i} (resp. t_r) as the normalized *expected number of time slots for which* s_i *(resp.* r*) is scheduled*. Namely,

$$t_{s_i} \triangleq \frac{1}{n}\mathsf{E}\left\{\sum_{\tau=1}^{n} 1_{\{\sigma(\tau)=s_i\}}\right\} \text{ and } t_r \triangleq \frac{1}{n}\mathsf{E}\left\{\sum_{\tau=1}^{n} 1_{\{\sigma(\tau)=r\}}\right\}.$$

By definition, t_{s_1}, t_{s_2}, and t_r must satisfy (4.5).

In the subsequent proofs, the logarithm is taken with base q. We prove (4.6) first. To that end, we notice that

$$I(\mathbf{W}_1;\hat{\mathbf{W}}_1) \leq I(\mathbf{W}_1;[\mathbf{Y}_{\{s_1,s_2,r\}\to d_1},\mathbf{Z}]_1^n) \tag{A.1}$$

$$=I(\mathbf{W}_1;[\mathbf{Z}]_1^n)+I(\mathbf{W}_1;[\mathbf{Y}_{\{s_1,s_2,r\}\to d_1}]_1^n|[\mathbf{Z}]_1^n) \tag{A.2}$$

$$\leq I(\mathbf{W}_1;[\mathbf{Y}_{\{s_1,s_2\}\to\{d_1,r\}}]_1^n|[\mathbf{Z}]_1^n) \tag{A.3}$$

$$=I(\mathbf{W}_1;[\mathbf{Y}_{s_1\to\{d_1,r\}}]_1^n|[\mathbf{Z}]_1^n) \tag{A.4}$$

$$\leq H([\mathbf{Y}_{s_1\to\{d_1,r\}}]_1^n|[\mathbf{Z}]_1^n)$$

$$=\sum_{t=1}^{n} H(\mathbf{Y}_{s_1\to\{d_1,r\}}(t)|[\mathbf{Z}]_1^n,[\mathbf{Y}_{s_1\to\{d_1,r\}}]_1^{t-1}) \tag{A.5}$$

$$\leq \sum_{t=1}^{n}\mathsf{E}\left\{1_{\{Z_{s_1\to d_1}(t)=1 \text{ or } Z_{s_1\to r}(t)=1\}}\circ 1_{\{\sigma(t)=s_1\}}\right\} \tag{A.6}$$

$$=nt_{s_1}p_1(d_1,r) \tag{A.7}$$

where (A.1) follow from (2.4); (A.2) follows from the chain rule; (A.3) follows from (2.3), the data processing inequality, and the fact that $\mathbf{Z}$ is independent of $\mathbf{W}_1$; (A.4) follows from that conditioning on $\mathbf{Z}$ (and σ since σ is a function of $\mathbf{Z}$) $\mathbf{Y}_{s_2\to\{d_1,r\}}$ is a function of $\mathbf{W}_2$ and is thus independent of $\mathbf{W}_1$; (A.5) follows from the chain rule;

(A.6) follows from that only when $1_{\{Z_{s_1\to d_1}(t)=1 \text{ or } Z_{s_1\to r}(t)=1\}} = 1$ and $1_{\{\sigma(t)=s_1\}} = 1$ will we have a non-zero entropy value $H(\mathbf{Y}_{s_1\to\{d_1,r\}}(t)|[\mathbf{Z}]_1^n, [\mathbf{Y}_{s_1\to\{d_1,r\}}]_1^{t-1})$, and when $H(\mathbf{Y}_{s_1\to\{d_1,r\}}(t)|[\mathbf{Z}]_1^n, [\mathbf{Y}_{s_1\to\{d_1,r\}}]_1^{t-1}) > 0$, it is upper bounded by 1 since the base of the logarithm is q; (A.7) follows from Wald's lemma.

On the other hand, Fano's inequality gives us

$$I(\mathbf{W}_1; \hat{\mathbf{W}}_1) \geq nR_1(1-\epsilon) - H(\epsilon). \tag{A.8}$$

Combining (A.7) and (A.8), we have

$$R_1(1-\epsilon) - \frac{H(\epsilon)}{n} \leq t_{s_1} p_1(d_1, r). \tag{A.9}$$

Letting $\epsilon \to 0$, (A.9) implies (4.6) for the case of $i = 1$. With symmetric arguments, we can derive (4.6) for $i = 2$.

We prove (4.8) by similar techniques as used in [16, 48]. Specifically, we create a new network from the original network by adding an auxiliary pipe that sends all information available at d_2 directly to d_1. Later we will show that even with the additional information, the achievable rates R_1 and R_2 are still upper bounded by (4.8). As a result, the achievable R_1 and R_2 for the original network must satisfy (4.8) as well. (4.7) is a symmetric version of (4.8).

With the additional information at d_1, the decoding function (see (2.4)) at d_1 for the new network becomes

$$\hat{\mathbf{W}}_1 = f_{d_1}([\mathbf{Y}_{\{s_1,s_2,r\}\to\{d_1,d_2\}}, \mathbf{Z}]_1^n). \tag{A.10}$$

For any $t \in [n]$, define

$$U(t) \triangleq (\mathbf{W}_2, [\mathbf{Y}_{\{s_1,s_2,r\}\to\{d_1,d_2\}}, \mathbf{Z}]_1^{t-1}). \tag{A.11}$$

We then have

$$
\begin{aligned}
nR_1 &= H(\mathbf{W}_1|\mathbf{W}_2) \\
&\le I(\mathbf{W}_1;\hat{\mathbf{W}}_1|\mathbf{W}_2) + n\epsilon_1 && \text{(A.12)} \\
&\le I(\mathbf{W}_1;[\mathbf{Y}_{\{s_1,s_2,r\}\to\{d_1,d_2\}},\mathbf{Z}]_1^n|\mathbf{W}_2) + n\epsilon_1 && \text{(A.13)} \\
&= I(\mathbf{W}_1;[\mathbf{Z}]_1^n|\mathbf{W}_2) \\
&\quad + I(\mathbf{W}_1;[\mathbf{Y}_{\{s_1,s_2,r\}\to\{d_1,d_2\}}]_1^n|\mathbf{W}_2,[\mathbf{Z}]_1^n) + n\epsilon_1 && \text{(A.14)} \\
&= \sum_{t=1}^{n} I(\mathbf{W}_1;\mathbf{Y}_{\{s_1,s_2,r\}\to\{d_1,d_2\}}(t) \\
&\qquad\quad |\mathbf{W}_2,[\mathbf{Z}]_1^n,[\mathbf{Y}_{\{s_1,s_2,r\}\to\{d_1,d_2\}}]_1^{t-1}) + n\epsilon_1 && \text{(A.15)} \\
&= n\epsilon_1 + \sum_{t=1}^{n} \big(I(\mathbf{W}_1;\mathbf{Y}_{r\to\{d_1,d_2\}}(t)|U(t),[\mathbf{Z}]_1^n) \\
&\quad + I(\mathbf{W}_1;\mathbf{Y}_{s_1\to\{d_1,d_2\}}(t)|U(t),\mathbf{Y}_{r\to\{d_1,d_2\}}(t),[\mathbf{Z}]_1^n) \\
&\quad + I(\mathbf{W}_1;\mathbf{Y}_{s_2\to\{d_1,d_2\}}(t) \\
&\qquad\quad |U(t),\mathbf{Y}_{\{r,s_1\}\to\{d_1,d_2\}}(t),[\mathbf{Z}]_1^n)\big) && \text{(A.16)} \\
&\le n\epsilon_1 + \left(\sum_{t=1}^{n} I(\mathbf{W}_1;\mathbf{Y}_{r\to\{d_1,d_2\}}(t)|U(t),[\mathbf{Z}]_1^n) \right) \\
&\quad + nt_{s_1}p_1(d_1,d_2) + 0 && \text{(A.17)} \\
&\le n\epsilon_1 + nt_{s_1}p_1(d_1,d_2) \\
&\quad + \sum_{t=1}^{n} I(X_r(t);\mathbf{Y}_{r\to\{d_1,d_2\}}(t)|U(t),[\mathbf{Z}]_1^n), && \text{(A.18)}
\end{aligned}
$$

where (A.12) follows from Fano's inequality where ϵ_1 goes to 0 when $\epsilon \to 0$; (A.13) follows from the data processing inequality and (A.10); (A.14), (A.15), and (A.16) follow from the chain rule and the fact that the distribution of $\mathbf{Z}$ is independent of $\mathbf{W}_1$ and $\mathbf{W}_2$; (A.17) follows from the observation that the second term of the summation can be upper bounded by Wald's lemma (similar to (A.7)) and $\mathbf{Y}_{s_2\to\{d_1,d_2\}}(t)$ is independent of $\mathbf{W}_1$ given $\mathbf{Z}$ (similar to (A.4)); and (A.18) follows from the data processing inequality.

To continue, we define the time sharing random variable $Q_t \in \{1, 2, ..., n\}$ with $\mathsf{Prob}(Q_t = i) = \frac{1}{n}$ for all $i \in \{1, 2, ..., n\}$ and Q_t being independent of $[\mathbf{Z}]_1^n$, $\mathbf{W}_1$, and $\mathbf{W}_2$. Since the mutual information is always non-negative, we can rewrite (A.18) as

$$
\begin{aligned}
&(R_1 - t_{s_1} p_1(d_1, d_2) - \epsilon_1)^+ \\
&\leq \sum_{t=1}^{n} \frac{1}{n} I(X_r(t); \mathbf{Y}_{r\to\{d_1,d_2\}}(t) | U(t), [\mathbf{Z}]_1^n) \\
&\leq \sum_{t=1}^{n} \frac{1}{n} H(\mathbf{Y}_{r\to\{d_1,d_2\}}(t) | U(t), [\mathbf{Z}]_1^n) \qquad \text{(A.19)} \\
&= \sum_{q_t=1}^{n} \mathsf{Prob}(Q_t = q_t) \cdot H(\mathbf{Y}_{r\to\{d_1,d_2\}}(q_t) | U(q_t), [\mathbf{Z}]_1^{q_t}, Q_t = q_t) \qquad \text{(A.20)}
\end{aligned}
$$

where (A.19) follows from the definition of the mutual information; (A.20) follows from replacing the time index t by the time sharing random variable Q_t and the distribution of $U(q_t)$ and $\mathbf{Y}_{r\to\{d_1,d_2\}}(q_t)$ does not depend on the future channel realization $[\mathbf{Z}]_{q_t+1}^n$.

We define three binary random variables $\Theta_\sigma \triangleq 1_{\{\sigma(Q_t)=r\}}$, $\Theta_{Z_1} \triangleq 1_{\{Z_{r\to d_1}(Q_t)=1\}}$, and $\Theta_{Z_2} \triangleq 1_{\{Z_{r\to d_2}(Q_t)=1\}}$, which are functions of Q_t and $[\mathbf{Z}]_1^{Q_t}$. Then we can rewrite (A.20) as the following.

$$\begin{aligned}
&(R_1 - t_{s_1} p_1(d_1, d_2) - \epsilon_1)^+ \\
&\leq \sum_{q_t=1}^{n} \frac{1}{n} H(\mathbf{Y}_{r\to\{d_1,d_2\}}(q_t) | U(q_t), [\mathbf{Z}]_1^{q_t}, Q_t = q_t, \Theta_\sigma, \Theta_{Z_1}, \Theta_{Z_2}) \qquad \text{(A.21)} \\
&= \sum_{q_t=1}^{n} \frac{1}{n} \sum_{\substack{\forall u, [z]_1^{q_t}, \\ \theta_\sigma, \theta_{Z_1}, \theta_{Z_2}}} p_{U(q_t), [\mathbf{Z}]_1^{q_t}, \Theta_\sigma, \Theta_{Z_1}, \Theta_{Z_2}}(u, [z]_1^{q_t}, \theta_\sigma, \theta_{Z_1}, \theta_{Z_2}) \\
&\quad \cdot H(\mathbf{Y}_{r\to\{d_1,d_2\}}(q_t) | U(q_t) = u, [\mathbf{Z}]_1^{q_t} = [z]_1^{q_t}, \\
&\qquad\qquad \Theta_\sigma = \theta_\sigma, \Theta_{Z_1} = \theta_{Z_1}, \Theta_{Z_2} = \theta_{Z_2}) \qquad \text{(A.22)} \\
&= \sum_{q_t=1}^{n} \frac{1}{n} \sum_{\substack{\forall u, [z]_1^{q_t}, \theta_{Z_1}, \theta_{Z_2} \\ \text{s.t. } \max\{\theta_{Z_1}, \theta_{Z_2}\} = 1}} p(u, [z]_1^{q_t}, 1, \theta_{Z_1}, \theta_{Z_2}) \\
&\quad \cdot H(X_r(q_t) | U(q_t) = u, [\mathbf{Z}]_1^{q_t} = [z]_1^{q_t}, \\
&\qquad\qquad \Theta_\sigma = 1, \Theta_{Z_1} = \theta_{Z_1}, \Theta_{Z_2} = \theta_{Z_2}) \qquad \text{(A.23)}
\end{aligned}$$

where (A.21) follows from the fact that Θ's are functions of Q and $[\mathbf{Z}]_1^{Q_t}$; (A.22) follows from the definition of the conditional entropy; and (A.23) follows from the fact that $Y_{r\to\{d_1,d_2\}}(q_t)$ is not erasure only if $\sigma(q_t) = r$ and at least one of $Z_{r\to d_1}$ and $Z_{r\to d_2}$ equals to one and furthermore $Y_{r\to\{d_1,d_2\}}(q_t) = X_r(q_t)$ under such a condition, where we use $p(u, [z]_1^{q_t}, 1, \theta_{Z_1}, \theta_{Z_2})$ as the shorthand of $p_{U(q_t), [\mathbf{Z}]_1^{q_t}, \Theta_\sigma, \Theta_{Z_1}, \Theta_{Z_2}}(u, [z]_1^{q_t}, 1, \theta_{Z_1}, \theta_{Z_2})$.

We can further simplify (A.23) by the following steps. We first note that conditioning on $U(q_t) = u$, $[\mathbf{Z}]_1^{q_t-1} = [z]_1^{q_t-1}$, and $\Theta_\sigma = 1$, the random variable $X_r(q_t)$ is independent of $\mathbf{Z}(q_t)$, Θ_{Z_1}, and Θ_{Z_2}. Notice that $[\mathbf{Z}]_1^{q_t-1}$ is a subset of $U(q_t)$. Therefore, we have

$$\begin{aligned}
&H(X_r(q_t) | U(q_t) = u, [\mathbf{Z}]_1^{q_t} = [z]_1^{q_t}, \Theta_\sigma = 1, \Theta_{Z_1} = \theta_{Z_1}, \\
&\qquad\qquad \Theta_{Z_2} = \theta_{Z_2}) \\
&= H(X_r(q_t) | U(q_t) = u, \Theta_\sigma = 1). \qquad \text{(A.24)}
\end{aligned}$$

Also the joint probability can be rewritten as

$$\sum_{\substack{\forall u,[z]_1^{q_t},\theta_{Z_1},\theta_{Z_2} \\ \text{s.t. } \max\{\theta_{Z_1},\theta_{Z_2}\}=1}} p_{U(q_t),[\mathbf{Z}]_1^{q_t},\Theta_\sigma,\Theta_{Z_1},\Theta_{Z_2}}(u,[z]_1^{q_t},1,\theta_{Z_1},\theta_{Z_2})$$

$$= \sum_{\forall u} p_{U(q_t),\Theta_\sigma}(u,1) \cdot \sum_{\substack{\forall z,\theta_{Z_1},\theta_{Z_2} \\ \text{s.t. } \max\{\theta_{Z_1},\theta_{Z_2}\}=1}}$$

$$p_{\mathbf{Z}(q_t),\Theta_{Z_1},\Theta_{Z_2}|U(q_t),\Theta_\sigma}(z,\theta_{Z_1},\theta_{Z_2}|u,1) \tag{A.25}$$

$$= \left(\sum_{\forall u} p_{U(q_t),\Theta_\sigma}(u,1)\right) \cdot p_r(d_1,d_2). \tag{A.26}$$

where (A.25) follows from the basic probability definition, and (A.26) follows from that the assumption that the channel is memoryless.

(A.24) and (A.26) helps us rewrite (A.23) as

$$(A.23) = t_r \cdot p_r(d_1,d_2) \cdot \frac{\sum_{q_t=1}^{n} \frac{1}{n} \sum_{\forall u} p(u,1) \cdot H(X_r(q_t)|u,1)}{t_r} \tag{A.27}$$

where $p(u,1)$ and $H(X_r(q_t)|u,1)$ are the shorthand for $p_{U(q_t),\Theta_\sigma}(u,1)$ and $H(X_r(q_t)|U(q_t) = u,\Theta_\sigma = 1)$, respectively.

We now focus on flow 2. By Fano's inequality, for some $\epsilon_2 > 0$ that goes to 0 as $\epsilon \to 0$, with similar steps as in (A.12)–(A.18), we can also show that

$$\begin{aligned}
&nR_2 = H(\mathbf{W}_2)\\
\leq& I(\mathbf{W}_2; [\mathbf{Y}_{\{s_1,s_2,r\}\to d_2}, \mathbf{Z}]_1^n) + n\epsilon_2\\
=& I(\mathbf{W}_2; [\mathbf{Z}]_1^n) + I(\mathbf{W}_2; [\mathbf{Y}_{\{s_1,s_2,r\}\to d_2}]_1^n | [\mathbf{Z}]_1^n) + n\epsilon_2 && \text{(A.28)}\\
=& \sum_{t=1}^{n} I(\mathbf{W}_2; \mathbf{Y}_{\{s_1,s_2,r\}\to d_2}(t) | [\mathbf{Y}_{\{s_1,s_2,r\}\to d_2}]_1^{t-1}, [\mathbf{Z}]_1^n)\\
&\quad + n\epsilon_2 && \text{(A.29)}\\
=& n\epsilon_2 + \sum_{t=1}^{n} \Big(I(\mathbf{W}_2; Y_{r\to d_2}(t) | [\mathbf{Y}_{\{s_1,s_2,r\}\to d_2}]_1^{t-1}, [\mathbf{Z}]_1^n)\\
&+ I(\mathbf{W}_2; Y_{s_2\to d_2}(t) | [\mathbf{Y}_{\{s_1,s_2\}\to d_2}]_1^{t-1}, [\mathbf{Y}_{r\to d_2}]_1^{t}, [\mathbf{Z}]_1^n)\\
&+ I(\mathbf{W}_2; Y_{s_1\to d_2}(t) | [\mathbf{Y}_{s_1\to d_2}]_1^{t-1}, [\mathbf{Y}_{\{s_2,r\}\to d_2}]_1^{t}, [\mathbf{Z}]_1^n)\Big) && \text{(A.30)}\\
\leq& n\epsilon_2 + \sum_{t=1}^{n} I(\mathbf{W}_2; Y_{r\to d_2}(t) | [\mathbf{Y}_{\{s_1,s_2,r\}\to d_2}]_1^{t-1}, [\mathbf{Z}]_1^n)\\
&+ nt_{s_2} p_2(d_2) + 0 && \text{(A.31)}
\end{aligned}$$

where (A.28), (A.29), and (A.30) follows from the chain rule and the independence between $\mathbf{W}_2$ and $[\mathbf{Z}]_1^n$; and (A.31) follows from similar derivation as in (A.17). We then have

