ABSTRACT

The main driving force for the evolution of wireless systems is to provide increased data-rate. Over the past few years, many new wireless techniques have emerged in multiple communication layers to provide more efficient use of bandwidth, such as rateless codes, hybrid ARQ, multi-user MIMO, full-duplex transmission. In order to support the ever-increasing wireless traffic demand, it is crucial to understand the benefits of these schemes in terms of throughput gains, and develop efficient algorithms to reap the full potential of these new wireless techniques. Hence, in this dissertation, we focus on the characterization of throughput and the design of control algorithms in wireless systems.

We first consider the point-to-point channel and focus on the distribution of delay, a metric that is closely related to link-throughput and determined jointly by the selection of physical-layer code-rate and link-layer retransmission scheme. Our investigations reveal the surprising results that, when decoder memory is not used to cache undecodable transmissions and there is a lack of redundancy in the packet, a light-tailed packet-size distribution may translate into a heavy-tailed transmission delay, leading to a poor link-throughput (possibly even zero-throughput), whereas the delay will be lighted-tailed if either the decoder uses memory to cache failed transmissions or the amount of redundancy in the packets passes a certain threshold. Next, we shift our focus from the point-to-point channel to the point-to-multipoint channel and investigate how the broadcast-throughput behaves as a function of network size.
and coding block size, when an optimal rateless codes is used. Using large deviation theory, we are able to obtain a close-form expression of the asymptotic throughput (asymptotic in the number of multicast receivers) for any mapping of the network size to the coding block size. This asymptotic throughput result leads us to find a lower-bound on the throughput for any finite values of coding block size and network size, which is also asymptotically tight.

We next shift our focus from link-layer throughput to the MAC-layer throughput-region in multi-hop wireless networks. By assuming a binary interference relationship between links, we provide a thorough comparison of the throughput-regions that can be achieved under different combinations of multi-antenna techniques, such as MIMO multiplexing, multi-user MIMO and wireless full-duplex, from a degree-of-freedom perspective. Our results give clear guidelines on which multi-antenna architecture and traffic pattern could result in throughput improvement for one scheme over another. While the throughput-regions of different schemes may be compared, the adoption of new schemes raises challenges in the design of control algorithms that aim to support the entire throughput-region. Indeed, when each node is capable of wireless full-duplex cut-through transmission, the MAC-layer throughput-region is directly a function of the routing decision, leading to a strong coupling between routing and scheduling, which has not been seen in the traditional half-duplex network. Also, it is unclear how to dynamically form/change cut-through routes based on the traffic rates and patterns. In this dissertation, we introduce a novel method to characterize the interference relationship between links in the network with cut-through transmission, which decouples the routing decision with the scheduling decision and enables a seamless adaptation of traditional half-duplex algorithms into wireless networks with full-duplex cut-through capabilities.
To my parents, Yanqin Sun and Wenming Yang
and my wife, Xuetao Ji.
ACKNOWLEDGMENTS

First and foremost, my sincere gratitude to Prof. Ness B. Shroff, a great advisor, an inspiring mentor and a caring friend to me. As an advisor, his insightful comments, thought-provoking feedbacks, and ability to simplify complex problems while looking at the high-level pictures provide immense help for the growth of this thesis work. As a mentor, he teaches me invaluable lessons on how to identify, formulate and solve problems; how to write high-quality papers and give effective presentations; and everything you need to learn in order to become an independent researcher. As a friend, he is a great source of encouragement and support.

I would like to thank Prof. Atilla Eryilmaz, Prof. Kannan Srinivasan, and Prof. Andrea Serrani for serving in my candidacy committee. Their insightful comments and suggestions have helped me significantly in working on my research for my dissertation. I would like to thank Prof. Hesham El-Gamal, Prof. Ashu Sabharwal, Prof. Xiaojun Lin, Prof. Changee Joo, Prof. Kannan Srinivasan, and Prof. Prasun Sinha for the inspiring discussions I had with them.

I would like to thank all my friends in the IPS lab for the fun we had together, and all my workout friends for the sweat we shed together. Thanks Jeri for organizing and coordinating various activities for us.
VITA

June 8th, 1987 .................. Born in Xinyang, Henan, China

2009 ......................... B.S., Electrical Engineering, Shanghai Jiao Tong University.

2014 .......................... M.S., Electrical and Computer Engineering, The Ohio State University

2009-Present .................. Graduate Research Associate, Graduate Teaching Associate, The Ohio State University

PUBLICATIONS


Fei Wu; Yin Sun; Yang Yang; Kannan Srinivasan; Ness B. Shroff “Constant Delay and Constant Feedback Moving Window Network Coding for Wireless Multicast: Design and Asymptotic Analysis,” IEEE JSAC Special Issue on Network Coding for Wireless Communications, 2015.

FIELDS OF STUDY

Major Field: Electrical and Computer Engineering

Specialization: Network Science
# TABLE OF CONTENTS

Abstract ................................................................. ii

Dedication ............................................................... iii

Acknowledgments ....................................................... v

Vita ................................................................. vi

List of Figures ........................................................ xi

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction ................................................. 1</td>
</tr>
<tr>
<td>1.1</td>
<td>Retransmission Schemes ...................................... 2</td>
</tr>
<tr>
<td>1.2</td>
<td>Coding Schemes ............................................... 3</td>
</tr>
<tr>
<td>1.3</td>
<td>Multi-antenna Techniques ................................... 4</td>
</tr>
<tr>
<td>1.4</td>
<td>Routing and Scheduling ...................................... 6</td>
</tr>
<tr>
<td>1.5</td>
<td>Contribution and Thesis Organization ...................... 7</td>
</tr>
</tbody>
</table>

| 2 | Delay Asymptotics with Retransmissions and Incremental Redundancy Codes over Erasure Channels ............................................. 12 |
| 2.1 | Introduction .................................................... 12 |
| 2.2 | System Model .................................................. 16 |
| 2.3 | Decoder that uses Memory ..................................... 20 |
| 2.3.1 | Codeword with infinite support ............................. 22 |
| 2.3.2 | Codeword with finite support ................................ 23 |
| 2.4 | Decoder that does not use Memory .......................... 25 |
| 2.4.1 | Codeword with infinite support ............................. 26 |
| 2.4.2 | Codeword with finite support ................................ 27 |
| 2.5 | Simulations .................................................... 29 |
| 2.6 | Conclusion ..................................................... 33 |

| 3 | Throughput of Rateless Codes over Broadcast Erasure Channels ............ 35 |
| 3.1 | Introduction .................................................... 35 |
### 3.1 Related Work
- Related Work: 38
- Key Contributions: 39