$$
\begin{aligned}
(A.31) = & n\epsilon_2 + nt_{s_2}p_2(d_2) \\
& + \sum_{t=1}^{n} \Big(H(Y_{r\to d_2}(t)|[\mathbf{Y}_{\{s_1,s_2,r\}\to d_2}]_1^{t-1}, [\mathbf{Z}]_1^n) \\
& - H(Y_{r\to d_2}(t)|\mathbf{W}_2, [\mathbf{Y}_{\{s_1,s_2,r\}\to d_2}]_1^{t-1}, [\mathbf{Z}]_1^n)\Big) \qquad (A.32) \\
\leq & n\epsilon_2 + nt_{s_2}p_2(d_2) \\
& + \sum_{t=1}^{n} \left(H(Y_{r\to d_2}(t)|[\mathbf{Z}]_1^n) - H(Y_{r\to d_2}(t)|U(t), [\mathbf{Z}]_1^n)\right) \qquad (A.33) \\
= & n\epsilon_2 + nt_{s_2}p_2(d_2) + \sum_{t=1}^{n} I(U(t); Y_{r\to d_2}(t)|[\mathbf{Z}]_1^n), \qquad (A.34)
\end{aligned}
$$

where (A.32) and (A.34) follows from the definition of the mutual information; and (A.33) follows from the fact that *conditioning does not increase the entropy* and $[\mathbf{Y}_{\{s_1,s_2,r\}\to d_2}]_1^{t-1}$ a subset of $U(t)$. Since the mutual information is always non-negative, we now have

$$
\begin{aligned}
& (R_2 - t_{s_2}p_{s_2}(d_2) - \epsilon_2)^+ \\
\leq & \frac{1}{n}\sum_{t=1}^{n} I(U(t); Y_{r\to d_2}(t)|[\sigma, \mathbf{Z}]_1^n) \\
= & \sum_{q_t=1}^{n} \mathsf{Prob}(Q_t = q_t) \cdot I(U(q_t); Y_{r\to d_2}(q_t)|[\mathbf{Z}]_1^{q_t}, Q_t = q_t) \qquad (A.35) \\
= & \sum_{q_t=1}^{n} \frac{1}{n} \cdot H(Y_{r\to d_2}(q_t)|[\mathbf{Z}]_1^{q_t}, Q_t = q_t) \\
& - \sum_{q_t=1}^{n} \frac{1}{n} \cdot H(Y_{r\to d_2}(q_t)|U(q_t), [\mathbf{Z}]_1^{q_t}, Q_t = q_t), \qquad (A.36)
\end{aligned}
$$

where (A.35) follows from the definition of the conditional mutual information and the fact that the distribution of $U(q_t)$ and $Y_{r\to d_2}(q_t)$ does not depend on the future channel realization $[\mathbf{Z}]_{q_t+1}^{n}$; and (A.36) follows from the definition of the mutual information. We first discuss the first summation in (A.36)

$$\begin{aligned}
&\sum_{q_t=1}^{n} \frac{1}{n} \cdot H(Y_{r\to d_2}(q_t)|[\mathbf{Z}]_1^{q_t}, Q_t = q_t) \\
&= \sum_{q_t=1}^{n} \frac{1}{n} \cdot H(Y_{r\to d_2}(q_t)|[\mathbf{Z}]_1^{q_t}, Q_t = q_t, \Theta_\sigma, \Theta_{Z_2}) \qquad \text{(A.37)} \\
&= \sum_{q_t=1}^{n} \frac{1}{n} \sum_{\substack{\forall [z]_1^{q_t}, \\ \theta_\sigma, \theta_{Z_2}}} p_{[\mathbf{Z}]_1^{q_t}, \Theta_\sigma, \Theta_{Z_2}}([z]_1^{q_t}, \theta_\sigma, \theta_{Z_2}) \\
&\quad \cdot H(Y_{r\to d_2}(q_t)|[\mathbf{Z}]_1^{q_t} = [z]_1^{q_t}, \\
&\qquad\qquad \Theta_\sigma = \theta_\sigma, \Theta_{Z_2} = \theta_{Z_2}) \qquad \text{(A.38)} \\
&= \sum_{q_t=1}^{n} \frac{1}{n} \sum_{\forall [z]_1^{q_t}} p_{[\mathbf{Z}]_1^{q_t}, \Theta_\sigma, \Theta_{Z_2}}([z]_1^{q_t}, 1, 1) \\
&\quad \cdot H(X_r(q_t)|[\mathbf{Z}]_1^{q_t} = [z]_1^{q_t}, \\
&\qquad\qquad \Theta_\sigma = 1, \Theta_{Z_2} = 1) \qquad \text{(A.39)}
\end{aligned}$$

where (A.37) follows from the fact that Θ's are functions of Q and $[\mathbf{Z}]_1^{Q_t}$; (A.38) follows from the definition of the conditional entropy; and (A.39) follows from the fact that $Y_{r\to d_2}(q_t)$ is not erasure only if $\sigma(q_t) = r$ and $Z_{r\to d_2}$ equals to one and furthermore $Y_{r\to d_2}(q_t) = X_r(q_t)$ under such a condition.

We can further simplify (A.39) by the following steps. We first note that conditioning on $[\mathbf{Z}]_1^{q_t-1} = [z]_1^{q_t-1}$ and $\Theta_\sigma = 1$, the random variable $X_r(q_t)$ is independent of $\mathbf{Z}(q_t)$ and Θ_{Z_2}. Therefore, we have

$$\begin{aligned}
&H(X_r(q_t)|[\mathbf{Z}]_1^{q_t} = [z]_1^{q_t}, \Theta_\sigma = 1, \Theta_{Z_2} = 1) \\
&= H(X_r(q_t)|[\mathbf{Z}]_1^{q_t-1} = [z]_1^{q_t-1}, \Theta_\sigma = 1). \qquad \text{(A.40)}
\end{aligned}$$

Also the joint probability can be rewritten as

$$
\begin{aligned}
&\sum_{\forall [z]_1^{q_t}} p_{[\mathbf{Z}]_1^{q_t},\Theta_\sigma,\Theta_{Z_2}}([z]_1^{q_t},1,1) \\
&= \sum_{\forall [z]_1^{q_t-1}} p_{[\mathbf{Z}]_1^{q_t-1},\Theta_\sigma}([z]_1^{q_t-1},1) \cdot \sum_{\forall z} \\
&\quad p_{\mathbf{Z}(q_t),\Theta_{Z_2}|[\mathbf{Z}]_1^{q_t-1},\Theta_\sigma}(z,1|[z]_1^{q_t-1},1) \qquad \text{(A.41)} \\
&= \left(\sum_{\forall [z]_1^{q_t-1}} p_{[\mathbf{Z}]_1^{q_t-1},\Theta_\sigma}([z]_1^{q_t-1},1) \right) \cdot p_r(d_2). \qquad \text{(A.42)}
\end{aligned}
$$

where (A.41) follows from the basic probability definition, and (A.42) follows from that the assumption that the channel is memoryless.

(A.40) and (A.42) helps us rewrite (A.39) as

$$
\begin{aligned}
&(A.39) = t_r \cdot p_r(d_2) \\
&\cdot \frac{\sum_{q_t=1}^{n} \frac{1}{n} \sum_{\forall [z]_1^{q_t-1}} p([z]_1^{q_t-1},1) \cdot H(X_r(q_t)|[z]_1^{q_t-1},1)}{t_r} \qquad \text{(A.43)}
\end{aligned}
$$

where $p([z]_1^{q_t-1},1)$ and $H(X_r(q_t)|[z]_1^{q_t-1},1)$ are the shorthand for $p_{[\mathbf{Z}]_1^{q_t-1},\Theta_\sigma}([z]_1^{q_t-1},1)$ and $H(X_r(q_t)|[\mathbf{Z}]_1^{q_t-1}=[z]_1^{q_t-1},\Theta_\sigma=1)$, respectively.

Similarly, for the second summation in (A.36),

$$
\begin{aligned}
&\sum_{q_t=1}^{n}\frac{1}{n}\cdot H(Y_{r\to d_2}(q_t)|U(q_t),[\mathbf{Z}]_1^{q_t},Q_t=q_t) \\
&=\sum_{q_t=1}^{n}\frac{1}{n}\cdot H(Y_{r\to d_2}(q_t)|U(q_t),[\mathbf{Z}]_1^{q_t},Q_t=q_t,\Theta_\sigma,\Theta_{Z_2}) && \text{(A.44)}\\
&=\sum_{q_t=1}^{n}\frac{1}{n}\sum_{\substack{\forall u,[z]_1^{q_t},\\ \theta_\sigma,\theta_{Z_2}}} p_{U(q_t),[\mathbf{Z}]_1^{q_t},\Theta_\sigma,\Theta_{Z_2}}(u,[z]_1^{q_t},\theta_\sigma,\theta_{Z_2})\\
&\quad\cdot H(Y_{r\to d_2}(q_t)|U(q_t)=u,[\mathbf{Z}]_1^{q_t}=[z]_1^{q_t},\\
&\qquad\qquad \Theta_\sigma=\theta_\sigma,\Theta_{Z_2}=\theta_{Z_2}) && \text{(A.45)}\\
&=\sum_{q_t=1}^{n}\frac{1}{n}\sum_{\forall u,[z]_1^{q_t}} p_{U(q_t),[\mathbf{Z}]_1^{q_t},\Theta_\sigma,\Theta_{Z_2}}(u,[z]_1^{q_t},1,1)\\
&\quad\cdot H(X_r(q_t)|U(q_t)=u,[\mathbf{Z}]_1^{q_t}=[z]_1^{q_t},\\
&\qquad\qquad \Theta_\sigma=1,\Theta_{Z_2}=1) && \text{(A.46)}
\end{aligned}
$$

where (A.44) follows from the fact that Θ's are functions of Q and $[\mathbf{Z}]_1^{Q_t}$; (A.45) follows from the definition of the conditional entropy; and (A.46) follows from the fact that $Y_{r\to d_2}(q_t)$ is not erasure only if $\sigma(q_t)=r$ and $Z_{r\to d_2}$ equals to one and furthermore $Y_{r\to d_2}(q_t)=X_r(q_t)$ under such a condition.

We can further simplify (A.46) by the following steps. We first note that conditioning on $U(q_t)=u$, $[\mathbf{Z}]_1^{q_t-1}=[z]_1^{q_t-1}$, and $\Theta_\sigma=1$, the random variable $X_r(q_t)$ is independent of $\mathbf{Z}(q_t)$ and Θ_{Z_2}. Notice that $[\mathbf{Z}]_1^{q_t-1}$ is a subset of $U(q_t)$. Therefore, we have

$$
\begin{aligned}
&H(X_r(q_t)|U(q_t)=u,[\mathbf{Z}]_1^{q_t}=[z]_1^{q_t},\Theta_\sigma=1,\Theta_{Z_2}=1)\\
&=H(X_r(q_t)|U(q_t)=u,\Theta_\sigma=1). && \text{(A.47)}
\end{aligned}
$$

Also the joint probability can be rewritten as

$$\begin{aligned}
&\sum_{\forall u,[z]_1^{q_t}} p_{U(q_t),[\mathbf{Z}]_1^{q_t},\Theta_\sigma,\Theta_{Z_2}}(u,[z]_1^{q_t},1,1) \\
&= \sum_{\forall u} p_{U(q_t),\Theta_\sigma}(u,1)\cdot\sum_{\forall z} \\
&\quad p_{\mathbf{Z}(q_t),\Theta_{Z_2}|U(q_t),\Theta_\sigma}(z,1|u,1) && \text{(A.48)} \\
&= \left(\sum_{\forall u} p_{U(q_t),\Theta_\sigma}(u,1)\right)\cdot p_r(d_2). && \text{(A.49)}
\end{aligned}$$

where (A.48) follows from the basic probability definition, and (A.49) follows from that the assumption that the channel is memoryless.

(A.47) and (A.49) helps us rewrite (A.46) as

$$\begin{aligned}
&(A.39) = t_r\cdot p_r(d_2) \\
&\cdot\frac{\sum_{q_t=1}^{n}\frac{1}{n}\sum_{\forall u} p(u,1)\cdot H(X_r(q_t)|u,1)}{t_r} && \text{(A.50)}
\end{aligned}$$

where $p(u,1)$ and $H(X_r(q_t)|u,1)$ are the shorthand for $p_{U(q_t),\Theta_\sigma}(u,1)$ and $H(X_r(q_t)|U(q_t)=u,\Theta_\sigma=1)$, respectively.

Combining (A.43) and (A.50), we can rewrite (A.36) in the following form.

$$\begin{aligned}
&(R_2 - t_{s_2}p_{s_2}(d_2)-\epsilon_2)^+ \\
&\le t_r\cdot p_r(d_2) \\
&\cdot\left(\frac{\sum_{q_t=1}^{n}\frac{1}{n}\sum_{\forall [z]_1^{q_t-1}} p([z]_1^{q_t-1},1)\cdot H(X_r(q_t)|[z]_1^{q_t-1},1)}{t_r}-\right. \\
&\left.\frac{\sum_{q_t=1}^{n}\frac{1}{n}\sum_{\forall u} p(u,1)\cdot H(X_r(q)|u,1)}{t_r}\right). && \text{(A.51)}
\end{aligned}$$

Summing up $\frac{(A.27)}{p_r(d_1,d_2)}$ and $\frac{(A.51)}{p_r(d_2)}$, we thus have

$$\begin{aligned}
&\frac{(R_1 - t_{s_1} p_1(d_1,d_2) - \epsilon_1)^+}{p_r(d_1,d_2)} + \frac{(R_2 - t_{s_2} p_2(d_2) - \epsilon_2)^+}{p_r(d_2)} \\
&\leq t_r \cdot \frac{\sum_{q_t=1}^{n} \frac{1}{n} \sum_{\forall [z]_1^{q_t-1}} p([z]_1^{q_t-1},1) \cdot H(X_r(q_t)|[z]_1^{q_t-1},1)}{t_r} \quad \text{(A.52)} \\
&\leq t_r, \quad \text{(A.53)}
\end{aligned}$$

where (A.53) is based on the following observations. We first note that by definition

$$\begin{aligned}
t_r &= \sum_{q_t=1}^{n} \frac{1}{n} \mathsf{Prob}(\sigma(q_t) = r) \\
&= \sum_{q_t=1}^{n} \frac{1}{n} \sum_{\forall [z]_1^{q_t-1}} p([z]_1^{q_t-1},1).
\end{aligned}$$

Therefore, the fraction term in (A.52) can be viewed as the normalization of the conditional entropy $H(X_r(q_t)|[z]_1^{q_t-1},1)$. Since each conditional entropy is no larger than 1 (with the base of the logarithm being q), we thus have (A.53).

(A.53) holds for arbitrary $\epsilon > 0$. Letting $\epsilon \to 0$[1], we thus have the following final inequality.

$$\frac{(R_1 - t_{s_1} p_1(d_1,d_2))^+}{p_r(d_1,d_2)} + \frac{(R_2 - t_{s_2} p_2(d_2))^+}{p_r(d_2)} \leq t_r,$$

which gives us (4.8). (4.7) can be proven by symmetry. The proof of the outer bound is thus complete.

[1]As a result, $\epsilon_1 \to 0$ and $\epsilon_2 \to 0$.

B. DETAILED ACHIEVABILITY ANALYSIS OF SBLNC

The feasibility for Policy $\Gamma_{s_1,0}$ and Policy $\Gamma_{s_1,1}$ has been proven in Section 4.3.1. In the following discussion about the rank of spaces, we again rely on the first order, expectation-based analysis and assume the application of the law of large numbers implicitly.

Policy $\Gamma_{s_1,2}$: Similar to the analysis for Policy $\Gamma_{s_1,1}$, assuming $q \geq 2$, the condition that (3.3) being non-empty is equivalent to whether the following rank-based inequality is satisfied.

$$\begin{aligned}&\mathrm{Rank}(S_2) - \mathrm{Rank}(S_2 \cap (S_1 \oplus S_r))\\ =&\mathrm{Rank}(S_1 \oplus S_2 \oplus S_r) - \mathrm{Rank}(S_1 \oplus S_r) > 0.\end{aligned} \tag{B.1}$$

where (B.1) follows from Lemma 3.1.2.

Similar to the discussion in $\Gamma_{s_1,0}$ and $\Gamma_{s_1,1}$, we will quantify individual ranks at the end of $\Gamma_{s_1,2}$, the policy of interest, and prove that even in the end of $\Gamma_{s_1,2}$, the rank difference in (B.1) is strictly larger than 0. Therefore, throughout the entire duration of $\Gamma_{s_1,2}$, (B.1) is larger than 0 and $\Gamma_{s_1,2}$ is always feasible.

We first focus on $\mathrm{Rank}(S_1 \oplus S_2 \oplus S_r)$. Since $S_1 \oplus S_2 \oplus S_r$ is a subset of the exclusion set in $\Gamma_{s_1,0}$, every time a $\Gamma_{s_1,0}$ packet is received by one of d_1, d_2, and r, $\mathrm{Rank}(S_1 \oplus S_2 \oplus S_r)$ will increase by one. On the other hand, notice that $S_1 \oplus S_2 \oplus S_r$ is a superset of the inclusion set in $\Gamma_{s_1,1}$ and $\Gamma_{s_1,2}$. Hence $\mathrm{Rank}(S_1 \oplus S_2 \oplus S_r)$ remains the same throughout $\Gamma_{s_1,1}$ and $\Gamma_{s_1,2}$. As a result, in the end of policy $\Gamma_{s_1,2}$, we have

$$\mathsf{E}\{\mathrm{Rank}(S_1 \oplus S_2 \oplus S_r)\} = n\omega_{s_1}^0 p_1(d_1, d_2, r). \tag{B.2}$$

We now focus on $\mathrm{Rank}(S_1 \oplus S_r)$. Since $S_1 \oplus S_r$ is a subset of the exclusion sets of $\Gamma_{s_1,0}$, $\Gamma_{s_1,1}$ and $\Gamma_{s_1,2}$, every time a packet of $\Gamma_{s_1,0}$, $\Gamma_{s_1,1}$, or $\Gamma_{s_1,2}$ is received by one of d_1 and r, $\mathrm{Rank}(S_1 \oplus S_r)$ will increase by one. As a result, in the end of policy $\Gamma_{s_1,2}$, we have

$$\mathsf{E}\{\mathrm{Rank}(S_1 \oplus S_r)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1 + \omega_{s_1}^2)p_1(d_1, r). \tag{B.3}$$

Jointly, (B.2), (B.3), and (4.15) imply (B.1) in the end of $\Gamma_{s_1,2}$.

Policy $\Gamma_{s_1,3}$: Similar to the analysis of the previous policies, assuming $q \geq 2$, the condition that (3.4) being non-empty is equivalent to whether the following rank-based inequality is satisfied.

$$\begin{aligned}
&\mathrm{Rank}(S_r) - \mathrm{Rank}(((S_2 \cap S_r) \oplus S_1) \cap S_r) \\
=&\mathrm{Rank}((S_2 \cap S_r) \oplus S_1 \oplus S_r) - \mathrm{Rank}((S_2 \cap S_r) \oplus S_1) && \text{(B.4)} \\
=&\mathrm{Rank}(S_1 \oplus S_r) - (\mathrm{Rank}(S_1) + \mathrm{Rank}(S_2 \cap S_r) \\
&- \mathrm{Rank}(S_1 \cap S_2 \cap S_r)) && \text{(B.5)} \\
=&\mathrm{Rank}(S_1 \oplus S_r) - \mathrm{Rank}(S_1) - (\mathrm{Rank}(S_2) + \mathrm{Rank}(S_r) \\
&- \mathrm{Rank}(S_2 \oplus S_r)) + \mathrm{Rank}(S_1 \cap S_2 \cap S_r) > 0, && \text{(B.6)}
\end{aligned}$$

where (B.4) follows from Lemma 3.1.2; (B.5) follows from simple set operations and from Lemma 3.1.2; and (B.6) follows from Lemma 3.1.2.

Similar to the previous discussion, we will quantify individual ranks at the end of $\Gamma_{s_1,3}$, the policy of interest and prove that even in the end of $\Gamma_{s_1,3}$, the rank difference in (B.6) is strictly larger than 0. Therefore, throughout the entire duration of $\Gamma_{s_1,3}$, (B.6) is larger than 0 and $\Gamma_{s_1,3}$ is always feasible.

By similar analysis,[1] in the end of $\Gamma_{s_1,3}$ we have

$$\mathsf{E}\{\mathrm{Rank}(S_1)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1 + \omega_{s_1}^2 + \omega_{s_1}^3)p_1(d_1), \tag{B.7}$$

$$\mathsf{E}\{\mathrm{Rank}(S_2)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1 + \omega_{s_1}^3)p_1(d_2), \tag{B.8}$$

$$\mathsf{E}\{\mathrm{Rank}(S_r)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1 + \omega_{s_1}^2)p_1(r), \tag{B.9}$$

$$\mathsf{E}\{\mathrm{Rank}(S_1 \oplus S_r)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1 + \omega_{s_1}^2)p_1(d_1, r), \tag{B.10}$$

$$\mathsf{E}\{\mathrm{Rank}(S_2 \oplus S_r)\} = n\omega_{s_1}^0 p_1(d_2, r) \tag{B.11}$$

What remains to be decided is the value of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ at the end of Policy $\Gamma_{s_1,3}$. To proceed, we introduce an auxiliary node a in the following way. Whenever a vector $\mathbf{v}$ sent by s_1 is received by both d_1 and r, we let the auxiliary node a observe such $\mathbf{v}$ as well. The knowledge space of a, denoted by S_a is thus the linear span of all vectors received by both d_1 and r.

We first argue that $S_a = S_1 \cap S_r$ in the end of policy $\Gamma_{s_1,2}$. Since a only observes those vectors commonly available at both d_1 and r, the knowledge space of S_a is a subset of $S_1 \cap S_r$. Knowing $S_a \subseteq S_1 \cap S_r$, we can quickly check that S_a is a subset of the exclusion sets in Policies $\Gamma_{s_1,0}$, $\Gamma_{s_1,1}$, and $\Gamma_{s_1,2}$. Therefore, every time node a receives a packet during policies $\Gamma_{s_1,0}$, $\Gamma_{s_1,1}$, and $\Gamma_{s_1,2}$, the rank of S_a will increase by one. Therefore, we have

$$\mathsf{E}\{\mathrm{Rank}(S_a)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1 + \omega_{s_1}^2)p_1(d_1 r) \tag{B.12}$$

in the end of $\Gamma_{s_1,2}$. On the other hand, by similar analysis as before, we have

$$\mathsf{E}\{\mathrm{Rank}(S_1)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1 + \omega_{s_1}^2)p_1(d_1),$$

$$\mathsf{E}\{\mathrm{Rank}(S_r)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1 + \omega_{s_1}^2)p_1(r),$$

$$\mathsf{E}\{\mathrm{Rank}(S_1 \oplus S_r)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1 + \omega_{s_1}^2)p_1(d_1, r)$$

[1]The derivation of (B.8) for the case of Policy $\Gamma_{s_1,3}$ uses the following inequality as well.

$$(3.4) \subseteq (S_r \backslash (S_2 \cap S_r)) = (S_r \backslash S_2).$$

in the end of policy $\Gamma_{s_1,2}$. By Lemma 3.1.2, we thus have $\mathrm{Rank}(S_a) = \mathrm{Rank}(S_1 \cap S_r)$. As a result, we have proven $S_a = (S_1 \cap S_r)$ in the end of $\Gamma_{s_1,2}$.

By the above analysis, we thus have $(S_1 \cap S_2 \cap S_r) = S_a \cap S_2$. By similarly rank-based analysis, in the end of $\Gamma_{s_1,2}$ we have

$$\mathsf{E}\{\mathrm{Rank}(S_2)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1)p_1(d_2) \tag{B.13}$$

$$\mathsf{E}\{\mathrm{Rank}(S_2 \oplus S_a)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1)p_1(d_2, d_1 r) \tag{B.14}$$

where $p_1(d_2, d_1 r)$ in (B.14) is the probability that at least one of node d_2 and node a receives the packet and (B.14) follows from the observation that $S_2 \oplus S_a$ is a subset of the exclusion sets of $\Gamma_{s_1,0}$, $\Gamma_{s_1,1}$ and is a superset of the inclusion set of $\Gamma_{s_1,2}$. By (B.12), (B.13), and (B.14), we have thus proven that

$$\begin{aligned} &\mathsf{E}\{\mathrm{Rank}(S_1 \cap S_2 \cap S_r)\} = \mathsf{E}\{\mathrm{Rank}(S_a \cap S_2)\} \\ =&\mathsf{E}\{\mathrm{Rank}(S_2)\} + \mathsf{E}\{\mathrm{Rank}(S_a)\} - \mathsf{E}\{\mathrm{Rank}(S_2 \oplus S_a)\} \\ =&n\left(\omega_{s_1}^0 + \omega_{s_1}^1\right) p_1(d_1 d_2 r) + n\omega_{s_1}^2 p_1(d_1 r) \end{aligned} \tag{B.15}$$

in the end of $\Gamma_{s_1,2}$.

In the following, we will quantify the increment of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ during $\Gamma_{s_1,3}$. To that end, we introduce two more auxiliary nodes b and c. In the beginning of $\Gamma_{s_1,3}$, we let node b (resp. c) be aware of the knowledge space $S_1 \cap S_r$ (resp. $S_2 \cap S_r$). During $\Gamma_{s_1,3}$, whenever a packet is received by d_1 (resp. d_2), we let the auxiliary node b (resp. c) observe such a packet as well. From the construction, it is clear that the following equalities hold in the beginning of $\Gamma_{s_1,3}$.

$$S_b = S_1 \cap S_r \tag{B.16}$$

$$S_c = S_2 \cap S_r. \tag{B.17}$$

We will prove that (B.16) and (B.17) hold even in the end of $\Gamma_{s_1,3}$ as well.

In the following, we will prove that (B.16) holds in the end of $\Gamma_{s_1,3}$. We first note that by our construction, we always have $S_1 \supset S_b \supset (S_1 \cap S_r)$. Knowing that S_b is always a subset of S_1 and S_1 is a subset of the exclusion sets in $\Gamma_{s_1,3}$, we can see that everytime d_1 receives a packet during policy $\Gamma_{s_1,3}$, $\mathrm{Rank}(S_b)$ will increase by one. Moreover, only when d_1 receives a packet during policy $\Gamma_{s_1,3}$ will $\mathrm{Rank}(S_b)$ increase. As a result, the increment of $\mathrm{Rank}(S_b)$ during $\Gamma_{s_1,3}$ equals the number of times d_1 receives a packet during $\Gamma_{s_1,3}$. On the other hand, $\mathrm{Rank}(S_1 \cap S_r) = \mathrm{Rank}(S_1) + \mathrm{Rank}(S_r) - \mathrm{Rank}(S_1 \oplus S_r)$. Since both S_r and $S_1 \oplus S_r$ are supersets of the inclusion set of $\Gamma_{s_1,3}$, both $\mathrm{Rank}(S_r)$ and $\mathrm{Rank}(S_1 \oplus S_r)$ remain identical during $\Gamma_{s_1,3}$. Therefore, the increment of $\mathrm{Rank}(S_1 \cap S_r)$ is identical to the increment of $\mathrm{Rank}(S_1)$ during $\Gamma_{s_1,3}$. As a result, the increment of $\mathrm{Rank}(S_1 \cap S_r)$ during $\Gamma_{s_1,3}$ equals the number of times d_1 receives a packet during $\Gamma_{s_1,3}$. We have thus proven $\mathrm{Rank}(S_b) = \mathrm{Rank}(S_1 \cap S_r)$ in the end of $\Gamma_{s_1,3}$, which implies (B.16). (B.17) can be proven by symmetry.

To quantify the increment of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ during $\Gamma_{s_1,3}$, we notice that $\mathrm{Rank}(S_1 \cap S_2 \cap S_r) = \mathrm{Rank}(S_b \cap S_c) = \mathrm{Rank}(S_b) + \mathrm{Rank}(S_c) - \mathrm{Rank}(S_b \oplus S_c)$. As a result, the increment of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ during policy $\Gamma_{s_1,3}$ is the summation of the increments of $\mathrm{Rank}(S_b)$ and $\mathrm{Rank}(S_c)$ minus the increment of $\mathrm{Rank}(S_b \oplus S_c)$ during $\Gamma_{s_1,3}$. By our construction, the increments of S_b, S_c, and $S_b \oplus S_c$ during $\Gamma_{s_1,3}$ is simply $n\omega_{s_1}^3 p_1(d_1)$, $n\omega_{s_1}^3 p_1(d_2)$, and $n\omega_{s_1}^3 p_1(d_1, d_2)$, respectively. As a result, the increment of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ during $\Gamma_{s_1,3}$ is simply $n\omega_{s_1}^3 p_1(d_1 d_2)$.

Combining (B.15), we have thus proven that

$$\begin{aligned} &\mathsf{E}\{\mathrm{Rank}(S_1 \cap S_2 \cap S_r)\} \\ =&n\left(\omega_{s_1}^0 + \omega_{s_1}^1\right) p_1(d_1 d_2 r) + n\omega_{s_1}^2 p_1(d_1 r) + n\omega_{s_1}^3 p_1(d_1 d_2) \end{aligned} \tag{B.18}$$

in the end of Policy $\Gamma_{s_1,3}$.

Jointly, (B.7) to (B.11), (B.18), and (4.16) imply (B.6) in the end of $\Gamma_{s_1,3}$.

Policy $\Gamma_{s_1,4}$: Similar to the analysis of the previous policies, the condition that (3.5) being non-empty is equivalent to whether the following rank-based inequality is satisfied in the end of $\Gamma_{s_1,4}$.

$$\begin{aligned}
&\mathrm{Rank}(S_2 \cap S_r) - \mathrm{Rank}(S_1 \cap S_2 \cap S_2) \\
=&(\mathrm{Rank}(S_2) + \mathrm{Rank}(S_r) - \mathrm{Rank}(S_2 \oplus S_r)) \\
&- \mathrm{Rank}(S_1 \cap S_2 \cap S_r) > 0.
\end{aligned} \tag{B.19}$$

Similar to the previous discussion, we will quantify individual ranks at the end of $\Gamma_{s_1,4}$ and prove that (B.19) holds in the end of $\Gamma_{s_1,4}$.

By similar analysis, we have

$$\mathsf{E}\{\mathrm{Rank}(S_2)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1 + \omega_{s_1}^3)p_1(d_2), \tag{B.20}$$

$$\mathsf{E}\{\mathrm{Rank}(S_r)\} = n(\omega_{s_1}^0 + \omega_{s_1}^1 + \omega_{s_1}^2)p_1(r), \tag{B.21}$$

$$\mathsf{E}\{\mathrm{Rank}(S_2 \oplus S_r)\} = n\omega_{s_1}^0 p_1(d_2, r) \tag{B.22}$$

in the end of $\Gamma_{s_1,4}$. What remains to be decided is the value of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ at the end of Policy $\Gamma_{s_1,4}$. In (B.18), we have already quantified $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ in the end of $\Gamma_{s_1,3}$. In the following, we will quantify the increment of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ during $\Gamma_{s_1,4}$. By (3.5), we can see that every time d_1 receives a packet during $\Gamma_{s_1,4}$, $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ will increase by one. As a result, the increment of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ during $\Gamma_{s_1,4}$ is $n\omega_{s_1}^4 p_1(d_1)$. Together with (B.18), we have proven that

$$\begin{aligned}
&\mathsf{E}\{\mathrm{Rank}(S_1 \cap S_2 \cap S_r)\} \\
=&n\left(\omega_{s_1}^0 + \omega_{s_1}^1\right) p_1(d_1 d_2 r) + n\omega_{s_1}^2 p_1(d_1 r) \\
&+ n\omega_{s_1}^3 p_1(d_1 d_2) + n\omega_{s_1}^4 p_1(d_1)
\end{aligned} \tag{B.23}$$

in the end of $\Gamma_{s_1,4}$. Jointly, (B.20) to (B.23) and (4.17) imply that (B.19) holds in the end of $\Gamma_{s_1,4}$.

The feasibility of policy $\Gamma_{s_2,k}$, $k=0,1,2,3,4$, can be proven by symmetry.

Policy $\Gamma_{r,1}$: We first notice that the inclusion space and exclusion space of Policy $\Gamma_{r,1}$ are the same as of Policy $\Gamma_{s_1,3}$. Hence to prove the feasibility of Policy $\Gamma_{r,1}$, we need to prove that (B.6) holds in the end of $\Gamma_{r,1}$. By similar analysis, we have

$$\begin{aligned}\mathsf{E}\{\mathrm{Rank}(S_1)\} =& n(\omega_{s_1}^0+\omega_{s_1}^1+\omega_{s_1}^2+\omega_{s_1}^3+\omega_{s_1}^4)p_1(d_1)\\ &+n\omega_{r,N}^1 p_r(d_1),\end{aligned} \tag{B.24}$$

$$\begin{aligned}\mathsf{E}\{\mathrm{Rank}(S_2)\} =& n(\omega_{s_1}^0+\omega_{s_1}^1+\omega_{s_1}^3)p_1(d_2)\\ &+n\omega_{r,\mathrm{N}}^1 p_r(d_2),\end{aligned} \tag{B.25}$$

$$\mathsf{E}\{\mathrm{Rank}(S_r)\} = n(\omega_{s_1}^0+\omega_{s_1}^1+\omega_{s_1}^2)p_1(r), \tag{B.26}$$

$$\mathsf{E}\{\mathrm{Rank}(S_1\oplus S_r)\} = n(\omega_{s_1}^0+\omega_{s_1}^1+\omega_{s_1}^2)p_1(d_1,r), \tag{B.27}$$

$$\mathsf{E}\{\mathrm{Rank}(S_2\oplus S_r)\} = n\omega_{s_1}^0 p_1(d_2,r). \tag{B.28}$$

in the end of $\Gamma_{r,1}$.

What remains to be decided is the value of $\mathrm{Rank}(S_1\cap S_2\cap S_r)$ at the end of Policy $\Gamma_{r,1}$. In (B.23) we have computed the value of $\mathrm{Rank}(S_1\cap S_2\cap S_r)$ in the end of $\Gamma_{s_1,4}$. As a result, we only need to quantify the increment of $\mathrm{Rank}(S_1\cap S_2\cap S_r)$ during $\Gamma_{r,1}$. By the same analysis as when we quantify the increment of $\mathrm{Rank}(S_1\cap S_2\cap S_r)$ during $\Gamma_{s_1,3}$, the increment of $\mathrm{Rank}(S_1\cap S_2\cap S_r)$ during $\Gamma_{r,1}$ is $n\omega_{r,\mathrm{N}}^1 p_r(d_1d_2)$. By (B.23), we have shown that

$$\begin{aligned}&\mathsf{E}\{\mathrm{Rank}(S_1\cap S_2\cap S_r)\}\\ =&n\left(\omega_{s_1}^0+\omega_{s_1}^1\right)p_1(d_1d_2r)+n\omega_{s_1}^2 p_1(d_1r)\\ &+n\omega_{s_1}^3 p_1(d_1d_2)+n\omega_{s_1}^4 p_1(d_1)+n\omega_{r,\mathrm{N}}^1 p_r(d_1d_2)\end{aligned} \tag{B.29}$$

in the end of $\Gamma_{r,1}$. Jointly, (B.24) to (B.29) and (4.18) imply that (B.6) holds in the end of $\Gamma_{r,1}$.

The discussion of Policy $\Gamma_{r,2}$ follows symmetrically.

Policy $\Gamma_{r,3}$ for $\mathbf{v}^{(1)}$: We will prove that for the first $n\omega^1_{r,\text{C}}$ time slots of Policy $\Gamma_{r,3}$, we can always choose $\mathbf{v}^{(1)}$ according to (3.13). To that end, we first notice that the inclusion space and exclusion space in (3.13) are the same as those of Policy $\Gamma_{s_1,4}$. Hence to prove that (3.13) remains non-empty during the first $n\omega^1_{r,\text{C}}$ time slots of Policy $\Gamma_{r,3}$, we need to prove that (B.19) holds in the end of the first $n\omega^1_{r,\text{C}}$ time slots of Policy $\Gamma_{r,3}$. By similar analysis as used in the previous policies, we have

$$\begin{aligned} \mathsf{E}\{\mathrm{Rank}(S_2)\} &= n(\omega^0_{s_1} + \omega^1_{s_1} + \omega^3_{s_1})p_1(d_2) \\ &\quad + n\omega^1_{r,\text{N}}p_r(d_2), \qquad \text{(B.30)} \\ \mathsf{E}\{\mathrm{Rank}(S_r)\} &= n(\omega^0_{s_1} + \omega^1_{s_1} + \omega^2_{s_1})p_1(r), \qquad \text{(B.31)} \\ \mathsf{E}\{\mathrm{Rank}(S_2 \oplus S_r)\} &= n\omega^0_{s_1}p_1(d_2, r). \qquad \text{(B.32)} \end{aligned}$$

in the end of the first $n\omega^1_{r,\text{C}}$ time slots of Policy $\Gamma_{r,3}$. What remains to be decided is the value of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ at the end of the first $n\omega^1_{r,\text{C}}$ time slots of Policy $\Gamma_{r,3}$. In (B.29) we have computed the value of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ in the end of $\Gamma_{r,1}$. As a result, we only need to quantify the increment of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ during the first $n\omega^1_{r,\text{C}}$ time slots of Policy $\Gamma_{r,3}$. By the same analysis as when we quantify the increment of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ during $\Gamma_{s_1,4}$, the increment of $\mathrm{Rank}(S_1 \cap S_2 \cap S_r)$ during the first $n\omega^1_{r,\text{C}}$ time slots of Policy $\Gamma_{r,3}$ is $n\omega^1_{r,\text{C}}p_r(d_1)$. By (B.29), we have shown that

$$\begin{aligned} &\mathsf{E}\{\mathrm{Rank}(S_1 \cap S_2 \cap S_r)\} \\ =&n\left(\omega^0_{s_1} + \omega^1_{s_1}\right)p_1(d_1d_2r) + n\omega^2_{s_1}p_1(d_1r) \\ &+ n\omega^3_{s_1}p_1(d_1d_2) + n\omega^4_{s_1}p_1(d_1) + n\omega^1_{r,\text{N}}p_r(d_1d_2) \\ &+ n\omega^1_{r,\text{C}}p_r(d_1) \qquad \text{(B.33)} \end{aligned}$$

in the end of the first $n\omega^1_{r,\text{C}}$ time slots of Policy $\Gamma_{r,3}$. Jointly, (B.30) to (B.33) and (4.19) imply that (B.19) holds in the end of the first $n\omega^1_{r,\text{C}}$ time slots of $\Gamma_{r,3}$.

The discussion of the first $n\omega^2_{r,\text{C}}$ time slots of $\Gamma_{r,3}$ follows symmetrically.

The above analysis completes the achievability proof started in Section 4.3.1.