### 3.2 System Model
- Large Deviation Principle for finite state Markov chains: 43

### 3.3 Asymptotic Throughput
- Throughput lower bound for finite $K$ and $n$: 48

### 3.4 Throughput lower bound for finite

### 3.5 Simulation
- Simulation: 55

### 3.6 Conclusion
- Conclusion: 58

### 4 Characterizing the Achievable Throughput in Wireless Networks with Two Active RF chains
- Introduction: 60
- System Model: 63
- Constraint set: 64
- Optimal throughput region: 70
- Experimental Verification: 77
- Experimental setting: 78
- Throughput result: 80
- Conclusion: 82

### 5 Scheduling in Wireless Networks with Full-Duplex Cut-through Transmission
- Introduction: 84
- System Model: 88
- Network model and half-duplex/full-duplex constraint: 88
- Traffic model: 90
- Modeling of interference relationship with full-duplex cut-through capability: 93
- Modeling using per-flow virtual links: 94
- Modeling using per-route virtual links: 94
- Modeling using neighbor-expanded graph: 95
- Queue-length based Cut-through CSMA algorithm: 98
- Numerical Simulation: 103
- Conclusion: 104

### 6 Conclusion
- Conclusion: 106

### Appendix A: Proofs for Chapter 2
- Proof of Lemma 2.3.1: 110
- Proof of Lemma 2.3.2: 113
- Proof of Lemma 2.4.1: 114
- Proof of Lemma A.4.1: 116
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>30</td>
</tr>
<tr>
<td>2.2</td>
<td>31</td>
</tr>
<tr>
<td>2.3</td>
<td>32</td>
</tr>
<tr>
<td>3.1</td>
<td>37</td>
</tr>
<tr>
<td>3.2</td>
<td>49</td>
</tr>
<tr>
<td>3.3</td>
<td>55</td>
</tr>
<tr>
<td>3.4</td>
<td>56</td>
</tr>
<tr>
<td>3.5</td>
<td>57</td>
</tr>
<tr>
<td>4.1</td>
<td>62</td>
</tr>
</tbody>
</table>

1.1 Simple-ARQ with fixed rate erasure codes.

1.2 Wireless transmission scenarios: unicast and broadcast.

1.3 Block diagram of transceiver structures with two antennas for (a) SISO half-duplex, (b) full-duplex, (c) MIMO, and (d) MIMO plus full-duplex. If a node has the transceiver structure shown in (d), then it can enable either MIMO or full-duplex communication. In transceiver structures (b) and (d), for ease of exposition, the full-duplex cancellation circuit is drawn in analog RF end, but the cancellation can also happen in analog baseband and digital domain.

2.1 Simulation results for Example 2.5.1

2.2 Simulation results for Example 2.5.2

2.3 Simulation results for Example 2.5.3

3.1 Broadcast with discrete time queueing model

3.2 Gilbert-Elliott Erasure Channel Model

3.3 Simulation results for Example 3.5.1

3.4 Simulation results for Example 3.5.2

3.5 Simulation results for Example 3.5.3

4.1 The interference footprint of the transmission between node A and node B in a wireless network where each node has two active RF chains and can enable either full-duplex or MIMO transmission. The circle around node A/B indicates its interference range. In (a), both node A and node B are operating in full-duplex, and a bidirectional transmission is established between the two nodes; in (b), two spatially multiplexed streams are sent from node A to node B.
4.2 Different communication modes. A dashed arrow represents an interference flow. A circle on two flows indicates that the two flows are transmitted by the same transmitter node. In Single-input Single-output (SISO) modes, a user can either receive a single flow (mode 2), or transmit a single flow while withstanding any number of interference flows (mode 1); in full-duplex modes, a user can transmit one flow and receive another flow simultaneously (mode 3 and mode 4); in MIMO multiplexing modes, a user can transmit/receive two independent flows to/from another user (mode 5/mode 6); in multi-user MIMO mode, a user can transmit/receive two flows to/from two different users (mode 7/mode 8), or receive one flow while withstanding another interference flow (mode 9 and mode 10).

4.3 Throughput regions of a five-node cyclic network under SS, FD, MMa/MMb, and FMa/FMb constraint sets. Here we only focus on the data links from A to E, from D to C and from B to A. Each dot in the throughput region plot represents a feasible schedule under the corresponding constraint set. The red arrows on each network graph illustrate one such feasible schedule, while the parentheses after each node represents its communication mode, and these feasible schedules correspond to the big red dots on the throughput region plot. (a) When every node has only one active RF chain at any time, no more than one data stream in these three links can be simultaneously activated. (b) When every node has full-duplex capability, then a data stream from B to A and a data stream from A to E can be activated together. (c,d,e) Under MMa/MMb/FMa constraint set, at any time we cannot support more than 2 simultaneous data streams in the three links. (f) When every node have both multi-user MIMO and full-duplex capabilities, three data streams, one for each data link, can be activated together. The comparison between (b,d) and (f) shows that a strictly larger throughput region can be achieved when each node can choose to operate in either MIMO or full-duplex mode, than the case when only MIMO, or only full-duplex is supported.
4.4 Throughput region of a six-node cyclic network under MMa/MMb constraint set. Here we only focus on three links: B to A, F to E, and D to C. Each dot in the throughput region plot represents a feasible schedule under the corresponding constraint set. The red arrows on each network graph illustrate one such feasible schedule, with the parentheses after each node represents its communication mode, and these feasible schedules correspond to the big red dots on the throughput region plot. (a) With only MIMO multiplexing enabled, no more than two data streams can be scheduled, but with multiuser MIMO capability at each node, three data streams, one for each link, can be activated simultaneously, as can be seen in (b).

4.5 Scheduling decision of two configurations for four different constraint sets: SS, FD, MMa, and FMb.

4.6 CDFs of the network throughput w.r.t. 30 different channel conditions. (a) and (b) are the results corresponding to config 1 and 2 shown in Fig. 4.5, respectively.

5.1 Full-duplex cut-through transmission.

5.2 The interference graph $G_I$ and the data link graph $G$ of an example wireless network with five nodes. $I(B) = \{A, C, D\}$.