C. BOUND-MATCHING OF WIRELESS BUTTERFLY NETWORKS

Here we are going to verify the proposed parameter assignment in the proof of *Lemma 4.3.1* satisfies (4.9), (4.10), (4.32) to (4.39). For (4.9),

$$\begin{aligned}
\sum_{k=0}^{3} \omega_{s_i}^k =& \frac{R_i}{p_i(d_j, r)} \\
& + R_i \left(\min\left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) \\
& + R_i \left(\frac{1}{p_i(r)} - \frac{1}{p_i(d_j)} \right)^+ \\
& + \min\left\{ R_i \left(\frac{1}{p_i(d_j)} - \frac{1}{p_i(r)} \right)^+, t_{s_i} - \frac{R_i}{p_i(r)} \right\} \\
\leq & \frac{R_i}{p_i(r)} \\
& + \min\left\{ R_i \left(\frac{1}{p_i(d_j)} - \frac{1}{p_i(r)} \right)^+, t_{s_i} - \frac{R_i}{p_i(r)} \right\} \\
\leq & t_r.
\end{aligned}$$

Hence it satisfies (4.9). For (4.10),

$$\begin{aligned}
& \omega_{r,\mathrm{N}}^i + \omega_{r,\mathrm{N}}^j + \omega_{r,\mathrm{C}}^i \\
=& \frac{(R_i - t_{s_i} p_i(d_j))^+}{p_r(d_i, d_j)} + \frac{(R_j - t_{s_j} p_{s_j}(d_i))^+}{p_r(d_i, d_j)} \\
& + \frac{R_i}{p_r(d_i)} - \frac{(R_i - t_{s_i} p_i(d_j))^+}{p_r(d_i, d_j)} \\
=& \frac{R_i}{p_r(d_i)} + \frac{(R_j - t_{s_j} p_{s_j}(d_i))^+}{p_r(d_i, d_j)} \leq t_r.
\end{aligned}$$

Hence it satisfies (4.10). For (4.32),

$$\omega_{s_i}^0 p_i(d_j, r) = \frac{R_i}{p_i(d_j, r)} p_i(d_j, r) = R_i.$$

Hence it satisfies (4.32). For (4.33),

$$\begin{aligned}
\text{LHS} =& R_i \left(\min\left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) p_i(r) \\
=& R_i \left(\min\left\{ 1, \frac{p_i(r)}{p_i(d_j)} \right\} - \frac{p_i(r)}{p_i(d_j, r)} \right) \\
\leq& R_i - R_i \frac{p_i(r)}{p_i(d_j, r)}, \\
\text{RHS} =& R_i \frac{p_i(d_j \overline{r})}{p_i(d_j, r)} = R_i - R_i \frac{p_i(r)}{p_i(d_j, r)}.
\end{aligned}$$

Hence it satisfies (4.33). For (4.34),

$$\begin{aligned}
\text{LHS} =& R_i \left(\min\left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) p_i(d_j) \\
=& R_i \left(\min\left\{ \frac{p_i(d_j)}{p_i(r)}, 1 \right\} - \frac{p_i(d_j)}{p_i(d_j, r)} \right) \\
\leq& R_i - R_i \frac{p_i(d_j)}{p_i(d_j, r)}, \\
\text{RHS} =& R_i \frac{p_i(r \overline{d_j})}{p_i(d_j, r)} = R_i - R_i \frac{p_i(d_j)}{p_i(d_j, r)}.
\end{aligned}$$

Hence it satisfies (4.34). For (4.35),

$$\begin{aligned}
\text{LHS} =& R_i \left(\frac{1}{p_i(r)} - \frac{1}{p_i(d_j)} \right)^+ p_i(r) = R_i \left(1 - \frac{p_i(r)}{p_i(d_j)} \right)^+, \\
\text{RHS} =& R_i \frac{p_i(d_j \overline{r})}{p_i(d_j, r)} \\
& - R_i \left(\min \left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) p_i(r) \\
=& R_i \left(1 - \frac{p_i(r)}{p_i(d_j, r)} \right) \\
& - R_i \left(\min \left\{ 1, \frac{p_i(r)}{p_i(d_j)} \right\} - \frac{p_i(r)}{p_i(d_j, r)} \right) \\
\geq& R_i \left(1 - \frac{p_i(r)}{p_i(d_j)} \right)^+.
\end{aligned}$$

Hence it satisfies (4.35). For (4.36),

$$\begin{aligned}
\text{LHS} =& \min \left\{ R_i \left(1 - \frac{p_i(d_j)}{p_i(r)} \right)^+, t_{s_i} p_i(d_j) - R_i \frac{p_i(d_j)}{p_i(r)} \right\}, \\
\text{RHS} =& R_i \frac{p_i(r \overline{d_j})}{p_i(d_j, r)} \\
& - R_i \left(\min \left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) p_i(d_j) \\
=& R_i \left(1 - \frac{p_i(d_j)}{p_i(d_j, r)} \right) \\
& - R_i \left(\min \left\{ \frac{p_i(d_j)}{p_i(r)}, 1 \right\} - \frac{p_i(d_j)}{p_i(d_j, r)} \right) \\
\geq& R_i \left(1 - \frac{p_i(d_j)}{p_i(r)} \right)^+.
\end{aligned}$$

Hence it satisfies (4.36). For (4.37),

$$
\begin{aligned}
\text{LHS} =& \left(R_i - t_{s_i} p_i(d_j)\right)^+ , \\
\text{RHS} =& R_i \frac{p_i(r\overline{d_j})}{p_i(d_j, r)} \\
& - R_i \left(\min\left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) p_i(d_j) \\
& - \min\left\{ R_i \left(1 - \frac{p_i(d_j)}{p_i(r)}\right)^+ , t_{s_i} p_i(d_j) - R_i \frac{p_i(d_j)}{p_i(r)} \right\} \\
\geq & R_i \left(1 - \frac{p_i(d_j)}{p_i(r)}\right)^+ \\
& - \min\left\{ R_i \left(1 - \frac{p_i(d_j)}{p_i(r)}\right)^+ , t_{s_i} p_i(d_j) - R_i \frac{p_i(d_j)}{p_i(r)} \right\} \\
\geq & \left(R_i - t_{s_i} p_i(d_j)\right)^+ .
\end{aligned}
$$

Hence it satisfies (4.37). For (4.38),

$$
\begin{aligned}
\text{LHS} =& R_i - (R_i - t_{s_i} p_i(d_j))^+ \frac{p_r(d_i)}{p_r(d_i, d_j)}, \\
\text{RHS} =& R_i \frac{p_i(d_j r)}{p_i(d_j, r)} \\
& + (p_i(d_j) + p_i(r)) \\
& \cdot R_i \left(\min \left\{ \frac{1}{p_i(r)}, \frac{1}{p_i(d_j)} \right\} - \frac{1}{p_i(d_j, r)} \right) \\
& + R_i \left(1 - \frac{p_i(r)}{p_i(d_j)} \right)^+ \\
& + \min \left\{ R_i \left(1 - \frac{p_i(d_j)}{p_i(r)} \right)^+, t_{s_i} p_i(d_j) - R_i \frac{p_i(d_j)}{p_i(r)} \right\} \\
& + \frac{(R_i - t_{s_i} p_i(d_j))^+}{p_r(d_i, d_j)} (p_r(d_i, d_j) - p_r(d_i)) \\
=& R_i \min \left\{ \frac{p_i(d_j)}{p_i(r)}, \frac{p_i(r)}{p_i(d_j)} \right\} + R_i \left(1 - \frac{p_i(r)}{p_i(d_j)} \right)^+ \\
& + \min \left\{ R_i \left(1 - \frac{p_i(d_j)}{p_i(r)} \right)^+, t_{s_i} p_i(d_j) - R_i \frac{p_i(d_j)}{p_i(r)} \right\} \\
& + (R_i - t_{s_i} p_i(d_j))^+ (1 - \frac{p_r(d_i)}{p_r(d_i, d_j)}) \\
\geq& R_i - (R_i - t_{s_i} p_i(d_j))^+ \frac{p_r(d_i)}{p_r(d_i, d_j)}.
\end{aligned}
$$

Hence it satisfies (4.38). For (4.39),

$$p_r(d_i)(\omega^i_{r,\mathrm{N}} + \omega^i_{r,\mathrm{C}}) = R_i$$

With the above verification, we conclude that this proposed assignment satisfies (4.9), (4.10), (4.32) to (4.39).

D. THE UPPER BOUND OF THE DIFFERENCE BETWEEN THE ACTUAL QUEUE AND INTERMEDIATE ACTUAL QUEUE

Recall that we assume the SPN under consideration is acyclic, and hence we could arrange the queues from the upstream to the downstream and index them from 1 (the most upstream) to K (the most dowsntream). Recall that we use $n(t)$ to denote the preferred SA chosen by the back-pressure scheduler. The proof of Lemma 6.2.1 consists of proving the following lemmas.

We first prove the following lemma.

Lemma D.0.1 *For any $k = 1, 2, ..., K$ and any time t,*

$$\begin{aligned} &|Q_k(t) - Q_k^{\textit{inter}}(t)| \\ &\leq \sum_{\tau=1}^{t-1} I(\exists k' \in \mathcal{I}_{n(\tau)} : Q_{k'}(\tau) = 0). \end{aligned} \tag{D.1}$$

Lemma D.0.2 *For any $k = 1, 2, ..., K$ and any time τ,*

$$\begin{aligned} &I(k \in \mathcal{I}_{n(\tau)}) \\ \leq &I(k \in \mathcal{I}_{n(\tau)}) I(Q_k^{\textit{inter}}(\tau) < \beta^{\textit{in}}_{k,n(\tau)}(\mathsf{cq}(\tau))) \\ &+ I(k \in \mathcal{I}_{n(\tau)}) I(Q_k(\tau) = 0) \\ &\cdot I(\beta^{\textit{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)) = 0) I(1 \leq Q_k^{\textit{inter}}(\tau)) \\ &+ I(k \in \mathcal{I}_{n(\tau)}) I(Q_k(\tau) = 0) \\ &\cdot I(\beta^{\textit{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)) = 0) I(0 \leq Q_k^{\textit{inter}}(\tau) < 1) \\ &+ I(k \in \mathcal{I}_{n(\tau)}) I(Q_k(\tau) = 0) \\ &\cdot I(\beta^{\textit{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)) = 1) I(1 \leq Q_k^{\textit{inter}}(\tau)). \end{aligned} \tag{D.2}$$

By combining Lemma D.0.1 and Lemma D.0.2 and by the union bound, we can upper bound the value of $|Q_k(t) - Q_k^{\text{inter}}(t)|$ as follows.

$$
\begin{aligned}
&|Q_k(t) - Q_k^{\text{inter}}(t)| \\
\leq &\sum_{k'=1}^{K} \sum_{\tau=1}^{t-1} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\text{inter}}(\tau) < \beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))) \\
&+ \sum_{k'=1}^{K} \sum_{\tau=1}^{t-1} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}(\tau) = 0) \\
&\cdot I(\beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 0) I(1 \leq Q_{k'}^{\text{inter}}(\tau)) \\
&+ \sum_{k'=1}^{K} \sum_{\tau=1}^{t-1} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}(\tau) = 0) \\
&\cdot I(\beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 0) I(0 \leq Q_{k'}^{\text{inter}}(\tau) < 1) \\
&+ \sum_{k'=1}^{K} \sum_{\tau=1}^{t-1} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}(\tau) = 0) \\
&\cdot I(\beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 1) I(1 \leq Q_{k'}^{\text{inter}}(\tau)).
\end{aligned} \tag{D.3}
$$

Recall that the goal of Lemma 6.2.1 is to upper bound the expectation of $|Q_k(t) - Q_k^{\text{inter}}(t)|$ as a weighted sum of $\mathsf{E}\{N_{\mathsf{NA},k'}(t)\}$. We then observe that the expectation of the first term of the RHS of (D.3) is indeed the sum of $\mathsf{E}\{N_{\mathsf{NA},k'}(t)\}$. Therefore, to complete the proof of Lemma 6.2.1, we only need to upper bound the expectation of the second to the fourth terms of the RHS of (D.3) by some weighted sum of $\mathsf{E}\{N_{\mathsf{NA},k'}(t)\}$. The following Lemmas D.0.3 to D.0.5 upper bound the expectation of the second to the fourth terms, respectively.

Lemma D.0.3 *For any $k = 1, ..., K$, there exists a constant γ_k such that*

$$\begin{aligned}
&\mathsf{E}(\sum_{\tau=1}^{t} I(k \in \mathcal{I}_{n(\tau)})I(Q_k(\tau) = 0) \\
&\cdot I(\beta^{in}_{k,n(\tau)}(\mathsf{cq}(\tau)) = 0)I(1 \leq Q^{inter}_k(\tau))) \\
&\leq \gamma_k \mathsf{E}(\sum_{\tau=1}^{t} I(k \in \mathcal{I}_{n(\tau)})I(Q_k(\tau) = 0) \\
&\cdot I(\beta^{in}_{k,n(\tau)}(\mathsf{cq}(\tau)) = 1)I(1 \leq Q^{inter}_k(\tau))
\end{aligned} \tag{D.4}$$

for all $t = 1$ to ∞.

Namely, the expectation of the second term of the RHS of (D.3) is upper bounded by γ_k times the expectation of the fourth term of the RHS of (D.3).

Lemma D.0.4 *For any $k = 1, ..., K$, there exists a constant γ_k such that*

$$\begin{aligned}
&\mathsf{E}(\sum_{\tau=1}^{t} I(k \in \mathcal{I}_{n(\tau)})I(Q_k(\tau) = 0) \\
&\cdot I(\beta^{in}_{k,n(\tau)}(\mathsf{cq}(\tau)) = 0)I(0 \leq Q^{inter}_k(\tau) < 1)) \\
&\leq \gamma_k \mathsf{E}(\sum_{\tau=1}^{t} I(k \in \mathcal{I}_{n(\tau)})I(Q^{inter}_k(\tau) < \beta^{in}_{k,n(\tau)}(\mathsf{cq}(\tau)))
\end{aligned}$$

for all $t = 1$ to ∞.

Lemma D.0.5 upper bounds the expectation of the fourth term of (D.3), which is also used to upper bound the second term of (D.3) through Lemma D.0.3.

Lemma D.0.5 *For any k value, there exists γ_1 to γ_{k-1} such that*

$$\begin{aligned}
&\mathsf{E}(\sum_{\tau=1}^{t} I(k \in \mathcal{I}_{n(\tau)})I(Q^{inter}_k(\tau) \geq 1) \\
&\cdot I(\beta^{in}_{k,n(\tau)}(\mathsf{cq}(\tau)) = 1)I(Q_k(\tau) = 0)) \\
&\leq \sum_{k'=1}^{k-1} \gamma_{k'} \mathsf{E}(\sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)})I(Q^{inter}_{k'}(\tau) < \beta^{in}_{k',n(\tau)}(\mathsf{cq}(\tau))))
\end{aligned} \tag{D.5}$$

for all $t = 1$ *to* ∞.

Finally by applying Lemma D.0.3 and Lemma D.0.5 to the second term of the RHS of (D.3), applying Lemma D.0.4 to the third term of the RHS of (D.3), and applying Lemma D.0.5 to the fourth term of the RHS of (D.3), we have proven the following statement: for any k, there exist γ_1 to γ_K such that

$$\begin{aligned}&\mathsf{E}(|Q_k(t) - Q_k^{\text{inter}}(t)|)\\&\leq \sum_{k'=1}^{K} \gamma_{k'} \mathsf{E}(\sum_{\tau=1}^{t-1} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\text{inter}}(\tau) < \beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))))\end{aligned}$$

for all $t = 1$ to ∞. The proof of Lemma D.0.1 to Lemma D.0.5 are relegated to Appedix E. Lemma 6.2.1 is thus proven. ■

E. PROOFS OF FOUR LEMMAS

Proof of Lemma D.0.1: Before proving Lemma D.0.1, we first rewrite the LHS of (D.1) as

$$\begin{aligned}&|Q_k(t) - Q_k^{\text{inter}}(t)|\\ &= \sum_{\tau=1}^{t-1} \left(|Q_k(\tau+1) - Q_k^{\text{inter}}(\tau+1)| - |Q_k(\tau) - Q_k^{\text{inter}}(\tau)| \right).\end{aligned}$$

So to prove (D.1), it suffices to show that the following inequality holds for all k and $\tau < t$.

$$\begin{aligned}&\left(|Q_k(\tau+1) - Q_k^{\text{inter}}(\tau+1)| - |Q_k(\tau) - Q_k^{\text{inter}}(\tau)| \right)\\ &\leq I(\exists k' \in \mathcal{I}_{n(\tau)} : Q_{k'}(\tau) = 0).\end{aligned} \tag{E.1}$$

We now prove (E.1). By set relationship, one can easily verify that one and only one of the following 3 possible cases is true at each time τ.

1. $k \notin \mathcal{I}_{n(\tau)} \cup \mathcal{O}_{n(\tau)}$.
2. $k \in \mathcal{I}_{n(\tau)} \cup \mathcal{O}_{n(\tau)}$ and SA $n(\tau)$ is feasible.
3. $k \in \mathcal{I}_{n(\tau)} \cup \mathcal{O}_{n(\tau)}$ and SA $n(\tau)$ is not feasible.

In the case of 1), the LHS of (E.1) at time τ is zero since $Q_k(\tau+1) - Q_k(\tau) = Q_k^{\text{inter}}(\tau+1) - Q_k^{\text{inter}}(\tau) = \sum_{m=1}^{M} \alpha_{k,m} a_m(\tau)$. Inequality (E.1) thus holds obviously.

In the case of 2), the scheduled SA $n(\tau)$ is feasible. Suppose $k \in \mathcal{O}_{n(\tau)}$. Then the LHS of (E.1) at time τ is always 0 since $Q_k(\tau+1) - Q_k(\tau) = Q_k^{\text{inter}}(\tau+1) - Q_k^{\text{inter}}(\tau) = \beta_{k,n(\tau)}^{\text{out}}(\mathsf{cq}(\tau)) + \sum_{m=1}^{M} \alpha_{k,m} a_m(\tau)$. Suppose $k \in \mathcal{I}_{n(\tau)}$. Then $Q_k(\tau+1) = Q_k(\tau) - \beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) + \sum_{m=1}^{M} \alpha_{k,m} a_m(\tau)$. There are now two sub-cases: $Q_k^{\text{inter}}(\tau) \geq Q_k(\tau) \geq 1$ or $Q_k^{\text{inter}}(\tau) < Q_k(\tau)$. (The case that $Q_k(\tau) = 0$ is not possible since we now consider the scenario SA $n(\tau)$ is feasible.) In the first sub-case, since $Q_k^{\text{inter}}(\tau) \geq Q_k(\tau) \geq 1$ and since SA $n(\tau)$ is feasible, we must have $Q_k^{\text{inter}}(\tau+1) - Q_k^{\text{inter}}(\tau) = Q_k(\tau+1) - Q_k(\tau) =$

$-\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)) + \sum_{m=1}^{M} \alpha_{k,m} a_m(\tau)$. As a result, the LHS of (E.1) at time τ is again 0. In the second sub-case, $Q^{\text{inter}}_k(\tau+1) = (Q^{\text{inter}}_k(\tau) - \beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)))^+ + \sum_{m=1}^{M} \alpha_{k,m} a_m(\tau)$, and $Q_k(\tau+1) = Q^{\text{inter}}_k(\tau) - \beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)) + \sum_{m=1}^{M} \alpha_{k,m} a_m(\tau)$ since in this case we assume SA $n(\tau)$ is feasible and thus $Q_k(\tau) \geq 1$. Recall that $Q^{\text{inter}}_k(\tau) < Q_k(\tau)$ in this sub-case. Therefore, $(Q^{\text{inter}}_k(\tau) - \beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)))^+ \leq (Q_k(\tau) - \beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)))^+ = (Q_k(\tau) - \beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)))$. We thus have $Q^{\text{inter}}_k(\tau+1) \leq Q_k(\tau+1)$. As a result, the LHS of (E.1) at time τ becomes

$$
\begin{aligned}
&(|Q_k(\tau+1) - Q^{\text{inter}}_k(\tau+1)| - |Q_k(\tau) - Q^{\text{inter}}_k(\tau)|) \\
=&(Q_k(\tau+1) - Q_k(\tau) + Q^{\text{inter}}_k(\tau) - Q^{\text{inter}}_k(\tau+1)) \\
=&-\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)) \\
&+ (Q^{\text{inter}}_k(\tau) - \max\{Q^{\text{inter}}_k(\tau) - \beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)), 0\}) \\
=&-\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)) + \min\{\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau)), Q^{\text{inter}}_k(\tau)\} \\
\leq&0.
\end{aligned}
$$

Since the RHS of (E.1) is always non-negative, (E.1) holds in the case of 2).