5.3 There are two flows that run on the network shown in Figure 5.2. The links $(A, B)$ and $(B, D)$ are already scheduled to serve flow 1. (a) Both flow 1 and flow 2 have only a single path. (b) Flow 1 has two alternative paths.

5.4 Neighbor-expanded graph of the data-link graph shown in Figure 5.2(b).

5.5 An example neighbor-expanded graph.

5.6 Simulation scenario.

5.7 Simulation results.

C.1 The summary of the indegree and outdegree of nodes in graph $G_R(S_1) = \{N_R(S_1), E_R(S_1)\}$ with different modes, for any FMb-feasible schedule $S_1$. Observe that if the indegree of a node is 1, then the node is the intended receiver of a single data flow.

C.2 Two examples of the construction of $S_1^{cycle}$ in step 2 of the proof of Theorem 4.3.3, given a network cycle with $2N_l + N_f$ nodes, where $N_l = 3$ and $N_f = 5$. The number beside each node in group X indicates its label.
CHAPTER 1
INTRODUCTION

The last decade has witnessed a proliferation of both wireless mobile devices and data-hungry applications, which has lead to an ever-increasing demand for wireless traffic. At the same time, various new wireless communication techniques have emerged across different network layers to provide more efficient use of bandwidth. These techniques include rateless codes, hybrid ARQ, multi-user MIMO, wireless full-duplex transmission, to name a few. Often, these schemes incur different hardware costs and require different levels of computational complexity. In order to exploit the maximum potential of these new techniques and strike the right balance between cost and performance in the evolution of these wireless systems, it is crucial to understand how the system throughput behaves under different wireless techniques as a function of various system settings, and develop efficient control algorithms. This is the overall goal of the dissertation.

The achievable throughput of a wireless system jointly depends upon operations across multiple communication layers. In this dissertation, we focus on four aspects of wireless operations, namely (1) retransmission schemes (2) coding schemes (3) multi-antenna techniques (4) routing and scheduling, and analyze their individual impact on throughput.
1.1 Retransmission Schemes

Retransmission is a simple and effective mechanism in combating channel uncertainties and guaranteeing link reliability, and serves as a basic building block in wireless data transmissions [1]. In traditional retransmission schemes (simple-ARQ), as is illustrated in Figure 1.1 and Figure 1.2a, a received packet is simply discarded and requested again through a Negative-ACK feedback if it cannot be decoded. Conventional belief suggests that the transmission delay caused by this operation is light-tailed (the likelihood of the delay exceeding a threshold exponentially decreases as the threshold increases), which is true when the packet error is independent of the packet size. However, under the more realistic model where the packet error probability is an increasing function of packet size, recent studies [2] [3] [4] have revealed the surprising result that simple-ARQ techniques can cause heavy-tailed transmission delay even when the packet sizes are light-tailed, resulting in very long delays and possibly zero throughput. Even when the packet sizes are upper bounded, i.e., when the packet size distribution has bounded support, [5] shows that the distribution of delay could still have a heavy-tailed main-body, and the main-body may have a dominant effect on throughput even for relatively small values of the maximum packet size.

To enhance the performance of simple-ARQ, hybrid-ARQ was rolled out in 3G systems to provide higher data rates. Hybrid-ARQ has two important distinctions over simple-ARQ (i) instead of simply discarding the undecodable packet, in hybrid-ARQ, undecodable packets are cached in the receiver memory (ii) instead of retransmitting the entire packet when a Negative-ACK-feedback is received, the transmitter sends incremental redundancy information about the packets. The incremental redundancy information is cached and combined at the receiver until the original data packet can be decoded. We thus revisit the delay analysis for the simple-ARQ scheme
and carefully investigate how the transmission-delay/link-throughput is impacted by hybrid ARQ as a function of the size of the incremental transmission and the level of redundancy in the data packet.

1.2 Coding Schemes

The selection of coding parameters plays an essential role in the throughput of wireless links. While information theory tell us that link-throughput can be made infinitesimally close to link-capacity using a capacity-achieving channel coding scheme and an infinite codeword length, in reality, we cannot have arbitrarily large codeword lengths, due to constraints on either the maximum allowable delay or on the complexity. Therefore, it is important to understand how the system throughput behaves as a function of the codeword length and channel settings.

Toward this end, in this dissertation, we consider a wireless broadcast channel
undecodable request retransmission

(a) Wireless unicast with erasure channel and variable packet length.

decoded

(b) Wireless broadcast with broadcast erasure channel and fixed packet length.

Figure 1.2: Wireless transmission scenarios: unicast and broadcast.

with broadcast traffic, as illustrated in Figure 1.2b, and focus on the use of optimal rateless codes, which are known to be capacity achieving in broadcast erasure channels. Specifically, we investigate how large the coding block size should be chosen as a function of the number of registered data-receivers and channel statistics in order to achieve a target throughput.

1.3 Multi-antenna Techniques

Multi-antenna/Multi-RF chain techniques serve as an important means to improve spectrum efficiency [6]. It has long been known that by using multiple RF chains at both the transmitter and the receiver, and transmitting independent data streams
on different antennas, MIMO can increase the spectral efficiency by a factor of the number of active RF chains.

Recent developments in wireless full-duplex technology introduce another way to utilize multiple active RF chains. It has been shown that while one RF chain is used for transmission, another one can be simultaneously used for reception. This is enabled by using advanced signal processing techniques between the transmit RF chain and receive RF chain to reduce self interference [7–9]. Since the interference from the transmitting antenna is cancelled at the receiving antenna, a twofold gain in throughput can be realized for a bidirectional flow, which is the same gain as that can be achieved by 2×2 MIMO multiplexing for a unidirectional flow.

It should be noted that different multi-RF chain techniques require different hardware configurations. When there are at least two antennas and two active RF chains, a node can have three different transceiver structures as shown in Fig. 1.3: (b) For a node to operate in full-duplex mode, it needs to have a complete set of RF chains plus a cancellation module which cancels self-interference [7–9]; (c) MIMO modes can be enabled when there are two complete sets of RF chains; (d) If a node has two complete sets of RF chains plus a self-interference cancellation module, then it can operate in either MIMO or full-duplex communication. The SISO mode shown in Fig. 1.3(a) is here just for reference.