In the case of 3), the preferred SA $n(\tau)$ is not feasible. Without loss of generality, we assume that there is no external arrival at queue k in time τ since any external arrival will change $Q_k(\tau)$ and $Q^{\text{inter}}_k(\tau)$ by the same amount. Since SA $n(\tau)$ is not feasible, we have $Q_k(\tau+1) = Q_k(\tau)$. On the other hand, $Q^{\text{inter}}_k(\tau)$ might still increase or decrease at most by 1 since the update rule of $Q^{\text{inter}}_k(\tau)$ (6.10) does not depend on whether SA $n(\tau)$ is feasible or not. Since $Q^{\text{inter}}_k(\tau)$ changes by at most 1, the LHS of (E.1) at time τ is upper bounded by 1 in this case while the RHS of (E.1) is always 1 since SA $n(\tau)$ is no feasible. Thus (E.1) holds in the case of 3).

In summary, for all k and $\tau < t$, (E.1) holds in all 3 possible cases. Lemma D.0.1 is proven. ■

Proof of Lemma D.0.2: Suppose $k \in \mathcal{I}_{n(\tau)}$. We claim that one and only one of the following 4 possible cases is true:

1. $Q_k^{\text{inter}}(\tau) < \beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))$.

2. $Q_k(\tau) = 0$, $\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 0$, and $1 \le Q_k^{\text{inter}}(\tau)$.

3. $Q_k(\tau) = 0$, $\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 0$, and $0 \le Q_k^{\text{inter}}(\tau) < 1$.

4. $Q_k(\tau) = 0$, $\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 1$, and $1 \le Q_k^{\text{inter}}(\tau)$.

The reason is as follows. For any fixed k, we either have $Q_k^{\text{inter}}(\tau) < \beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))$; or $Q_k(\tau) = 0$ and $Q_k^{\text{inter}}(\tau) \ge \beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))$. In the former scenario, we have 1). In the latter scenario, we can further partition the event based on the values of $\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))$ and $Q_k^{\text{inter}}(\tau)$ and we thus have 2) to 4). The four cases correspond to the four terms in the RHS of (D.2). The proof of Lemma D.0.2 is complete. ■

Proof of Lemma D.0.3: Obviously we have

$$\begin{aligned}
&\mathsf{E}\Big(\sum_{\tau=1}^{t} I(k \in \mathcal{I}_{n(\tau)}) I(Q_k(\tau) = 0)\\
&\cdot I(\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 0) I(1 \le Q_k^{\text{inter}}(\tau))\Big)\\
&\le \mathsf{E}\Big(\sum_{\tau=1}^{t} I(k \in \mathcal{I}_{n(\tau)}) I(Q_k(\tau) = 0)\\
&\cdot I(1 \le Q_k^{\text{inter}}(\tau))\Big).
\end{aligned}$$

We then observe that $\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))$, the channel realization from queue k to SA $n(\tau)$ during time τ, is independent of $n(\tau)$, $Q_k(\tau)$, and $Q_k^{\text{inter}}(\tau)$, which depend only on the history from time 1 to $(\tau - 1)$, not on the realization of $\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))$ in time τ.

Furthermore, recall that $\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))$ is a Bernoulli random variable with $\mathsf{E}(I(\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 1)) = \overline{\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))}$. Define

$$\gamma_k = \frac{1}{\min_{c \in \mathsf{CQ}, n \in [1,N], k \in \mathcal{I}_n} \overline{\overline{\beta_{k,n}^{\text{in}}(c)}}},$$

which always exists since we assume $\min_{c\in\mathsf{CQ},n\in[1,N],k\in\mathcal{I}_n}\overline{\beta^{\text{in}}_{k,n}(c)}>0$ in the SPN of interest (Assumption 3). Since whether $\beta^{\text{in}}_{k,n}(\mathsf{cq}(\tau))=1$ is independent of $n(\tau)$, $Q_k(\tau)$, and $Q^{\text{inter}}_k(\tau)$, we have

$$\begin{aligned}&\mathsf{E}(\sum_{\tau=1}^{t}I(k\in\mathcal{I}_{n(\tau)})I(Q_k(\tau)=0)\\&\cdot I(1\leq Q^{\text{inter}}_k(\tau)))\\&\leq\gamma_k\mathsf{E}(\sum_{\tau=1}^{t}I(k\in\mathcal{I}_{n(\tau)})I(Q_k(\tau)=0)\\&\cdot I(\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau))=1)I(1\leq Q^{\text{inter}}_k(\tau)),\end{aligned}$$

which completes the proof of Lemma D.0.3. ■

Proof of Lemma D.0.4: We have

$$\begin{aligned}&\mathsf{E}(\sum_{\tau=1}^{t}I(k\in\mathcal{I}_{n(\tau)})I(Q_k(\tau)=0)\\&\cdot I(\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau))=0)I(0\leq Q^{\text{inter}}_k(\tau)<1))\\\leq&\gamma_k\mathsf{E}(\sum_{\tau=1}^{t}I(k\in\mathcal{I}_{n(\tau)})I(Q_k(\tau)=0)\\&\cdot I(\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau))=1)I(0\leq Q^{\text{inter}}_k(\tau)<1))\qquad\text{(E.2)}\\\leq&\gamma_k\mathsf{E}(\sum_{\tau=1}^{t}I(k\in\mathcal{I}_{n(\tau)})I(Q^{\text{inter}}_k(\tau)<\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau))),\qquad\text{(E.3)}\end{aligned}$$

where (E.2) follows from the same argument as used in the proof of Lemma D.0.3; (E.3) follows from the fact that if $\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau))=1$ and $0\leq Q^{\text{inter}}_k(\tau)<1$, then $Q^{\text{inter}}_k(\tau)<\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau))$. Thus Lemma D.0.4 is proven. ■

Proof of Lemma D.0.5: Define $\Delta Q_k(\tau)\overset{\triangle}{=}Q^{\text{inter}}_k(\tau)-Q_k(\tau)$. We first state the following four claims and use these claims to prove Lemma D.0.5. The proof of these four claims are relegated to Appendix F.

Claim E.0.1 *For the most upstream queue ($k=1$) we have $Q_1(\tau)\geq Q^{\text{inter}}_1(\tau)$ for all τ.*

Claim E.0.2 *For any $k = 1, 2, ..., K$ and any time t, we have*

$$\sum_{\tau=1}^{t} I(k \in \mathcal{I}_{n(\tau)}) I(Q_k^{\textit{inter}}(\tau) \geq 1)$$
$$\cdot I(\beta_{k,n(\tau)}^{\textit{in}}(\mathsf{cq}(\tau)) = 1) I(Q_k(\tau) = 0)$$
$$\leq \sum_{\tau=1}^{t} I(k \in \mathcal{O}_{n(\tau)}) I(\Delta Q_k(\tau+1) > \Delta Q_k(\tau)). \tag{E.4}$$

Claim E.0.3 *For any $\tau = 1$ to ∞, we have*

$$I\left(\Delta Q_k(\tau+1) > \Delta Q_k(\tau)\right) I(k \in \mathcal{O}_{n(\tau)})$$
$$\leq \sum_{k'=1}^{k-1} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\textit{inter}}(\tau) < \beta_{k',n(\tau)}^{\textit{in}}(\mathsf{cq}(\tau)))$$
$$+ \sum_{k'=1}^{k-1} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}(\tau) = 0)$$
$$\cdot I(\beta_{k',n(\tau)}^{\textit{in}}(\mathsf{cq}(\tau))) = 0) I(1 \leq Q_{k'}^{\textit{inter}}(\tau))$$
$$+ \sum_{k'=1}^{k-1} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}(\tau) = 0)$$
$$\cdot I(\beta_{k',n(\tau)}^{\textit{in}}(\mathsf{cq}(\tau))) = 0) I(0 \leq Q_{k'}^{\textit{inter}}(\tau) < 1)$$
$$+ \sum_{k'=1}^{k-1} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}(\tau) = 0)$$
$$\cdot I(\beta_{k',n(\tau)}^{\textit{in}}(\mathsf{cq}(\tau))) = 1) I(1 \leq Q_{k'}^{\textit{inter}}(\tau)). \tag{E.5}$$

Claim E.0.4 *For any $k = 1, 2, ..., K$ and any time t, we have*

$$\begin{aligned}
&\sum_{\tau=1}^{t} I(k \in \mathcal{I}_{n(\tau)}) I(Q_k^{\mathit{inter}}(\tau) \geq 1) \\
&\cdot I(\beta_{k,n(\tau)}^{\mathit{in}}(\mathsf{cq}(\tau)) = 1) I(Q_k(\tau) = 0) \\
&\leq \sum_{k'=1}^{k-1} \sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_k^{\mathit{inter}}(\tau) < \beta_{k',n(\tau)}^{\mathit{in}}(\mathsf{cq}(\tau))) \\
&+ \sum_{k'=1}^{k-1} \sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\mathit{inter}}(\tau) \geq 1) \\
&\cdot I(\beta_{k',n(\tau)}^{\mathit{in}}(\mathsf{cq}(\tau)) = 1) I(Q_{k'}(\tau) = 0) \\
&+ \sum_{k'=1}^{k-1} \sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\mathit{inter}}(\tau) \geq 1) \\
&\cdot I(\beta_{k',n(\tau)}^{\mathit{in}}(\mathsf{cq}(\tau)) = 0) I(Q_{k'}(\tau) = 0) \\
&+ \sum_{k'=1}^{k-1} \sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(1 > Q_{k'}^{\mathit{inter}}(\tau) \geq 0) \\
&\cdot I(\beta_{k',n(\tau)}^{\mathit{in}}(\mathsf{cq}(\tau)) = 0) I(Q_{k'}(\tau) = 0).
\end{aligned} \tag{E.6}$$

With the above four claims, we are now ready to prove Lemma D.0.5. We prove Lemma D.0.5 by induction on the value of k. Consider the case of $k = 1$ first. By Claim E.0.1, we have $Q_1(\tau) \geq Q_1^{\text{inter}}(\tau)$ for all τ. Therefore, whenever $Q_1(\tau) = 0$, we must have $Q_1^{\text{inter}}(\tau) = 0$. As a result, the LHS of (D.5) is always 0 for $k = 1$. Lemma D.0.5 thus holds for $k = 1$.

Now consider general k. By Lemma E.0.4 and taking the expectation on both sides, we have

$$
\begin{aligned}
&\mathsf{E}(\sum_{\tau=1}^{t} I(k \in \mathcal{I}_{n(\tau)}) I(Q_k^{\text{inter}}(\tau) \geq 1) \\
&\cdot I(\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 1) I(Q_k(\tau) = 0)) \\
\leq &\sum_{k'=1}^{k-1} \mathsf{E}(\sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_k^{\text{inter}}(\tau) < \beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)))) \\
&+ \sum_{k'=1}^{k-1} \mathsf{E}(\sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\text{inter}}(\tau) \geq 1) \\
&\cdot I(\beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 1) I(Q_{k'}(\tau) = 0)) \\
&+ \sum_{k'=1}^{k-1} \mathsf{E}(\sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\text{inter}}(\tau) \geq 1) \\
&\cdot I(\beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 0) I(Q_{k'}(\tau) = 0)) \\
&+ \sum_{k'=1}^{k-1} \mathsf{E}(\sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(1 > Q_{k'}^{\text{inter}}(\tau) \geq 0) \\
&\cdot I(\beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 0) I(Q_{k'}(\tau) = 0)).
\end{aligned} \tag{E.7}
$$

We look at the second term of the RHS of (E.7) first. Notice that by induction hypothesis, for $k' = 1, ..., k-1$, there exists $\gamma_{k',1}$ to $\gamma_{k',k'-1}$ such that

$$
\begin{aligned}
&\mathsf{E}(\sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\text{inter}}(\tau) \geq 1) \\
&\cdot I(\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 1) I(Q_{k'} = 0)) \\
\leq &\sum_{k''=1}^{k'-1} \gamma_{k',k''} \\
&\cdot \mathsf{E}(\sum_{\tau=1}^{t} I(k'' \in \mathcal{I}_{n(\tau)}) I(Q_{k''}^{\text{inter}}(\tau) < \beta_{k'',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)))).
\end{aligned}
$$

The above inequality shows that the second term of the RHS of (E.7) can be bounded by a weighted sum of $\mathsf{E}(\sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\text{inter}}(\tau) < \beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))))$ for $k' = 1, ..., k-1$.

We now look at the third term of the RHS of (E.7). By Lemma D.0.3, for $k' = 1, ..., k-1$, there exists a constant $\gamma'_{k'}$ such that

$$\begin{aligned}
&\mathsf{E}(\sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\text{inter}}(\tau) \geq 1) \\
&\quad \cdot I(\beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 0) I(Q_{k'}(\tau) = 0)) \\
&\leq \gamma'_{k'} \mathsf{E}(\sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\text{inter}}(\tau) \geq 1) \\
&\quad \cdot I(\beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 1) I(Q_{k'}(\tau) = 0)).
\end{aligned}$$

Again by the same argument for the second term of the RHS of (E.7), the third term of the RHS of (E.7) can also be bounded by a weighted sum of $\mathsf{E}(\sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\text{inter}}(\tau) < \beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))))$ for $k' = 1, ..., k-1$.

By Lemma D.0.4, the fourth term of the RHS of (E.7) can also be bounded by a weighted sum of $\mathsf{E}(\sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\text{inter}}(\tau) < \beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))))$ for $k' = 1, ..., k-1$.

Since all the 4 terms in the RHS of (E.7) can be upper bounded by a weighted sum of $\mathsf{E}(\sum_{\tau=1}^{t} I(k' \in \mathcal{I}_{n(\tau)}) I(Q_{k'}^{\text{inter}}(\tau) < \beta_{k',n(\tau)}^{\text{in}}(\mathsf{cq}(\tau))))$ for $k' = 1, ..., k-1$, we have thus proven Lemma D.0.5. ■

F. PROOFS OF FOUR CLAIMS

Proof of Claim E.0.1: By the definition of $Q_k(t)$ and $Q_k^{\text{inter}}(t)$, we have $Q_k(1) = 0 = Q_k^{\text{inter}}(1)$. The desired inequality holds when $\tau = 1$. Suppose the inequality holds for some τ. We now prove that the inequality also holds for $\tau + 1$. To that end, we first notice that any external arrival at time τ will increase the $Q_1^{\text{inter}}(\tau)$ and $Q_1(\tau)$ by the same amount. Therefore, the external arrivals will not affect the order between $Q_1^{\text{inter}}(\tau)$ and $Q_1(\tau)$ and we can thus assume there is no external arrival in time t without loss of generality. Consider the first scenario in which $1 \notin \mathcal{I}_{n(\tau)}$. Since $1 \notin \mathcal{I}_{n(\tau)}$, no packets will leave queue 1. Since we have $Q_1^{\text{inter}}(\tau) \leq Q_1(\tau)$ to begin with, we will still have $Q_1^{\text{inter}}(\tau + 1) \leq Q_1(\tau + 1)$.

Now consider the scenario of $1 \in \mathcal{I}_{n(\tau)}$ and the following two cases: Case 1: $Q_1^{\text{inter}}(\tau) < \beta^{\text{in}}_{1,n(\tau)}(\mathsf{cq}(\tau))$. In this case, at the beginning of time $\tau + 1$, $Q_1^{\text{inter}}(\tau + 1) = 0$ due to the update rule (6.10). Since the actual queue length $Q_k(\tau)$ is non-negative, we must have $Q_1^{\text{inter}}(\tau + 1) \leq Q_1(\tau + 1)$. Case 2: $Q_1^{\text{inter}}(\tau) \geq \beta^{\text{in}}_{1,n(\tau)}(\mathsf{cq}(\tau))$. In this case, we have $Q_1^{\text{inter}}(\tau + 1) = Q_1^{\text{inter}}(\tau) - \beta^{\text{in}}_{1,n(\tau)}(\mathsf{cq}(\tau))$ (recall that we assume no external). We observe that the actual queue length Q_1 either decreases by $\beta^{\text{in}}_{1,n(\tau)}(\mathsf{cq}(\tau))$ or remain the same, depending on whether SA $n(\tau)$ can be carried out successfully or not (see Difference 1 in the Section 6.2.1). Therefore, the decrease amount of $Q_1^{\text{inter}}(\tau)$ is no less than the decrease amount of $Q_1(\tau)$, which together with the fact that $Q_1^{\text{inter}}(\tau) \leq Q_1(\tau)$ imply $Q_1^{\text{inter}}(\tau+1) \leq Q_1(\tau + 1)$. By induction, we have proven that $Q_1(\tau) \geq Q_1^{\text{inter}}(\tau)$ for all τ. ■

Proof of Claim E.0.2: Since both $Q_k(\tau)$ and $Q_k^{\text{inter}}(\tau)$ are integer-valued random processes, $\Delta Q_k(\tau)$ is also an integer-valued random process. Furthermore, we observe that the changes of $Q_k(\tau)$ and $Q_k^{\text{inter}}(\tau)$ is always in the same direction. Namely, if $Q_k^{\text{inter}}(\tau)$ increases,[1] then it means that k is one of the output queues of SA $n(\tau)$, which means that $Q_k(\tau)$ can either increase or remain the same (the latter is due to the fact that the preferred

[1]For ease of exposition, we do not count the external arrivals since any external arrival will increase Q_k and Q_k^{inter} by the same amount.

SA $n(\tau)$ may be infeasible). Similarly, if $Q_k^{\text{inter}}(\tau)$ decreases then $Q_k(\tau)$ can decrease or remain the same. Since the largest change of Q_k (resp. Q_k^{inter}) is at most 1 and they move in the same direction, it can be easily shown that the change of $\Delta Q_k(\tau)$ is also at most 1. To simplify the expression, for the time being, we sometimes ignore the queue index k in $\Delta Q_k(\tau)$. That is, we will write $\Delta Q_k(\tau)$ as $\Delta Q(\tau)$ in the remaining of this proof.

In the following, we iteratively define two sequences of time instants, $\{s_i : \forall i\}$ and $\{t_i : \forall i\}$. The first time instant is $s_1 = 1$. Then for any i, define $t_i \in (s_i, t+1]$ as the largest time instant such that for all time instant $\tilde{\tau} \in (s_i, t_i)$, we have $\Delta Q(\tilde{\tau}) > 0$. Note that for all $\tilde{\tau} \in (s_i, s_i + 1)$ we have $\Delta Q(\tilde{\tau}) > 0$ since $(s_i, s_i + 1)$ is an empty interval. As a result, t_i always exists and is uniquely defined as long as we have $s_i \leq t$ to begin with. Furthermore, since $\Delta Q(\tau)$ is an integer-valued random process with change at most 1 at each time slot, we can observe that $\Delta Q(t_i) = 0$ if $t_i \leq t$ and $\Delta Q(t_i) > 0$ if $t_i = t + 1$. In summary, $\Delta Q(t_i) \geq 0$.

After defining t_i, we define $s_{i+1} \in [t_i, t]$ as the time instant such that for all time instant $\tilde{\tau} \in (t_i, s_{i+1}]$, we have $\Delta Q(\tilde{\tau}) \leq 0$ and $\Delta Q(s_{i+1} + 1) > 0$. This time, such s_{i+1} may or may not exist. For example, if $\Delta Q(\tilde{\tau}) \leq 0$ for all $\tilde{\tau} \in [t_i, t + 1]$, then s_{i+1} does not exist since even the largest possible choice of $s_{i+1} = t$ still does not satisfy the requirement $\Delta Q(s_{i+1} + 1) > 0$. However, one may observe that we must have $\Delta Q(s_{i+1}) = 0$ whenever s_{i+1} exists. The reason is that $\Delta Q(\tau)$ changes by at most one in any two consecutive time slots. Therefore, the facts that $\Delta Q(\tilde{\tau}) \leq 0$ and $\Delta Q(s_{i+1} + 1) > 0$ jointly imply $\Delta Q(s_{i+1}) = 0$. In summary, $[s_i, t_i)$ is the i-th "continuous interval" such that all $\tau \in (s_i, t_i)$ satisfy $\Delta Q(\tau) > 0$.