This raises a natural question: under what conditions might a wireless system get more throughput gain from one technique over the other? While it is easy to compare the throughput gain of MIMO multiplexing and full-duplex in an elementary setting such as a link between a pair of nodes, when we view it from a general wireless network’s perspective, the answer is less clear. Nonetheless, in Chapter 4, we are able to provide a concrete answer to this fundamental question, and analytically compare the performance of different combinations of these multi-antenna techniques.
Figure 1.3: Block diagram of transceiver structures with two antennas for (a) SISO half-duplex, (b) full-duplex, (c) MIMO, and (d) MIMO plus full-duplex. If a node has the transceiver structure shown in (d), then it can enable either MIMO or full-duplex communication. In transceiver structures (b) and (d), for ease of exposition, the full-duplex cancellation circuit is drawn in analog RF end, but the cancellation can also happen in analog baseband and digital domain.

1.4 Routing and Scheduling

In a multi-hop wireless network, it is critical to have a resource allocation algorithm that efficiently utilizes the limited wireless spectrum and provides high network throughput. The seminal work of [10] develops a joint dynamic routing and scheduling algorithm, namely back-pressure, which is proven to be throughput-optimal, i.e., it can stabilize any network load that can be stabilized by some joint routing and scheduling algorithm. Later, through a utility maximization framework [11–13], it is shown that, when the flow-level throughput region can be expressed as a function of the MAC layer rate region, as is the case in conventional wireless half-duplex networks, the wireless resource allocation problem can be optimally decomposed into three parts: transport layer rate control, network layer routing, and MAC layer scheduling, with minimal coupling among the layers. With the continuous evolution
of wireless physical layer technologies, many new transmission schemes has emerged, such as wireless full-duplex, interference alignment, distributed multi-user MIMO, noisy network coding. On the one hand, these new schemes keep breaking conventional transmission constraints and expanding the rate region of wireless networks. On the other hand, they bring new interference relationship among wireless links and bring many challenges in the design of scheduling algorithms. For example, when each wireless node is capable of carrying out full-duplex wireless cut-through transmission, the flow-level throughput region cannot be directly expressed as a function of MAC-layer rate-region, creating a direct coupling between network layer routing decision and MAC-layer scheduling decision. As a result, the conventional resource allocation algorithms that are developed based on the cross-layer decoupling result [11] no longer work, and thus a completely new design in the routing/scheduling algorithm is called for.

1.5 Contribution and Thesis Organization

This dissertation focuses on characterizing the achievable throughput and designing efficient control algorithms regarding four specific network functionalities: retransmission schemes (Chapter 2), coding schemes (Chapter 2 and Chapter 3), multi-antenna techniques (Chapter 4) and routing/scheduling (Chapter 5). Specifically, in Chapter 2 we investigate how the use of incremental redundancy codes to transmit information could affect the tail of transmission-delay distribution in a point-to-point erasure channel; in Chapter 3, we look at the transmission of broadcast information using rateless codes over a broadcast erasure channel, where the broadcast-throughput is a function of channel statistics, number of receivers and coding block size, and provide both asymptotic and non-asymptotic throughput analyses; Next, in Chapter 4 and
Chapter 5, we shift our focus from charactering link-layer throughput in point-to-point/broadcast channels to characterizing network-layer throughput-region in multi-hop wireless networks; In Chapter 4, we characterize the relationship between the network throughput-regions achieved under different multi-antenna/multi-RF-chain techniques; In Chapter 5, we propose an optimal routing/scheduling algorithm for wireless networks with cut-through transmission capabilities. The contributions of this dissertation are described in more detail below.

In Chapter 2, we characterize the asymptotic delay distribution in a point-point Markovian-modulated binary erasure channel for both simple-ARQ (in which undecodable packet is simply discarded) and hybrid-ARQ (incremental redundancy transmission) with the transmitted packet being encoded using fixed-rate erasure codes. The objective is to understand how the delay-distribution is affected by (1) the level of redundancy in the data-packet (dictated by the rate of the erasure codes), (2) channel statistics and (3) the choice of different ARQ schemes. Our results show that (a) under the simple-ARQ scheme, the delay distribution follows a threshold phenomenon: when the coding rate is above the channel capacity, the delay distribution is heavy-tailed, otherwise it is light-tailed. (b) The transmission delay can be dramatically reduced and is always light-tailed by allowing the decoder to cache all undecodable packets, as is the case in hybrid ARQ. (c) While (a) and (b) assume a light-tailed (but with infinite-support) packet length distribution, when the packet length has finite support, similar to [5], we show that the delay-tail will have a main-body with a decay rate similar to that in the infinite-support packet length scenario. The result indicates that even when the packet length is upper-bounded, if simple-ARQ is used and the code-rate for the packet is mismatched with the channel capacity, the link-throughput may be very close to zero.

In Chapter 3, we consider a broadcast erasure channel, where the source node has
common information for all of the receivers. The broadcast-channel can be arbitrarily correlated in time but independent across different receivers (For each receiver, the channel model is the same as the point-to-point model assumed in Chapter 2). It is easy to realize that optimal rateless codes are capacity achieving under any finite network size. However, when the coding block size of the optimal rateless codes is finite, the achievable throughput strongly depends on the channel statistics and network size. In our analysis, we first find a close-form expression for the asymptotic throughput as the network-size approaches infinity for any arbitrary mapping $f$ from the network-size $n$ to the coding block size $K$. It turns out that the asymptotic throughput is a function of $\lim_{n \to \infty} \frac{f(n)}{\log n}$ and it can obtain any value between 0 and broadcast capacity. Next, by using the asymptotic throughput result, we are able to find a lower bound on the broadcast-throughput for any non-asymptotic cases, and the lower-bound is shown to be asymptotically-tight and significantly better than all previously known lower bounds. The result provides a guideline on how the coding block size should scale with the network size and be adjusted with the channel statistics in order to achieve a target throughput.

In Chapter 4, we consider a multi-hop wireless network where each node can simultaneously allow two of its RF-chains to be activated with no channel state information at the transmitter. There are mainly three wireless technologies that can take advantage of the simultaneous activation of multiple RF-chains: MIMO multiplexing, multi-user MIMO and wireless full-duplex. We investigate the difference in the throughput-regions achieved by any combination of these three schemes: (1) We find the if and only if condition on the network topology under which MIMO multiplexing plus multi-user MIMO has a strictly better throughput region than MIMO multiplexing itself. (2) Although the throughput region achieved by full-duplex transmission is a subset of that under MIMO transmission, it is quite surprising to find out that
when wireless nodes has the flexibility of performing either full-duplex transmission or MIMO transmission, a strictly larger throughput-region may be achieved, compared with the MIMO schemes alone, and we characterize the equivalence condition on the network topology under which the strictly-larger relationship happens. Our results provide a clear guideline on which RF-architecture and traffic pattern could result in throughput improvement for one multi-antenna technology over another.

In Chapter 5, we focus on designing routing and scheduling algorithm in multi-hop wireless networks when each node is capable of performing wireless full-duplex cut-through transmission. As will be introduced in Chapter 5, wireless full-duplex cut-through is a transmission technique where every node along the route of a multi-hop data flow can be simultaneously activated to serve that flow. Cut-through transmission can be activated when each wireless node is capable of canceling its self-interference (full-duplex capable) together with a stream of known cross-interference (by using advanced signal processing techniques). This wireless transmission scheme brings new challenges in the design of MAC layer algorithms that aim to reap its full benefit. First, the MAC layer rate region of the cut-through enabled network is directly a function of the routing decision, leading to a strong coupling between routing and scheduling. Second, it is unclear how to dynamically form/change cut-through routes based on the traffic rates and patterns. We introduce a novel method to characterize the interference relationship between links in the network with cut-through transmission, which decouples the routing decision with the scheduling decision and enables a seamless adaptation of traditional half-duplex routing/scheduling algorithms into wireless networks with full-duplex cut-through capabilities. Based on this interference model, a queue-length based CSMA-type scheduling algorithm is proposed, which both leverages the flexibility of full-duplex cut-through transmission and permits distributed implementation.
Finally, concluding remarks and possible future research directions are presented in Chapter 6.
CHAPTER 2
DELAY ASYMPTOTICS WITH RETRANSMISSIONS
AND INCREMENTAL REDUNDANCY CODES OVER
ERASURE CHANNELS

2.1 Introduction

Retransmission is the basic component used in most medium access control protocols and it is used to ensure reliable transfer of data over communication channels with failures [1]. Recent studies [2] [3] [4] have revealed the surprising result that retransmission-based protocols could cause heavy-tailed transmission delays even if the packet length is light tail distributed, resulting in very long delays and possibly zero throughput. Moreover, [5] shows that even when the packet sizes are upper bounded, the distribution of delay, although eventually light-tailed, may still have a heavy-tailed main body, and that the heavy-tailed main body could dominate even for relatively small values of the maximum packet size. In this chapter we investigate the use of coding techniques to transmit information in order to alleviate the impact of heavy tails, and substantially reduce the incurred transmission delay.

In our analysis, we focus on the Binary Erasure Channel. Erasures in communication systems can arise in different layers. At the physical layer, if the received signal falls outside acceptable bounds, it is declared as an erasure. At the data link layer, some packets may be dropped because of checksum errors. At the network
layer, packets that traverse through the network may be dropped because of buffer overflow at intermediate nodes and therefore never reach the destination. All these errors can result in erasures in the received bit stream.

In order to investigate how different transmission techniques would affect the delay distribution, we use a general coding framework called incremental redundancy codes. In this framework, each codeword is split into several pieces with equal size, which are called codeword trunks. The sender sends only one codeword trunk at a time. If the receiver cannot decode the information, it will request the sender to send another piece of the codeword trunk. Therefore, at every transmission, the receiver gains extra information, which is called incremental redundancy.

In order to combat channel erasures, we use erasure codes as channel coding to encode the information. Erasure codes represent a group of coding schemes which ensure that even when some portions of the codeword are lost, it is still possible for the receiver to recover the corresponding information. Roughly speaking, the encoder transforms a data packet of \( l \) symbols into a longer codeword of \( l_c \) symbols, where the ratio \( \beta = l/l_c \) is called the code-rate. A lower \( \beta \) indicates a larger redundancy in the codeword. An erasure code is said to be near optimal if it requires slightly more than \( l \) symbols, say \((1 + \varepsilon)l\) symbols, to recover the information, where \( \varepsilon \) can be made arbitrary small at the cost of increased encoding and decoding complexity. Many elegant low complexity erasure codes have been designed for erasure channels, e.g., Tornado Code [14], LT code [15], and Raptor code [16]. For the sake of simplicity, throughout this chapter, we assume \( \varepsilon = 0 \). In other words, any \( \beta \) fraction of the codeword can recover the corresponding information.

We specify two different scenarios in this chapter. In the first scenario, as shown in Fig. 1.1, the entire codeword is transmitted as a unit, and received bits are simply discarded if the corresponding information cannot be recovered. Note that in
this scenario, the decoder memory is not exploited for caching received bits across different transmissions. This scenario occurs in some random access protocols such as 802.11a/b [17] [18], where the base station cannot control who can transmit or retransmit a packet and when a transmission can be initiated, and may not have the requisite computation power/storage to keep track of all the undecodable packets from many different users, especially when the base station is responsible for handling a very large number of flows simultaneously. In the second scenario, we assume that the receiver has enough memory space and computational power to cache received bits from different (re)transmissions, which enables the use of incremental redundancy codes, where a codeword of length $l_c$ is split into $r$ codeword trunks with equal size, and these codeword trunks are transmitted one at a time. At the receiver, all successfully received bits from every transmission are buffered at the receiver memory according to their positions in the codeword. If the receiver cannot decode the corresponding information, it will request the sender to send another piece of codeword trunk. At the sender, these codeword trunks are transmitted in a round-robin manner. We call these two scenarios **decoder that does not use memory** and **decoder that uses memory**, respectively.

Given the above two different types of decoder, there are two more factors that can affect the distribution of delay. (I) **Channel Dynamics:** In order to capture the time correlation nature of the wireless channels, we assume that the channel is Markovian modulated. More specifically, we assume a time slotted system where one bit can be transmitted per time slot, and the current channel state distribution depends on channel states in the previous $k$ time slots. When $k = 0$, it corresponds to the *i.i.d.* channel model. (II) **Codeword length distribution:** We assume throughout this chapter that the codeword length is light tail distributed, which implies that the system works in a benign environment. We consider two different codeword length
distributions, namely, codeword length with \textit{infinite support} and codeword length with \textit{finite support}, respectively. For the former, the codeword length distribution has an exponentially decaying tail with decay rate $\lambda$, for the latter, the codeword length has an upper bound $b$.