Define M_s as the number of (s_i, t_i) pairs that do exist. Since $s_1 = 1$ is clearly defined, we have $M_s \geq 1$. We will now argue that for any $i = 1$ to M_s, we have

$$\begin{aligned}
&\sum_{\tau=s_i}^{t_i-1} I(\Delta Q(\tau+1) < \Delta Q(\tau)) \\
\leq &\sum_{\tau=s_i}^{t_i-1} I(\Delta Q(\tau+1) > \Delta Q(\tau)). \qquad \text{(F.1)}
\end{aligned}$$

To see the correctness of (F.1), we first observe that

$$\Delta Q(t_i) = \Delta Q(s_i) + \sum_{\tau=s_i}^{t_i-1} (\Delta Q(\tau+1) - \Delta Q(\tau)).$$

Since $\Delta Q(s_i) = 0$ and $\Delta Q(t_i) \geq 0$, we have

$$\begin{aligned} &\sum_{\tau=s_i}^{t_i-1} (\Delta Q(\tau+1) - \Delta Q(\tau))^+ \\ \geq &\sum_{\tau=s_i}^{t_i-1} (\Delta Q(\tau+1) - \Delta Q(\tau))^-, \end{aligned}$$

where $(v)^+ = \max\{0, v\}$ and $(v)^- = \max\{0, -v\}$. Since $\Delta Q(\tau)$ moves by at most 1, we thus have (F.1).

Now we turn our focus back to proving Claim E.0.2. We notice that when

$$\begin{aligned} &I(k \in \mathcal{I}_{n(\tau)}) I(Q_k^{\text{inter}}(\tau) \geq 1) \\ &\cdot I(\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 1) I(Q_k(\tau) = 0) = 1, \end{aligned} \tag{F.2}$$

we have $\Delta Q(\tau) = Q_k^{\text{inter}}(\tau) - Q_k(\tau) \geq 1$. Moreover, we argue that $\Delta Q(\tau+1) = \Delta Q(\tau) - 1$. The reason is as follows. Since queue k is one of the input queues of the preferred SA at τ and the queue lengths satisfy $Q_k^{\text{inter}}(\tau) \geq 1$ and $Q_k(\tau) = 0$, Q_k^{inter} will decrease by one according to the update rule (6.10) while Q_k remains zero since the preferred SA $n(\tau)$ is infeasible. As a result, $\Delta Q(\tau+1) = \Delta Q(\tau) - 1$.

We thus have the following,

$$\begin{aligned} &\sum_{\tau=1}^{t} I(k \in \mathcal{I}_{n(\tau)}) I(Q_k^{\text{inter}}(\tau) \geq 1) \\ &\cdot I(\beta_{k,n(\tau)}^{\text{in}}(\mathsf{cq}(\tau)) = 1) I(Q_k(\tau) = 0) \\ \leq &\sum_{i=1}^{M_s} \sum_{\tau=s_i}^{t_i-1} I(\Delta Q(\tau+1) < \Delta Q(\tau)) \end{aligned} \tag{F.3}$$

The reason is that any τ that satisfies (F.2) will have $\Delta Q(\tau) \geq 1$, and by our construction of s_i and t_i, for all $i = 1$ to M_s, such τ must fall into one of the intervals $[s_i, t_i)$. Also, any τ that satisfies (F.2) will have $\Delta Q(\tau + 1) < \Delta Q(\tau)$. As a result, we have (F.3). We can continue upper bounding (F.3) by

$$(F.3) \leq \sum_{i=1}^{M_s} \sum_{\tau=s_i}^{t_i-1} I(\Delta Q(\tau+1) > \Delta Q(\tau)) \tag{F.4}$$

$$= \sum_{i=1}^{M_s} \sum_{\tau=s_i}^{t_i-1} I(\Delta Q(\tau+1) > \Delta Q(\tau)) I(k \in \mathcal{O}_{n(\tau)}) \tag{F.5}$$

$$\leq \sum_{\tau=1}^{t} I(\Delta Q(\tau+1) > \Delta Q(\tau)) I(k \in \mathcal{O}_{n(\tau)}), \tag{F.6}$$

where (F.4) follows from (F.1); and (F.6) follows from including additionally those τ not in any of the interval $[s_i, t_i)$. Except for proving (F.5), the proof of Claim E.0.2 is complete.

In the remaining part of this proof, we will rigorously prove (F.5). To that end, we first notice that

$$\begin{aligned} & I(\Delta Q(\tau+1) - \Delta Q(\tau) > 0) \\ =& I(\Delta Q(\tau+1) - \Delta Q(\tau) > 0) \cdot I(k \in \mathcal{O}_{n(\tau)}) \\ & + I(\Delta Q(\tau+1) - \Delta Q(\tau) > 0) \cdot I(k \notin \mathcal{O}_{n(\tau)}) \end{aligned} \tag{F.7}$$

In the next paragraph, we will prove that when $k \notin \mathcal{O}_{n(\tau)}$, we always have either "$\Delta Q(\tau+1) - \Delta Q(\tau) \leq 0$" or "$\Delta Q(\tau) < 0$." It means that the term $I(\Delta Q(\tau+1) - \Delta Q(\tau) > 0) \cdot I(k \notin \mathcal{O}_{n(\tau)})$ is either 0 or the τ value is not counted in any of the $[s_i, t_i)$ intervals since by our construction we always have $\Delta Q(s_i) = 0$ and any $\tilde{\tau} \in (s_i, t_i)$ satisfying $\Delta Q(\tilde{\tau}) > 0$. As a result, (F.5) is true.

Consider the situation when $k \notin \mathcal{O}_{n(\tau)}$ and consider two sub-cases: If SA $n(\tau)$ turns out to be infeasible, then $Q_k(\tau+1) = Q_k(\tau) + \sum_{m=1}^{M} \alpha_{k,m} a_m(\tau)$. Also, we always have $Q_k^{\text{inter}}(\tau+1) \leq Q_k^{\text{inter}}(\tau) + \sum_{m=1}^{M} \alpha_{k,m} a_m(\tau)$ since $k \notin \mathcal{O}_{n(\tau)}$ implies that the intermediate queue length Q_k^{inter} can only decrease or remain the same (except when there is external arrival $\sum_{m=1}^{M}\alpha_{k,m}a_m(t)$). As a result, in this sub-case, we have $\Delta Q(\tau+1)-\Delta Q(\tau)\leq 0$.

In the second sub-case: SA $n(\tau)$ is feasible, we have $Q_k(\tau+1)=Q_k(\tau)+\sum_{m=1}^{M}\alpha_{k,m}a_m(\tau)-I(k\in\mathcal{I}_{n(\tau)})\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau))$. Namely, when not counting the external arrival, Q_k can now possibly decrease if $k\in\mathcal{I}_{n(\tau)}$ or it will remain the same if $k\notin\mathcal{I}_{n(\tau)}$. Our goal is to show that either "$\Delta Q(\tau+1)-\Delta Q(\tau)\leq 0$" or "$\Delta Q(\tau)<0$." To that end, we prove the equivalent statement that $\Delta Q(\tau)\geq 0$ implies $\Delta Q(\tau+1)=\Delta Q(\tau)$. Since SA $n(\tau)$ is feasible, we have $Q_k(\tau)\geq 1$. Since $\Delta Q(\tau)=Q_k^{\text{inter}}(\tau)-Q_k(\tau)\geq 0$, we have $Q_k^{\text{inter}}(\tau)\geq Q_k(\tau)\geq 1$. Therefore, if $k\in\mathcal{I}_{n(\tau)}$, then both $Q_k^{\text{inter}}(\tau)$ and $Q_k(\tau)$ will decrease by the same amount $\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau))$; and if $k\notin\mathcal{I}_{n(\tau)}$, both $Q_k^{\text{inter}}(\tau)$ and $Q_k(\tau)$ will remain the same except for the external arrival. We thus have $Q_k^{\text{inter}}(\tau+1)=Q_k^{\text{inter}}(\tau)+\sum_{m=1}^{M}\alpha_{k,m}a_m(t)-I(k\in\mathcal{I}_{n(\tau)})\beta^{\text{in}}_{k,n(\tau)}(\mathsf{cq}(\tau))$. Namely, $Q_k^{\text{inter}}(\tau)$ will experience the same change as $Q_k(\tau)$. As a result $\Delta Q(\tau+1)=\Delta Q(\tau)$. The proof of (F.5) is complete and the proof of Claim E.0.2 is thus also complete. ■

Proof of Claim E.0.3: If $k\notin\mathcal{O}_{n(\tau)}$, the LHS of (E.5) is zero and the inequality always holds. If $k\in\mathcal{O}_{n(\tau)}$, we claim that at least of the following 5 possible cases is true:

1. For all queues $k'\in\mathcal{I}_{n(\tau)}$, $Q_{k'}(\tau)\geq 1$ and $Q_{k'}^{\text{inter}}(\tau)\geq\beta^{\text{in}}_{k',n(\tau)}(\mathsf{cq}(\tau))$.
2. There exists a queue $k'\in\mathcal{I}_{n(\tau)}$ with $Q_{k'}^{\text{inter}}(\tau)<\beta^{\text{in}}_{k',n(\tau)}(\mathsf{cq}(\tau))$.
3. There exists a queue $k'\in\mathcal{I}_{n(\tau)}$ with $Q_{k'}(\tau)=0$; $\beta^{\text{in}}_{k',n(\tau)}(\mathsf{cq}(\tau))=0$; and $1\leq Q_{k'}^{\text{inter}}(\tau)$.
4. There exists a queue $k'\in\mathcal{I}_{n(\tau)}$ with $Q_{k'}(\tau)=0$; $\beta^{\text{in}}_{k',n(\tau)}(\mathsf{cq}(\tau))=0$; and $0\leq Q_{k'}^{\text{inter}}(\tau)<1$.
5. There exists a queue $k'\in\mathcal{I}_{n(\tau)}$ with $Q_{k'}(\tau)=0$ and $\beta^{\text{in}}_{k',n(\tau)}(\mathsf{cq}(\tau))=1$, and $1\leq Q_{k'}^{\text{inter}}(\tau)$.

The reason is as follows. If 1) does not hold, either there exists a k' such that $Q_{k'}^{\text{inter}}(\tau)<\beta^{\text{in}}_{k',n(\tau)}(\mathsf{cq}(\tau))$; or there exists a k' such that $Q_{k'}(\tau)=0$ and $Q_{k'}^{\text{inter}}(\tau)\geq\beta^{\text{in}}_{k',n(\tau)}(\mathsf{cq}(\tau))$. In the former scenario, we have 2). In the latter scenario, we can further partition the event based on the values of $\beta^{\text{in}}_{k',n(\tau)}(\mathsf{cq}(\tau))$ and $Q_{k'}^{\text{inter}}(\tau)$, which leads to 3) to 5).

In the case of 1), SA $n(\tau)$ is feasible in the beginning of time τ. Hence SA $n(\tau)$ will be activated. Since we now consider the scenario of $k \in \mathcal{O}_{n(\tau)}$, both $Q_k(\tau)$ and $Q_k^{\text{inter}}(\tau)$ increase by the same amount, $\beta^{\text{out}}_{k,n(\tau)}(\mathsf{cq}(\tau)) + \sum_{m=1}^{M} \alpha_{k,m} a_m(\tau)$. As a result, $\Delta Q(\tau+1) = \Delta Q(\tau)$. The LHS of (E.5) equals to zero and the inequality (E.5) holds.

In the case of 2), the first term of the RHS of (E.5) is at least 1 because there exists a queue $k' \in \mathcal{I}_{n(\tau)}$ such that $I(k' \in \mathcal{I}_{n(\tau)}) I(Q^{\text{inter}}_{k'}(\tau) < \beta^{\text{in}}_{k',n(\tau)}(\mathsf{cq}(\tau))) = 1$. Since the LHS of (E.5) is at most 1, the inequality (E.5) holds. We can observe the same relationship between 3) and the second term, 4) and the third term, and 5) and the fourth term of the RHS of (E.5). Since (E.5) holds for all 5 cases, the proof of Claim E.0.3 is complete. ■

Proof of Claim E.0.4: Notice that joinly Claims E.0.2 and E.0.3 immediately give us Claim 4. ■

G. SUBLINEARLY GROWTH OF THE INTERMEDIATE ACTUAL QUEUE AND THE AGGREGATED NULL ACTIVITIES

In the next lemma, we will shows that $\mathsf{SCH}_{\text{avg}}$ can sublinearly stabilize $Q_k^{\text{inter}}(t)$ and $N_{\mathsf{NA},k}(t)$ for all k.

Lemma G.0.6 *Consider any rate vector $\mathbf{R}$ such that there exist $\mathbf{s}_c \in \Lambda^\circ$ for all $c \in \mathsf{CQ}$ satisfying (6.5). The proposed SCH_{avg} can sublinearly stabilize $q_k^{inter}(t)$, $N_{\mathsf{NA},k}(t)$, and $Q_k^{inter}(t)$ for all k.*

We will prove the sublinear growth of the four quantities separately.

Proof of sublinearly growing $q_k(t)$ and $q_k^{inter}(t)$: First, we provide the conventional stability definition.

Definition G.0.1 *A queue length $q(t)$ is stable if*

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} \mathsf{E}\{|q(t)|\} < \infty. \tag{G.1}$$

And the network is stable if all the queues are stable.

As discussed in Section 6.2.2, the back-pressure vector computation (6.1) and the update rule (6.2) are only based on the expected input and output service rate matrix $\overline{\mathcal{B}^{\text{in}}(\mathsf{cq}(t))}$ and $\overline{\mathcal{B}^{\text{out}}(\mathsf{cq}(t))}$, which are deterministic matrices. As a result, they can be viewed as the virtual queue lengths of a deterministic SPN. In the existing proof in [31], it has been shown that the virtual queue length $\mathbf{q}(t)$ of a deterministic SPN can be stabilized by $\mathsf{SCH}_{\text{avg}}$. As a result, $\mathsf{SCH}_{\text{avg}}$ can also stabilize the virtual queue length $q_k(t)$ for all k in the given (0,1) random SPN.

Notice that given the past arrival vectors and the past and current channel quality, i.e., given $\mathsf{cq}(t)$ and $\{\mathbf{a}, \mathsf{cq}\}_1^{t-1}$, the quantity $\mathbf{q}(t)$ and $\mathbf{x}^*(t)$ is no longer random and is of deterministic value, see the update rules of (6.1) and (6.2). The following lemma establishes the connection between $\mathbf{q}(t)$ and $\mathbf{q}^{\text{inter}}(t)$.

Lemma G.0.7 $\mathbf{q}(t)$ *is the expectation of* $\mathbf{q}^{inter}(t)$ *conditioned on* $\{\mathbf{a}, \mathsf{cq}\}_1^{t-1}$*. That is,* $\mathbf{q}(t) = \mathsf{E}\{\mathbf{q}^{inter}(t)|\{\mathbf{a}, \mathsf{cq}\}_1^{t-1}\}$.

Proof of Lemma G.0.7: This lemma can be proven iteratively. When $t = 1$, since $\mathbf{q}(t) = \mathbf{q}^{\text{inter}}(t) = \mathbf{0}$, the zero vector, Lemma G.0.7 holds automatically. Suppose Lemma G.0.7 holds for some t. By comparing (6.2) and (6.6), we can see that Lemma G.0.7 holds for $t + 1$ as well. ■

For any $k \in \{1, 2, ..., K\}$, we square both sides of (6.7) and we thus have

$$\begin{aligned}
&q_k^{\text{inter}}(t+1)^2 - q_k^{\text{inter}}(t)^2 \\
=&(\mu_{\text{out},k}(t) - \mu_{\text{in},k}(t))^2 - 2q_k^{\text{inter}}(t)\left(\mu_{\text{out},k}(t) - \mu_{\text{in},k}(t)\right).
\end{aligned}$$

Similar to (6.8) and (6.9) we can define the average arrival rate and departure rate of queue k as follows.

$$\begin{aligned}
\overline{\mu_{\text{out},k}}(t) &= \sum_{n=1}^{N}\left(\overline{\beta_{k,n}^{\text{in}}(\mathsf{cq}(t))}x_n^*(t)\right), \\
\overline{\mu_{\text{in},k}}(t) &= \sum_{m=1}^{M}\left(\alpha_{k,m}a_m(t)\right) + \sum_{n=1}^{N}\left(\overline{\beta_{k,n}^{\text{out}}(\mathsf{cq}(t))}x_n^*(t)\right).
\end{aligned} \tag{G.2}$$

By taking the expectation conditioned on the past and current arrival vectors and past channel quality on both sides until time t, we have

$$\begin{aligned}
&\mathsf{E}\{q_k^{\text{inter}}(t+1)^2|\{\mathbf{a},\mathsf{cq}\}_1^t\} - \mathsf{E}\{q_k^{\text{inter}}(t)^2|\{\mathbf{a},\mathsf{cq}\}_1^t\}\\
=&\mathsf{E}\{(\mu_{\text{out},k}(t)-\mu_{\text{in},k}(t))^2|\{\mathbf{a},\mathsf{cq}\}_1^t\}\\
&-2\mathsf{E}\{q_k^{\text{inter}}(t)\left(\mu_{\text{out},k}(t)-\mu_{\text{in},k}(t)\right)|\{\mathbf{a},\mathsf{cq}\}_1^t\}\\
=&\mathsf{E}\{(\mu_{\text{out},k}(t)-\mu_{\text{in},k}(t))^2|\{\mathbf{a},\mathsf{cq}\}_1^t\}\\
&-2q_k(t)\left(\overline{\mu_{\text{out},k}}(t)-\overline{\mu_{\text{in},k}}(t)\right) \qquad \text{(G.3)}\\
\le&C^2+2|q_k(t)|U, \qquad \text{(G.4)}
\end{aligned}$$

where (G.3) follows from the observation that $q_k^{\text{inter}}(t)$ is a constant given $\{\mathbf{a},\mathsf{cq}\}_1^{t-1}$ and $\overline{\mu_{\text{out},k}}(t)$ and $\overline{\mu_{\text{in},k}}(t)$ are the conditional expectation of $\mu_{\text{out},k}(t)$ and $\mu_{\text{in},k}(t)$ (G.2) given $\{\mathbf{a},\mathsf{cq}\}_1^t$; and (G.4) follows from defining C to be the upper bound of $|\mu_{\text{out},k}(t)-\mu_{\text{in},k}(t)|$ and U to be the upper bound[1] of $|\overline{\mu_{\text{out},k}}(t)-\overline{\mu_{\text{in},k}}(t)|$. Now we take the expectation over all possible past arrival vectors and past channel quality.

$$\mathsf{E}\{q_k^{\text{inter}}(t+1)^2\} - \mathsf{E}\{q_k^{\text{inter}}(t)^2\} \le C^2 + 2U\mathsf{E}\{|q_k(t)|\}. \qquad \text{(G.5)}$$

Eq. (G.5) also holds if we replace the time index t by τ. By summing up (G.5) (with time index τ) for $\tau=1$ to $\tau=t-1$ and by noticing $q_k^{\text{inter}}(1)=0$, we have

$$\begin{aligned}
&\mathsf{E}\{q_k^{\text{inter}}(t)^2\} - \mathsf{E}\{q_k^{\text{inter}}(1)^2\}\\
=&\mathsf{E}\{q_k^{\text{inter}}(t)^2\} \le (t-1)C^2 + 2U\sum_{\tau=1}^{t-1}\mathsf{E}\{|q_k(\tau)|\}.
\end{aligned}$$

[1] C and U exist because $\mu_{\text{out},k}(t)$ and $\mu_{\text{in},k}(t)$ have bounded support by our definition.

Since $q_k(t)$ is stable and thus satisfies $\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=1}^{t}\mathsf{E}\{|q_k(\tau)|\}<\infty$, there exists an L value such that $\frac{1}{t}\sum_{\tau=1}^{t}\mathsf{E}\{|q_k(\tau)|\}\leq L$ for all possible t values. We then have

$$\begin{aligned}
&\frac{1}{t-1}\mathsf{E}\{q_k^{\text{inter}}(t)^2\}\\
\leq& C^2+2U\frac{1}{t-1}\sum_{\tau=1}^{t-1}\mathsf{E}\{|q_k(\tau)|\}\\
\leq& C^2+2UL.
\end{aligned}$$

for arbitrary t values.