\textbf{Contribution}

The main results of this chapter are the following:

- When decoder memory is not exploited, the tail of the delay distribution depends on the code rate. Specifically, we show that when the coding rate is above a certain threshold, the delay distribution is heavy tailed, otherwise it is light tailed. This shows that substantial gains in delay can be achieved over the standard retransmission case (no forward error correction) by adding a certain amount of redundancy in the codeword. As mentioned earlier, prior work has shown that plain ARQ without FEC results in heavy tailed delays even when the packet size are light tailed.

- When decoder memory is exploited, the tail of the delay distribution is always light-tailed. This implies that the use of receiver memory results in a substantial reduction in the transmission delay.

- The aforementioned results are for the case when the codeword size can have infinite support. We also characterize the transmission delay for each of the above cases when the codeword size has finite support (zero-tailed), and show similar impact of the coding rate and the use of receiver memory in terms of the main body of the delay distribution (rather than the eventual tail).

The remainder of this chapter is structured as follows: In Section 2.2, we describe the system model. In Section 2.3 we consider the scenario where the decoder memory
is exploited. Then, in Section 2.4 we investigate the situation where the decoder does not use memory. Finally, in Section 2.5, we provide numerical studies to verify our main results.

2.2 System Model

The channel dynamics are modeled as a slotted system where one bit can be transmitted per slot. Furthermore, we assume that the slotted channel is characterized by a binary stochastic process \( \{ X_n \}_{n \geq 1} \), where \( X_n = 1 \) corresponds to the situation when the bit transmitted at time slot \( n \) is successfully received, and \( X_n = 0 \) when the bit is erased (called an erasure).

Since, in practice, the channel dynamics are often temporarily correlated, we investigate the situation in which the current channel state distribution depends on the channel states in the preceding \( k \) time slots. More precisely, for \( \mathcal{F}_n = \{ X_i \}_{i \leq n} \) and fixed \( k \), we define \( \mathcal{H}_n = \{ X_n, \ldots, X_{n-k+1} \} \) for \( n \geq k \geq 1 \) with \( \mathcal{H}_n = \{ \emptyset, \Omega \} \) for \( k = 0 \), and assume that \( \mathbb{P}[X_n = 1|\mathcal{F}_{n-1}] = \mathbb{P}[X_n = 1|\mathcal{H}_{n-1}] \) for all \( n \geq k \). To put it another way, the augmented state \( Y_n \triangleq [X_n, \ldots, X_{n-k}], n \geq k \) forms a Markov chain. Let \( \Pi \) denote the transition matrix of the Markov chain \( \{ Y_n \}_{n \geq k+1} \), where

\[
\Pi = [\pi(s, u)]_{s,u \in \{0,1\}^k},
\]

with \( \pi(s, u) \) being the one-step transition probability from state \( s \) to state \( u \). Throughout this chapter, we assume that \( \Pi \) is irreducible and aperiodic, which ensures that this Markov chain is ergodic [19]. Therefore, for any initial value \( \mathcal{H}_k \), the parameter \( \gamma \) is well defined and given by

\[
\gamma = \lim_{n \to \infty} \mathbb{P}[X_n = 1],
\]

16
and, from ergodic theorem (see Theorem 1.10.2 in [19])

\[ \mathbb{P}\left[ \lim_{n \to \infty} \frac{\sum_{k=1}^{n} X_i}{n} = \gamma \right] = 1, \]

which means the long-term fraction of the bits that can be successfully received is equal to \( \gamma \). Therefore, we call \( \gamma \) the **channel capacity**.

In the degenerated case when \( k = 0 \), we have a memoryless binary erasure channel (i.i.d. binary erasure channel). Correspondingly, \( \mathcal{H}_n = \{\emptyset, \Omega\} \) and \( \Pi = [\gamma] \).

We focus on erasure codes with code rate \( \beta \). Furthermore, we assume that the erasure codes considered are optimal, in the sense that any \( \beta \) fraction of bits in the codeword can lead to a successful decoding. In other words, for any code rate \( \beta \) that is less than the channel capacity \( \gamma \), an arbitrarily small error probability can be achieved given a sufficiently large codeword size.

We let \( L_c \) denote the number of bits in the codeword with infinite support, and assume that there exist \( \lambda > 0 \) and \( z > 0 \) such that

\[ \lim_{x \to \infty} \frac{\log \mathbb{P}[x < L_c < x + z]}{x} = -\lambda. \] (2.2.1)

The above equation implies that the distribution of the codeword length \( L_c \) has an exponentially decaying tail with decay rate \( \lambda \). This assumption is quite general since it captures a group of distributions such as gamma distribution, hyperbolic distribution, Laplace distribution, etc.

Note that in reality, transmitted packet sizes are often bounded by the maximum transmission unit (MTU). This means that the distribution of the codeword size often has finite support.

We let \( L_c(b) \) denote codeword size with finite support, with \( b \) being the maximum codeword length, and let \( \mathbb{P}[L_c(b) > x] = \mathbb{P}[L_c > x|L_c < b] \) for any \( x > 0 \).
As mentioned in the introduction, we study two different scenarios in this chapter, namely decoder that uses memory and decoder that does not use memory. In the first scenario, the sender splits a codeword into $r$ codeword trunks with equal size and transmits them one at a time in a round-robin manner, while the receiver uses memory to cache all previously successfully received bits according to their positions in the codeword. In the second scenario, the sender transmits the entire codeword as a unit, and the receiver discards a received codeword if it cannot recover the original information.

Formal definitions of the number of retransmissions and the delays are given as follows:

**Definition 2.2.1** (Decoder that uses memory). The total number of transmissions for a codeword with variable length $L_c$ and number of codeword trunks $r$ when the decoder uses memory is defined as

$$N_m^{(r)} \triangleq \inf \left\{ n : \sum_{l=1}^{r} \sum_{i=(L_c/r)(l-1)+1}^{(L_c/r)l} 1 \left( \sum_{j=1}^{[(n-l)/r]+1} X_{(j-1)L_c+i} \geq 1 \right) > \beta L_c \right\}.$$  

The transmission delay is defined as $T_m^{(r)} = N_m^{(r)} L_c / r$.

**Definition 2.2.2** (Decoder that does not use memory). The total number of transmissions for a codeword with variable length $L_c$ when the decoder does not use memory is defined as

$$N_f \triangleq \inf \left\{ n : \sum_{i=1}^{L_c} X_{(n-1)L_c+i} > \beta L_c \right\}.$$  

The transmission delay is defined as $T_f = N_f L_c$.

For a codeword with variable length $L_c(b)$, the corresponding numbers of transmissions and delays are denoted as $N_m^{(r)}(b), T_m^{(r)}(b), N_f(b)$, and $T_f(b)$, respectively.
Notations

In order to present the main results, we introduce some necessary notations here.

**Notation 2.2.1.** Let $\rho(M)$ denote the Perron-Frobenius eigenvalue (see Theorem 3.11 in [20]) of the matrix $M$, which is the largest eigenvalue of $M$.

**Notation 2.2.2.** For $k \geq 1$, let $\{s_i\}_{1 \leq i \leq 2^k} = \{0,1\}^k$ denote the state space of $\{Y_n\}_{n \geq k+1}$, where $s_i = [s_{i1}, s_{i2}, \ldots, s_{ik}]$ and $s_{ij} \in \{0,1\}$ $\forall i, j$. Then, we define a mapping $f$ from $\{s_i\}_{1 \leq i \leq 2^k}$ to $\{0,1\}$ as

$$f(s_i) = 1 - s_{ik}.$$ 

**Notation 2.2.3.** Let $\Lambda_n(\beta, \Pi)$ denote the large deviation rate function, which is given by

$$\Lambda_n(\beta, \Pi) = \sup_{\theta} \left\{ \theta(1 - \beta) - \log \rho_n(\theta, \Pi) \right\},$$

where

$$\rho_n(\theta, \Pi) = \begin{cases} \rho \left( e^{\theta D \otimes n} \Pi \otimes n \right) & k \geq 1 \\ \rho \left( (1 - \gamma)^n + (1 - (1 - \gamma)^n) e^\theta \right) & k = 0 \end{cases},$$

$$D = \text{diag}[f(s_1), f(s_2), \ldots, f(s_{2^k})] \text{ for } k \geq 1.$$ 

**Notation 2.2.4.** Let $\mu_n$ denote the root of the rate function $\Lambda_n(\beta, \Pi)$. More precisely,

$$\Lambda_n(\mu_n, \Pi) = 0.$$ 

---

1For a matrix $A$, $A \otimes^n$ is the $n$-fold Kronecker product of $A$ with itself, or we can call it the $n^{th}$ Kronecker power of $A$. 

19
Notation 2.2.5.

\[ \alpha = \inf \{ n : \mu_n \geq \beta \} \]

\[ \Lambda_1^o = \inf_{n \in \mathbb{N}} \frac{\lambda + \Lambda_n(\beta, \Pi) \mathbf{1}(n \geq \alpha)}{n + 1} \]

\[ \Lambda_2^o = \inf_{n \in \mathbb{N}} \frac{\lambda + \Lambda_{n+1}(\beta, \Pi) \mathbf{1}(n \geq \alpha - 1)}{n + 1} \]

\[ \tilde{\Lambda}_3^o = \min_{[r\beta/\gamma] \leq m \leq r-1} \frac{\lambda + \Lambda_1 \left( \frac{r\beta}{m}, \Pi \right)}{(m + 1)/r} \]

\[ \Lambda_3^o = \begin{cases} 
\lambda & \text{if } \beta > \gamma \\
\min \left\{ \frac{\lambda r}{[r\beta/\gamma]}, \tilde{\Lambda}_3^o \right\} & \text{if } \beta \leq \gamma \end{cases} \]

2.3 Decoder that uses Memory

When the decoder uses memory to cache all previously successfully received bits, we can apply incremental redundancy codes, where the sender splits a codeword into \( r \) codeword trunks and transmits one codeword trunk at a time. If the receiver, after receiving a codeword trunk, is not able to decode the corresponding information, it will use memory to cache this codeword trunk and request the sender to send another codeword trunk. In this way, at every transmission, the receiver gains extra information, which we call incremental redundancy. The sender will send these codeword trunks in a round-robin manner, meaning that if all of the codeword trunks have been requested, it will start over again with the first codeword trunk.

Similar to the definition of the error exponent in [21], here we define the error exponent after \( n \) transmissions as

\[ - \lim_{l_c \to \infty} \frac{\mathbb{P} \left[ N_m^{(r)} > n | L_c = l_c \right]}{l_c} \]
This error exponent represents the decay rate of the decoding error probability after \( n \) (re)transmissions, as we increase the codeword length \( l_c \).

In the special case when \( r = 1 \), for every transmission, the whole codeword is transmitted as a unit. Since the decoder retains all the previously received codewords, the retransmission under \( r = 1 \) can be seen as additional repetition coding. The following lemma shows that after \( n \) (re)transmissions of a codeword with rate \( \beta \), the error exponent is \( \Lambda_n(\beta, \Pi)1(n \geq \alpha) \), where \( \Lambda_n(\beta, \Pi) \) and \( \alpha \) are defined in Notation 2.2.3 and 2.2.5.

**Lemma 2.3.1.** For any \( n \in \mathbb{N} \), we have,

\[
\mathbb{P}[N_{m}^{(1)} > n | L_c = l_c] = e^{-l_c \Lambda_n(\beta, \Pi)1(n \geq \alpha) + g_n(l_c)},
\]

where

\[
g_n(l_c) \in \begin{cases} 
  o(l_c) & \text{if } n \geq \alpha \\
  o(1) & \text{otherwise}
\end{cases}.
\]

**Proof.** see Appendix A.1.

For more general cases when \( r > 1 \), the error exponents are characterized in the following lemma.
Lemma 2.3.2. For any $n, B \in \mathbb{N}$ with $0 < B < r$, we have,

\begin{align*}
- \lim_{l_c \to \infty} \frac{\log P \left[ N_m^{(r)} > nr | L_c = l_c \right]}{l_c} &= \Lambda_n(\beta, \Pi)1(n \geq \alpha), \\
- \lim_{l_c \to \infty} \frac{\log P \left[ N_m^{(r)} > B | L_c = l_c \right]}{l_c} &= \Lambda_1 \left( \frac{\beta r}{B}, \Pi \right)1 \left( \frac{\beta r}{B} < \gamma \right), \\
- \lim_{l_c \to \infty} \frac{\log P \left[ N_m^{(r)} > nr + B | L_c = l_c \right]}{l_c} &= \min \left\{ \Lambda_{n+1} \left( \frac{z_1 r}{B}, \Pi \right)1 \left( \frac{z_1 r}{B} < \mu_{n+1} \right) + \Lambda_n \left( \frac{z_2 r}{r - B}, \Pi \right)1 \left( \frac{z_2 r}{r - B} < \mu_n \right) \right\}. \tag{2.3.1}
\end{align*}

Proof. see Appendix A.2.