For any arbitrarily given $\epsilon'>0$, we now apply Markov inequality with the second moment expression to derive

$$\mathsf{Prob}(|q_k^{\text{inter}}(t)|\geq\epsilon' t)\leq\frac{1}{\epsilon'^2t^2}\mathsf{E}\{q_k^{\text{inter}}(t)^2\}\leq\frac{C+2UL}{\epsilon'^2t}.$$

For any arbitrarily given $\delta>0$, let t_0 be the first t such that $\frac{C+2UL}{\epsilon'^2t}<\delta$. Then we have

$$\mathsf{Prob}(|q_k^{\text{inter}}(t)|\geq\epsilon' t)<\delta,\ \forall t>t_0.$$

Thus we have proven the sublinear growth of $\mathbf{q}^{\text{inter}}(t)$. ■

Before we continue our proofs of sublinearly growing $N_{\mathsf{NA},k}(t)$ and $Q_k^{\text{inter}}(t)$, we state the following claim first. Define *the deficit*, D_k, for all k as the difference between Q_k^{inter} and q_k^{inter}. That is, at any time t,

$$D_k(t)=Q_k^{\text{inter}}(t)-q_k^{\text{inter}}(t),\ \forall k. \tag{G.6}$$

Claim G.0.5 *For all k, the function $D_k(t)$ is non-decreasing and it grows sublinearly.*

The proof of Claim G.0.5 is relegated to Appendix J. We now continue our proofs.

Proof of sublinearly growing $N_{\mathsf{NA},k}(t)$: Recall the definition of the null activity at queue k ($k\in\mathcal{I}_{n(t)}$, and $Q_k^{\text{inter}}(t)<\mu_{\text{out},k}(t)$). In the proof of Claim G.0.5, in particular (J.1), we can see that the null activity occurs at queue k at time t if and only if $D_k(t+1) > D_k(t)$. As a result,

$$N_{\mathsf{NA},k}(t) = \sum_{\tau=1}^{t} I(D_k(\tau+1) > D_k(\tau)).$$

Recall that $Q_k^{\text{inter}}(t)$ is an integer-valued random process and so is $\mu_{\text{out},k}(t) = \sum_{n=1}^{N} \left(\beta_{k,n}^{\text{in}}(\mathsf{cq}(t)) \cdot x_n^*(t)\right)$. As a result, whenever $\mu_{\text{out},k}(t) - Q_k^{\text{inter}}(t) > 0$, we must have $\mu_{\text{out},k}(t) - Q_k^{\text{inter}}(t) \geq 1$. Using this observation and the fact that $D_k(t)$ is non-decreasing, we have

$$\sum_{\tau=1}^{t} I(D_k(\tau+1) > D_k(\tau)) \leq D_k(t+1).$$

The above argument implies $N_{\mathsf{NA},k}(t) \leq D_k(t+1)$. Since $D_k(t)$ grows sublinearly as proven in Claim G.0.5, we have proven that $N_{\mathsf{NA},k}(t)$ also grows sublinearly. ■

Proof of sublinearly growing $Q_k^{\text{inter}}(t)$: By (G.6),

$$Q_k^{\text{inter}}(t) = q_k^{\text{inter}}(t) + D_k(t).$$

We have shown that both q_k^{inter} and $D_k(t)$ grow sublinearly, and hence Q_k^{inter} also grows sublinearly. ■

The above discussion on $q_k(t)$, $q_k^{\text{inter}}(t)$, $N_{\mathsf{NA},k}(t)$, and $Q_k^{\text{inter}}(t)$ completes the proof of Lemma G.0.6.

H. THE MATCH OF THE SHANNON CAPACITY AND THE STABILITY REGION FOR 2-FLOW DOWNLINK BROADCAST PEC

To compare polytopes in Proposition 5.3.1 and Proposition 6.2.2, we first list all the linear constraints describing each region separately. For Proposition 6.2.2, the region can be described by (6.5). Following from Table 6.2, we can explicitly write $\mathcal{A}$ and $\mathcal{B}$ as follows. To facilitate matrix labeling, we order the 7 operations as $[\text{NC1}, \text{NC2}, \text{DX1}, \text{DX2}, \text{PM}, \text{RC}, \text{CX}]$, and order the 5 queues as $\left[\mathbf{Q}^1_{\emptyset}, \mathbf{Q}^2_{\emptyset}, \mathbf{Q}^1_{\{2\}}, \mathbf{Q}^2_{\{1\}}, \mathbf{Q}_{\text{mix}}\right]$. Let $\vec{p}^{[c]} \triangleq \vec{p}(c)$ for all $c \in \mathsf{CQ}$ be the probability vector which represents the reception status probabilities when the channel quality is c. Given the above definitions, we can write $\mathcal{A}$, $\overline{\mathcal{B}^{\text{in}}}$, $\overline{\mathcal{B}^{\text{out}}}$, and the average service vector, s_c, under channel quality c for any $c \in \mathsf{CQ}$ as

$$\mathcal{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}},$$

$$\overline{\mathcal{B}^{\text{in}}(c)} \\ = \begin{bmatrix} p^{[c]}_{d_1 \vee d_2} & 0 & 0 & 0 & p^{[c]}_{d_1 \vee d_2} & 0 & 0 \\ 0 & p^{[c]}_{d_1 \vee d_2} & 0 & 0 & p^{[c]}_{d_1 \vee d_2} & 0 & 0 \\ 0 & 0 & p^{[c]}_{d_1} & 0 & 0 & 0 & p^{[c]}_{d_1} \\ 0 & 0 & 0 & p^{[c]}_{d_2} & 0 & 0 & p^{[c]}_{d_2} \\ 0 & 0 & 0 & 0 & 0 & p^{[c]}_{d_1 \vee d_2} & 0 \end{bmatrix},$$

$$\overline{\mathcal{B}^{\text{out}}(c)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ p^{[c]}_{\overline{d_1} d_2} & 0 & 0 & 0 & 0 & p^{[c]}_{\overline{d_1} d_2} & 0 \\ 0 & p^{[c]}_{d_1 \overline{d_2}} & 0 & 0 & 0 & p^{[c]}_{d_1 \overline{d_2}} & 0 \\ 0 & 0 & 0 & 0 & p^{[c]}_{d_1 \vee d_2} & 0 & 0 \end{bmatrix},$$

$$\mathbf{s}_c = \left[x^{[c]}_{\text{NC1}}\ x^{[c]}_{\text{NC2}}\ x^{[c]}_{\text{DX1}}\ x^{[c]}_{\text{DX2}}\ x^{[c]}_{\text{PM}}\ x^{[c]}_{\text{RC}}\ x^{[c]}_{\text{CX}} \right]^{\mathsf{T}}.$$

As a result, the throughput region in Proposition 6.2.2 can be expressed by a collection of 5+1 linear (in)equalities, where the first 5 equalities correspond to the flow-conservation law of queues 1 to 5 and the 6-th inequalities follows from $\mathbf{s}_c$ being drawn from the convex hull Λ: That is,

$$\sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_{\text{NC1}} + x^{[c]}_{\text{PM}} \right) p^{[c]}_{d_1 \vee d_2} = R_1, \tag{H.1}$$

$$\sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_{\text{NC2}} + x^{[c]}_{\text{PM}} \right) p^{[c]}_{d_1 \vee d_2} = R_2, \tag{H.2}$$

$$\sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_{\text{CX}} + x^{[c]}_{\text{DX1}} \right) p^{[c]}_{d_1} = \sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_{\text{NC1}} + x^{[c]}_{\text{RC}} \right) p^{[c]}_{\overline{d_1} d_2}, \tag{H.3}$$

$$\sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_{\text{CX}} + x^{[c]}_{\text{DX2}} \right) p^{[c]}_{d_2} = \sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_{\text{NC2}} + x^{[c]}_{\text{RC}} \right) p^{[c]}_{d_1 \overline{d_2}}, \tag{H.4}$$

$$\sum_{\forall c \in \mathsf{CQ}} f_c x^{[c]}_{\text{RC}} p^{[c]}_{d_1 \vee d_2} = \sum_{\forall c \in \mathsf{CQ}} f_c x^{[c]}_{\text{PM}} p^{[c]}_{d_1 \vee d_2}, \tag{H.5}$$

$$x^{[c]}_{\text{NC1}} + x^{[c]}_{\text{NC2}} + x^{[c]}_{\text{DX1}} + x^{[c]}_{\text{DX2}} + x^{[c]}_{\text{PM}} + x^{[c]}_{\text{RC}} + x^{[c]}_{\text{CX}} \leq 1, \quad \forall c \in \mathsf{CQ}. \tag{H.6}$$

On the other hand, by Lemma 8 of [43], the polytype in Proposition 5.3.1 can also be expressed by another collection of linear (in)equalities:

$$x^{[c]}_0 + x^{[c]}_9 + x^{[c]}_{18} + x^{[c]}_{27} + x^{[c]}_{31} + x^{[c]}_{63} + x^{[c]}_{95} \leq 1, \forall c \in \mathsf{CQ}, \tag{H.7}$$

$$y_1 = \sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_0 + x^{[c]}_9 + x^{[c]}_{18} + x^{[c]}_{27} + x^{[c]}_{31} + x^{[c]}_{63} \right) p^{[c]}_{d_1}, \tag{H.8}$$

$$y_2 = \sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_0 + x^{[c]}_9 + x^{[c]}_{18} + x^{[c]}_{27} + x^{[c]}_{31} + x^{[c]}_{95} \right) p^{[c]}_{d_2}, \tag{H.9}$$

$$y_3 = R_1 + \sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_0 + x^{[c]}_9 \right) p^{[c]}_{d_1}, \tag{H.10}$$

$$y_4 = R_1 + \sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_0 + x^{[c]}_{18} + x^{[c]}_{27} \right) p^{[c]}_{d_2}, \tag{H.11}$$

$$y_5 = \sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_0 + x^{[c]}_9 + x^{[c]}_{18} \right) p^{[c]}_{d_1 \vee d_2}, \tag{H.12}$$

$$y_6 = R_1 + \sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_0 + x^{[c]}_9 \right) p^{[c]}_{d_1 \vee d_2}, \tag{H.13}$$

$$y_7 = R_2 + \sum_{\forall c \in \mathsf{CQ}} f_c \left(x^{[c]}_0 + x^{[c]}_{18} \right) p^{[c]}_{d_1 \vee d_2}, \tag{H.14}$$

and

$$y_1 = y_3; \quad y_2 = y_4; \tag{H.15}$$

$$y_5 = y_6 = y_7 = R_1 + R_2. \tag{H.16}$$

To prove that the dynamic-arrival stability region in (H.1)-(H.6) matches the block-coding capacity in (H.7)–(H.16), we need to prove that for any (R_1, R_2) and the accompanying $x_{(\cdot)}^{[c]}$ and $y_{(\cdot)}$ variables satisfying (H.7) to (H.16), we can always find out another set of $\mathbf{s}_c = [x_{\text{NC1}}^{[c]}, x_{\text{NC2}}^{[c]}, x_{\text{DX1}}^{[c]}, x_{\text{DX2}}^{[c]}, x_{\text{PM}}^{[c]}, x_{\text{RC}}^{[c]}, x_{\text{CX}}^{[c]}]$ variables such that (R_1, R_2) and $\mathbf{s}_c$ jointly satisfying (H.1) to (H.6). To do so, we will verify that the following one-to-one mapping $\mathbf{x}_{(\cdot)}^{[c]}$ satisfies (H.1) to (H.6).

$$\begin{aligned}
&x_{\text{NC1}}^{[c]} = x_{18}^{[c]},\ x_{\text{NC2}}^{[c]} = x_{9}^{[c]},\ x_{\text{DX1}}^{[c]} = x_{63}^{[c]}, x_{\text{DX2}}^{[c]} = x_{95}^{[c]},\\
&x_{\text{PM}}^{[c]} = x_{0}^{[c]},\ x_{\text{RC}}^{[c]} = x_{27}^{[c]},\ x_{\text{CX}}^{[c]} = x_{31}^{[c]}.
\end{aligned} \tag{H.17}$$

Ineq. (H.6) is true as a direct result of (H.7). We now prove that (H.1) holds. By (H.16), we have $y_7 = R_1 + R_2$. By (H.14), we then have

$$\begin{aligned}
&y_7 = R_2 + \sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{PM}}^{[c]} + x_{\text{NC1}}^{[c]} \right) p_{d_1 \vee d_2}^{[c]} = R_1 + R_2\\
\Rightarrow &\sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{PM}}^{[c]} + x_{\text{NC1}}^{[c]} \right) p_{d_1 \vee d_2}^{[c]} = R_1,
\end{aligned}$$

which implies (H.1). (H.2) can be proven by symmetric arguments. Next we check (H.5). Again by the fact that $y_5 = R_1 + R_2$, we have

$$\begin{aligned}
y_5 &= \sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{PM}}^{[c]} + x_{\text{NC1}}^{[c]} + x_{\text{NC2}}^{[c]} + x_{\text{RC}}^{[c]} \right) p_{d_1 \vee d_2}^{[c]}\\
&= R_1 + R_2\\
\Rightarrow &\sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{NC1}}^{[c]} + x_{\text{RC}}^{[c]} \right) p_{d_1 \vee d_2}^{[c]} = R_1,
\end{aligned} \tag{H.18}$$

where (H.18) follows from substituting (H.2) into $y_5 = R_1 + R_2$. Combining (H.18) with (H.1), we have

$$\begin{aligned}
R_1 &= \sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{NC1}}^{[c]} + x_{\text{RC}}^{[c]} \right) p_{d_1 \vee d_2}^{[c]} \\
&= \sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{NC1}}^{[c]} + x_{\text{PM}}^{[c]} \right) p_{d_1 \vee d_2}^{[c]} \\
\Rightarrow \sum_{\forall c \in \mathsf{CQ}} f_c x_{\text{RC}}^{[c]} p_{d_1 \vee d_2}^{[c]} &= \sum_{\forall c \in \mathsf{CQ}} f_c x_{\text{PM}}^{[c]} p_{d_1 \vee d_2}^{[c]},
\end{aligned}$$

which implies (H.5). Finally we check (H.3) and (H.4). By (H.15), we have

$$\begin{aligned}
& y_3 = y_1 \\
\Rightarrow & \sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{PM}}^{[c]} + x_{\text{NC1}}^{[c]} + x_{\text{NC2}}^{[c]} + x_{\text{RC}}^{[c]} + x_{\text{CX}}^{[c]} + x_{\text{DX1}}^{[c]} \right) p_{d_1}^{[c]} \\
& = R_1 + \sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{PM}}^{[c]} + x_{\text{NC2}}^{[c]} \right) p_{d_1}^{[c]} \\
\Rightarrow & \sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{NC1}}^{[c]} + x_{\text{RC}}^{[c]} + x_{\text{CX}}^{[c]} + x_{\text{DX1}}^{[c]} \right) p_{d_1}^{[c]} = R_1.
\end{aligned} \tag{H.19}$$

Combining (H.19) and (H.18), we have

$$\begin{aligned}
R_1 &= \sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{NC1}}^{[c]} + x_{\text{RC}}^{[c]} + x_{\text{CX}}^{[c]} + x_{\text{DX1}}^{[c]} \right) p_{d_1}^{[c]} \\
&= \sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{NC1}}^{[c]} + x_{\text{RC}}^{[c]} \right) p_{d_1 \vee d_2}^{[c]}.
\end{aligned} \tag{H.20}$$

Following from the fact that $p_{d_1 \vee d_2}^{[c]} = p_{d_1}^{[c]} + p_{\overline{d_1} d_2}^{[c]}$, we can rewrite (H.20) as

$$\sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{CX}}^{[c]} + x_{\text{DX1}}^{[c]} \right) p_{d_1}^{[c]} = \sum_{\forall c \in \mathsf{CQ}} f_c \left(x_{\text{NC1}}^{[c]} + x_{\text{RC}}^{[c]} \right) p_{\overline{d_1} d_2}^{[c]},$$

which implies(H.3). (H.4) can be derived by symmetric arguments. Thus we complete the proof of Proposition 6.3.1.

I. THE LOWER BOUND OF THE SUMMATION OF RANDOM VARIABLES

We use $\mathcal{P}$ to denote a finite collection of probability distributions and each distribution is of zero mean and finite support. For simplicity, we say $\mathcal{P} = \{P_1, P_2, \cdots, P_K\}$ where $K = |\mathcal{P}|$.

Lemma I.0.8 *There exists a fixed constant $C > 0$ such that for any arbitrary K non-negative integers $L_1, L_2, ..., L_K$, the following inequality always holds.*

$$\mathsf{Prob}(\sum_{k=1}^{K}\sum_{i=1}^{L_k} X_i^{(k)} \geq 0) > C \tag{I.1}$$

where for any k, the random variables $X_i^{(k)} \sim P_k$ are i.i.d. for different i values and the random processes $\{X_i^{(k)} : i\}$ are independently distributed for different k values.

Proof: We prove this lemma by induction on the size of $\mathcal{P}$. When $K = |\mathcal{P}| = 1$, the probability of interest becomes $\mathsf{Prob}(\sum_i^L X_i \geq 0)$ where we drop the index k for simplicity. By the central limit theorem, there exists an l_0 such that when $L > l_0$, the probability of interest is $> 1/4$ (which can be made arbitrarily close to $1/2$ but we choose $1/4$ for simplicity). Choose $C = \min(\min\{\mathsf{Prob}(\sum_{l=1}^L X_i \geq 0) : l \leq l_0\}, 1/4)$. We claim that such a C value is strictly positive. The reason is that $\min\{\mathsf{Prob}(\sum_{l=1}^L X_i \geq 0) : l \leq l_0\} \neq 0$ because of the assumptions that X_i is zero mean and i.i.d. From the above construction we have

$$\mathsf{Prob}(\sum_{i=1}^{L} X_i \geq 0) > C, \quad \forall L. \tag{I.2}$$

We now consider the case of $K = |\mathcal{P}| \geq 2$. For any arbitrarily given L_1 to L_K, the probability of interest satisfies

$$\begin{aligned} &\mathsf{Prob}(\sum_{k=1}^{K}\sum_{i=1}^{L_1} X_i^{(k)} \geq 0) \\ \geq &\mathsf{Prob}(\sum_{i=1}^{L_k} X_i^{(k)} \geq 0, \forall k) \\ = &\prod_{k=1}^{K} \mathsf{Prob}(\sum_{i=1}^{L_k} X_i^{(k)} \geq 0). \end{aligned} \tag{I.3}$$

We have shown that for each k, there exists a constant $C_k > 0$ such that $\mathsf{Prob}(\sum_{i=1}^{L_k} X_i^{(k)} \geq 0) > C_k$ for any arbitrary L_k. Hence the product in (I.3) is larger than $C \triangleq \prod_{k=1}^{K} C_k$ for any arbitrary L_1 to L_K. Lemma I.0.8 is thus proven.

J. THE NON-DECREASING PROPERTY AND THE SUBLINEAR GROWTH OF THE DEFICITS

For all k, the reason why $D_k(t)$ is non-decreasing is because

$$\begin{aligned}
D_k(t+1) &= Q_k^{\text{inter}}(t+1) - q_k^{\text{inter}}(t+1) \\
&= \left(Q_k^{\text{inter}}(t) - \mu_{\text{out},k}(t)\right)^+ - \left(q_k^{\text{inter}}(t) - \mu_{\text{out},k}(t)\right) \\
&= Q_k^{\text{inter}}(t) - \mu_{\text{out},k}(t) + \left(\mu_{\text{out},k}(t) - Q_k^{\text{inter}}(t)\right)^+ \\
&\quad - \left(q_k^{\text{inter}}(t) - \mu_{\text{out},k}(t)\right) \\
&= D_k(t) + \left(\mu_{\text{out},k}(t) - Q_k^{\text{inter}}(t)\right)^+ . \qquad \text{(J.1)}
\end{aligned}$$

We now prove that $D_k(t)$ grows sublinearly for all k. Define $p_k(t) \triangleq -q_k^{\text{inter}}(t-1) + \mu_{\text{out},k}(t-1)$ for all $t \geq 2$ and $p_k(1) = -q_k^{\text{inter}}(1) = 0$. Notice that $p_k(t)$ grows sublinearly because $q_k^{\text{inter}}(t)$ grows sublinearly and $\mu_{\text{out},k}(t-1)$ is bounded. We notice that $D_k(t)$ is the running maximum of $p_k(t)$ since by (J.1),

$$\begin{aligned}
D_k(t) &= D_k(t-1) + \left(\mu_{\text{out},k}(t-1) - Q_k^{\text{inter}}(t-1)\right)^+ \\
&= D_k(t-1) + \max\{0, \mu_{\text{out},k}(t-1) - Q_k^{\text{inter}}(t-1)\} \\
&= \max\{D_k(t-1), -q_k^{\text{inter}}(t-1) + \mu_{\text{out},k}(t-1)\} \qquad \text{(J.2)} \\
&= \max\{D_k(t-1), p_k(t)\} \\
&= \max_{1 \leq \tau \leq t} p_k(\tau),
\end{aligned}$$

where (J.2) follows from (G.6).

Recall $\overline{\mu_{\text{out},k}}(t)$ and $\overline{\mu_{\text{in},k}}(t)$ are the expectation of $\mu_{\text{out},k}(t)$ and $\mu_{\text{in},k}(t)$, respectively, conditioned on the arrival vectors and the channel quality until time t. And by (6.2), we can update $q_k(t)$ as

$$q_k(t+1) = q_k(t) - \overline{\mu_{\text{out},k}}(t) + \overline{\mu_{\text{in},k}}(t),\ \forall k. \tag{J.3}$$

Define $\overline{p_k}(t) \stackrel{\triangle}{=} -q_k(t-1) + \overline{\mu_{\text{out},k}}(t-1) = -q_k(t) + \overline{\mu_{\text{in},k}}(t-1)$. That is, $\overline{p_k}(t)$ is the conditional expectation of $p_k(t)$ given $\{\mathbf{a}, \mathsf{cq}\}_1^{t-1}$. Define $p'_k(t) \stackrel{\triangle}{=} p_k(t) - \overline{p_k}(t)$. That is, $p'_k(t)$ is the difference between the random variable $p_k(t)$ and its conditional expectation $\overline{p_k}(t)$. Thus far, we have decompose

$$p_k(t) = \overline{p_k}(t) + p'_k(t)$$

as the summation of the average term $\overline{p_k}(t)$ and the random variation term $p'_k(t)$, where the latter has zero mean. We now define $D'_k(t)$ to be the running maximum of the $p'_k(t)$ and $\overline{D_k}(t)$ to be the running maximum of $\overline{p_k}(t)$. That is,

$$D'_k(t) = \max_{1\leq\tau\leq t} p'_k(\tau),$$
$$\overline{D_k}(t) = \max_{1\leq\tau\leq t} \overline{p_k}(\tau).$$

In the following, we will prove: Step 1: $\overline{p_k}(t)$ is stable and $p'_k(t)$ grows sublinearly; Step 2: $D'_k(t)$ grows sublinearly; and Step 3: $\overline{D_k}(t)$ grows sublinearly. Note that by definition, we always have $0 \leq D_k(t) \leq D'_k(t) + \overline{D_k}(t)$. As a result, Steps 2 and 3 imply $D_k(t)$ also grows sublinearly. The proof is complete.

Step 1: $\overline{p_k}(t)$ is stable because $q_k(t)$ is stable and $\overline{\mu_{\text{in},k}}(t-1)$ is bounded. Furthermore, $p'_k(t)$ grows sublinearly from the fact that the summation/difference of one stable queue and one sublinearly stable queue is sublinearly stable[1]. The proof of Step 1 is complete. ■

Step 2: We now show that $D_k(t)$ grows sublinearly. Recall that $p_k(t)$ is the random variation term with mean zero and $D_k(t)$ is the running maximum of the random variation.

[1]One can easily verify that with bounded initial value, stability implies sublinear stability.

As a result, in essence, the $D_k(t)$ is similar to the running maximum of a random walk with zero drift. The following proof is adapted from the standard proof that the running maximum of a zero-drift random walk is sublinearly growing [Chapter 4, [49]].

Let $T'_k(b) \triangleq \min\{t \geq 1 : p'_k(t) \geq b\}$ be the hitting time of $p'_k(t)$ exceeding the threshold b.

Claim J.0.6 *There exists $C > 0$ such that for all $t \geq 1$, all $b > 0$, and all possible past arrival vector realizations and channel quality realizations $\{\mathbf{a}, \mathsf{cq}\}_1^{t-1}$, we have*

$$\mathsf{Prob}(p'_k(t) \geq b | T'_k(b) \leq t, \{\mathbf{a}, \mathsf{cq}\}_1^{t-1}) > C. \tag{J.4}$$

Proof of Claim J.0.6: Let $\Delta\mu_{\text{in},k}(t) \triangleq \mu_{\text{in},k}(t) - \overline{\mu_{\text{in},k}}(t)$, $\Delta\mu_{\text{out},k}(t) \triangleq \mu_{\text{out},k}(t) - \overline{\mu_{\text{out},k}}(t)$, and $\Delta\mu_k(t) \triangleq \Delta\mu_{\text{in},k}(t) - \Delta\mu_{\text{out},k}(t)$. By (J.3) and (6.7),

$$\begin{aligned} q_k^{\text{inter}}(t) &= \sum_{\tau=1}^{t-1} (\mu_{\text{in},k}(\tau) - \mu_{\text{out},k}(\tau)), \\ q_k(t) &= \sum_{\tau=1}^{t-1} (\overline{\mu_{\text{in},k}}(\tau) - \overline{\mu_{\text{out},k}}(\tau)), \\ \text{and } q_k^{\text{inter}}(t) - q_k(t) &= \sum_{\tau=1}^{t-1} \Delta\mu_k(\tau). \end{aligned}$$

Then by the definitions of $p_k(t)$, $\overline{p_k}(t)$, and $p'_k(t)$, we have

$$\begin{aligned} p_k(t) &= -\sum_{\tau=1}^{t-2} (\mu_{\text{in},k}(\tau) - \mu_{\text{out},k}(\tau)) + \mu_{\text{out},k}(t-1), \\ \overline{p_k}(t) &= -\sum_{\tau=1}^{t-2} (\overline{\mu_{\text{in},k}}(\tau) - \overline{\mu_{\text{out},k}}(\tau)) + \overline{\mu_{\text{out},k}}(t-1), \\ p'_k(t) &= -\sum_{\tau=1}^{t-2} \Delta\mu_k(\tau) + \Delta\mu_{\text{out},k}(t-1). \end{aligned} \tag{J.5}$$

By (J.5) we have

$$
\begin{aligned}
& p'_k(t) - p'_k(T'_k(b)) \\
&= \left(\sum_{\tau=1}^{t-2} \Delta\mu_k(\tau) + \Delta\mu_{\text{out},k}(t-1) \right) \\
&\quad - \left(\sum_{\tau=1}^{T'_k(b)-2} \Delta\mu_k(\tau) + \Delta\mu_{\text{out},k}(T'_k(b)-1) \right) \\
&= \sum_{\tau=T'_k(b)-1}^{t-2} \Delta\mu_k(\tau) + \Delta\mu_{\text{out},k}(t-1) - \Delta\mu_{\text{out},k}(T'_k(b)-1) \\
&= \sum_{\tau=T'_k(b)}^{t-2} \Delta\mu_k(\tau) + \Delta\mu_{\text{out},k}(t-1) \\
&\quad + (\Delta\mu_{\text{in},k}(T'_k(b)-1) - 2\Delta\mu_{\text{out},k}(T'_k(b)-1))
\end{aligned}
$$

Define $\Delta\hat{\mu}_k(T'_k(b)-1) \stackrel{\triangle}{=} \Delta\mu_{\text{in},k}(T'_k(b)-1) - 2\Delta\mu_{\text{out},k}(T'_k(b)-1)$. Thus, we have

$$
\begin{aligned}
& \mathsf{Prob}(p'_k(t) \geq b | T'_k(b) \leq t, \{\mathbf{a}, \mathsf{cq}\}_1^t) \\
\geq & \mathsf{Prob}\Bigg(\Delta\hat{\mu}_k(T'_k(b)-1) + \sum_{\tau=T'_k(b)}^{t-2} \Delta\mu_k(\tau) \\
& \quad + \Delta\mu_{\text{out},k}(t-1) \geq 0 | T'_k(b) \leq t, \{\mathbf{a}, \mathsf{cq}\}_1^t \Bigg)
\end{aligned} \tag{J.6}
$$

We now notice that in the RHS of (J.6), there are $(t - T'_k(b) + 1)$ summands in the probability expression, one for each $\tau \in [T'_k(b)-1, t-1]$. One can easily verify that conditioning on the past arrival vectors and past channel quality $\{\mathbf{a}, \mathsf{cq}\}_1^t$, each summand is independently distributed. The reason is that when conditioning on $\{\mathbf{a}, \mathsf{cq}\}_1^t$, both the virtual queue length vector $\mathbf{q}(\tau)$ and back-pressure scheduler become deterministic for all $\tau = 1$ to t, see (5.2), (6.1), and (6.2). As a result, the randomness of each summand depends only on the realization of $\beta_{k,n}^{\text{in}}(\tau)$ and $\beta_{k,n}^{\text{out}}(\tau)$ and they are independently distributed in our SPN model. Moreover, each summand is also of zero mean and bounded support. The reason is that the definitions of $\Delta\mu_{\text{in},k}(\tau)$, $\Delta\mu_{\text{out},k}(\tau)$, and $\Delta\mu_k(\tau)$ ensure that these random variables are of zero mean. Also, since $\mathcal{B}^{\text{in}}(\tau)$ and $\mathcal{B}^{\text{out}}(\tau)$ are of bounded support, so are $\Delta\mu_{\text{in},}(\tau)$, $\Delta\mu_{\text{out},k}(\tau)$, and $\Delta\mu_k(\tau)$.

Obviously the conditional distribution of each of the $t - T'_k(b) + 1$ summands given $\{\mathbf{a}, \mathsf{cq}\}_1^t$ depends on the values of $T'_k(b)$ and t and the realization $\{\mathbf{a}, \mathsf{cq}\}_1^t$. However, we further argue that there is a bounded number of distributions, denoted by $\mathcal{P}$, and each of the conditional distribution must be of a distribution $P \in \mathcal{P}$ regardless what are the values of t, $T'_k(b)$, and the realization $\{\mathbf{a}, \mathsf{cq}\}_1^t$. Namely, even though there are infinitely many ways of having the t, $T'_k(b)$, and the realization $\{\mathbf{a}, \mathsf{cq}\}_1^t$ values, the number of possible distributions for all the summands is bounded. The reason is that the distributions of $\Delta\mu_{\text{in},}(\tau)$, $\Delta\mu_{\text{out},k}(\tau)$, and $\Delta\mu_k(\tau)$ depend only on what is the actual schedule at time τ. Since there is only a bounded number of possible scheduling decisions, the number of possible distributions for all the summands is bounded.

By Lemma I.0.8 in Appendix I, there exists a $C > 0$ such that

$$(J.6) > C \tag{J.7}$$

for all t and all possible past arrival vector realizations and channel quality realizations $\{\mathbf{a}, \mathsf{cq}\}_1^{t-1}$. The proof of Claim J.0.6 is complete. ■

Notice that by Claim J.0.6, there exists C such that for all possible past arrival vector and channel quality realizations

$$\begin{aligned}
&\mathsf{Prob}(p'_k(t) \geq b | T'_k(b) \leq t, \{\mathbf{a}, \mathsf{cq}\}_1^{t-1}) \\
=&\frac{\mathsf{Prob}(p'_k(t) \geq b, T'_k(b) \leq t | \{\mathbf{a}, \mathsf{cq}\}_1^{t})}{\mathsf{Prob}(T'_k(b) \leq t | \{\mathbf{a}, \mathsf{cq}\}_1^{t-1})} \\
=&\frac{\mathsf{Prob}(p'_k(t) \geq b | \{\mathbf{a}, \mathsf{cq}\}_1^{t-1})}{\mathsf{Prob}(T'_k(b) \leq t | \{\mathbf{a}, \mathsf{cq}\}_1^{t-1})} > C.
\end{aligned} \tag{J.8}$$

Meanwhile, since $D'_k(t)$ is the running maximum of $p'_k(t)$, we have

$$\begin{aligned}&\mathsf{Prob}(D'_k(t) \geq b | \{\mathbf{a}, \mathsf{cq}\}_1^{t-1}) = \mathsf{Prob}(T'_k(b) \leq t | \{\mathbf{a}, \mathsf{cq}\}_1^{t-1}) \\ <& \frac{1}{C}\mathsf{Prob}(p'_k(t) \geq b | \{\mathbf{a}, \mathsf{cq}\}_1^{t-1}).\end{aligned} \tag{J.9}$$

Taking the expectation on both sides over all possible past arrival vectors and past channel quality, we have

$$\mathsf{Prob}(D'_k(t) \geq b) < \frac{1}{C}\mathsf{Prob}(p'_k(t) \geq b). \tag{J.10}$$

Substituting b by ϵt in the above equation and using the fact that $p'_k(t)$ grows sublinearly, we have proven that $D'_k(t)$ grows sublinearly. The proof of Step 2 is complete. ■

Step 3: We now prove the following claim.

Claim J.0.7 *The following two inequalities are true for all possible realizations.*

1. $\overline{D_k}(t+1)^2 - \overline{D_k}(t)^2 \leq \max\{\overline{p_k}(t+1)^2 - \overline{p_k}(t)^2, 0\} + U^2$, *where U is the supremum over all possible $|\overline{\mu_{\mathit{out},k}}(t) - \overline{\mu_{\mathit{in},k}}(t-1)|$. Note that U always exists since in the random external arrivals and the random movements of the packets all have bounded support and $\overline{\mu_{\mathit{in},k}}(t)$ and $\overline{\mu_{\mathit{out},k}}(t)$ are computed from the expected values of the random packets arrival and departures.*

2. $\max\{\overline{p_k}(t+1)^2 - \overline{p_k}(t)^2, 0\} + U^2 \leq 2|\overline{p_k}(t)|U + 2U^2$.

Proof of Claim J.0.7: We first prove 1). There are three possible cases.

Case 1: $\overline{D_k}(t) \geq \overline{p_k}(t+1)$. Since $\overline{D_k}(t)$ is the running maximum of $\overline{p_k}(t)$, $\overline{D_k}(t+1) = \overline{D_k}(t)$ in this case. Thus the left hand side of (i) is zero and the inequality holds.

Case 2: $\overline{D_k}(t) < \overline{p_k}(t+1)$ and $\overline{p_k}(t) \geq 0$. By the definition of $\overline{D_k}(t)$, we have $\overline{D_k}(t+1) = \overline{p_k}(t+1)$. Also, since $\overline{D_k}(t)$ is the running maximum of $\overline{p_k}(t)$, we have $0 \leq \overline{p_k}(t) \leq \overline{D_k}(t)$, which implies $(\overline{p_k}(t))^2 \leq (\overline{D_k}(t))^2$. Jointly, we thus have $\overline{D_k}(t+1)^2 - \overline{D_k}(t)^2 \leq \overline{p_k}(t+1)^2 - \overline{p_k}(t)^2 \leq \max\{\overline{p_k}(t+1)^2 - \overline{p_k}(t)^2, 0\} + U^2$.

Case 3: $\overline{D_k}(t) < \overline{p_k}(t+1)$ and $\overline{p_k}(t) < 0$. By the definition of U and by (J.5), we have

$\overline{p_k}(t+1) \le \overline{p_k}(t) + U$, which, together with the inequality $D_k(t) < \overline{p_k}(t+1)$ and the definition that $\overline{D_k}(t)$ being the running maximum of $\overline{p_k}(t)$, implies

$$\overline{D_k}(t) - U \le \overline{p_k}(t) \le \overline{D_k}(t).$$

Since $\overline{D_k}(t)$ is always no less than zero, we thus have $-U \le \overline{p_k}(t) < 0$, which in turn implies $U^2 - \overline{p_k}(t)^2 \ge 0$. Since $\overline{D_k}(t+1) = \overline{p_k}(t+1)$, we now have, $\overline{D_k}(t+1)^2 - \overline{D_k}(t)^2 \le \overline{p_k}(t+1)^2 \le \max\{\overline{p_k}(t+1)^2 - \overline{p_k}(t)^2, 0\} + U^2$. The proof of 1) is complete.

We now prove 2). Define $\Delta\overline{p_k}(t+1) \triangleq \overline{p_k}(t+1) - \overline{p_k}(t)$. Then

$$\begin{aligned}
&\max\{\overline{p_k}(t+1)^2 - \overline{p_k}(t)^2, 0\} + U^2 \\
=&\max\{(\overline{p_k}(t) + \Delta\overline{p_k}(t+1))^2 - \overline{p_k}(t)^2, 0\} + U^2 \\
=&\max\{2\overline{p_k}(t)\Delta\overline{p_k}(t+1) + \Delta\overline{p_k}(t+1)^2, 0\} + U^2 \\
\le&2|\overline{p_k}(t)\Delta\overline{p_k}(t+1)| + |\Delta\overline{p_k}(t+1)^2| + U^2 \\
\le&2|\overline{p_k}(t)|U + 2U^2,
\end{aligned}$$

where the last inequality follows from rewriting $\overline{\mu_{\text{in},k}}(\tau)$ and $\overline{\mu_{\text{out},k}}(\tau)$ based on (J.5) and by the definition of U. ■

Following from Claim J.0.7 and taking the expectation on both sides over all possible arrival vectors,

$$\mathsf{E}\{\overline{D_k}(t+1)^2\} - \mathsf{E}\{\overline{D_k}(t)^2\} \le 2\mathsf{E}\{|\overline{p_k}(t)|\}U + 2U^2.$$

Replacing the time index t by τ and then summing up the above inequality with time index τ) from $\tau = 1$ to $\tau = t-1$, we then have

$$\begin{aligned}
&\mathsf{E}\{\overline{D_k}(t)^2\} \le 2U\sum_{\tau=1}^{t-1}\mathsf{E}\{|\overline{p_k}(\tau)|\} + 2U^2(t-1) \\
\Rightarrow&\frac{1}{t}\mathsf{E}\{\overline{D_k}(t)^2\} \le 2U\frac{1}{t-1}\sum_{\tau=1}^{t-1}\mathsf{E}\{|\overline{p_k}(\tau)|\} + 2U^2.
\end{aligned}$$

The fact that $\overline{p_k}(t)$ is stable implies that there exists an L value such that $\frac{1}{t-1}\sum_{\tau=1}^{t-1} \mathsf{E}\{|\overline{p_k}(\tau)|\} \leq L$ for all t. For any $\epsilon' > 0$, $\delta > 0$, we then apply the Markov inequality,

$$\mathsf{Prob}(\overline{D_k}(t) > \epsilon' t) \leq \frac{1}{\epsilon'^2 t^2}\mathsf{E}\{\overline{D_k}(t)^2\} \leq \frac{1}{\epsilon'^2 t}\left(2UL + 2U^2\right).$$

Let t_0 be the smallest t such that $\frac{1}{\epsilon'^2 t}\left(2UL + 2U^2\right) < \delta$. Then $\mathsf{Prob}(\overline{D_k}(t) > \epsilon' t) < \delta$ for all $t > t_0$, which completes the proof of Step 3. ■

VITA

VITA

Wei-Cheng Kuo received his B.S. in Electrical Engineering at National Chiao-Tung University in 2008. He is working towards the PhD degree at Purdue University under the supervision of Prof Chih-Chun Wang since 2009. His research interest mainly covers Shannon capacity region analysis and stability region analysis for local wireless network.