\end{document}

2.3.1 Codeword with infinite support

When the distribution of the codeword length $L_c$ has an exponentially decaying tail with decay rate $\lambda$, as indicated by Equation (2.2.1), we find that the delay will always be light-tailed, and we characterize the decay rate in Theorem 2.3.1.

Theorem 2.3.1. In the case when the decoder uses memory, when we apply an incremental redundancy code with parameter $r$ to transmit a codeword with variable length $L_c$, we obtain a lower and upper bound on the decay rate of the delay,

\begin{align*}
- \lim_{t \to \infty} \frac{\log P \left[ T_m^{(r)} > t \right]}{t} &\leq \min \{ \Lambda_2^o, \Lambda_3^o \}, \\
- \lim_{t \to \infty} \frac{\log P \left[ T_m^{(r)} > t \right]}{t} &\geq \min \{ \Lambda_1^o, \Lambda_3^o \}.
\end{align*}
In the special case when \( r = 1 \),

\[
- \lim_{n \to \infty} \log \mathbb{P} \left[ \frac{T_m^{(1)} > t}{t} \right] = \min \{ \Lambda_1^o, \lambda \}.
\]

The definitions of \( \Lambda_1^o, \Lambda_2^o \) and \( \Lambda_3^o \) can be found in Notation 2.2.5.

Proof. see Appendix A.5. \qed

**Remark 2.3.1.1.** Although it is possible to characterize the exact decay rate of the delay for any \( r \), from Lemma 2.3.2 we know that the expression of that decay rate contains a combinatorial optimization problem (as can be seen from Equation (2.3.1)) which is hard to evaluate and does not give useful insights. Therefore, in Theorem 2.3.1, instead of giving the exact decay rate of the delay, we obtain an upper bound and an lower bound on the delay rate, which are much easier to evaluate.

**Remark 2.3.1.2.** From the definitions of \( \Lambda_1^o, \Lambda_2^o \) and \( \Lambda_3^o \) in Notation 2.2.5 we observe that firstly, the decay rate of the delay when \( r = 1 \) is no greater than the decay rate of the delay when \( r > 1 \) (\( \min \{ \Lambda_1^o, \lambda \} \leq \min \{ \Lambda_1^o, \Lambda_3^o \} \)), which means that incremental redundancy codes (\( r > 1 \)) outperform fixed rate erasure codes (\( r = 1 \)); secondly, the decay rate of the delay increases with the increase of \( r \), which means we can reduce delay by increasing the number of codeword trunks \( r \); thirdly, when \( \Lambda_1^o > \Lambda_3^o \) and \( \Lambda_2^o > \Lambda_3^o \), the two bounds on the decay rate of the delay coincide. These observations are verified through Example 1 in Section 2.5.

### 2.3.2 Codeword with finite support

In practice, codeword length is bounded by the maximum transmission unit (MTU). Therefore, we investigate the case when the codeword has variable length \( L_c(b) \), with \( b \) being the maximum codeword length, and characterize the corresponding delay distribution in Theorem 2.3.2.
Theorem 2.3.2. In the case when the decoder uses memory, when we apply an incremental redundancy code with parameter $r$ to transmit a codeword with variable length $L_c(b)$, for any $\eta, \delta > 0$, we can find $b(\eta, \delta) > 0$ such that for any $b > b(\eta, \delta)$, we have

1) $\forall t \in [n_2^\circ(b - \delta), n_2^\circ b]$, \[
(1 - \eta)\Lambda_1^b \leq -\frac{\log \mathbb{P}\left[T_m^{(r)}(b) > t\right]}{t} \leq (1 + \eta)\Lambda_2^b.
\]

2) in the special case when $r = 1$, $\forall t \in [n_1^\circ(b - \delta), n_1^\circ b]$, \[
1 - \eta \leq -\frac{\log \mathbb{P}\left[T_m^{(l)}(b) > t\right]}{t} \frac{1}{\Lambda^b} \leq 1 + \eta,
\]

where

\[
n_1^\circ = \arg \inf_{n \in \mathbb{N}} \frac{(\lambda + \Lambda_n(\beta, \Pi)1(n \geq \alpha))}{(n + 1)},
\]

\[
n_2^\circ = \arg \inf_{n \in \mathbb{N}} \frac{(\lambda + \Lambda_{n+1}(\beta, \Pi)1(n \geq \alpha - 1))}{(n + 1)},
\]

\[
\Lambda_1^b = \Lambda_1^\circ + \min\{0, \Lambda_3^\circ - \Lambda_1^\circ\}1(n_2^\circ = 1),
\]

\[
\Lambda_2^b = \Lambda_2^\circ + \min\{0, \Lambda_3^\circ - \Lambda_2^\circ\}1(n_2^\circ = 1),
\]

\[
\Lambda^b = \Lambda_1^\circ + \min\{0, \lambda - \Lambda_1^\circ\}1(n_1^\circ = 1).
\]

Proof. see Appendix A.6. \qed

Remark 2.3.2.1. This theorem shows that even if the codeword length has an upper bound $b$, the distribution of the delay still has a light-tailed main body whose decay rate is similar to the decay rate under the infinite support scenario. The waist of this main body is at least $n_2^\circ b$ when $r > 1$ and $n_1^\circ b$ when $r = 1$. Since both $n_2^\circ$ and $n_1^\circ$ are independent of $b$, we know that the waist of this light-tailed main body scales linearly
with respect to the maximum codeword length $b$. This theorem is verified through Example 2 in Section 2.5.

### 2.4 Decoder that does not use Memory

There are a number of systems that use hybrid ARQ and perform soft combing to improve the probability of successful decoding. However, there are also several systems that use plain ARQ. For example, in the IEEE 802.11a/b [17] [18] protocol family, whenever an undecodable packet arrives, the receiver simply discards it and asks for a retransmission by sending a feedback message. The reason is that 802.11 uses a random access MAC layer protocol and thus the base station has no control of exactly who can use the channel to transmit or retransmit a packet and when a transmission can be initiated. In other words, if the decoder at the 802.11 base station were to use hybrid ARQ to combine different transmissions, it would have to keep track of which packets came from which users in the physical layer all the time in order to combine the right packets. Hence, a lot of decoder memory would have to be allocated to store packets from many different users, especially when the number of users is large.

Therefore, in this section, we study the case when the decoder does not use memory, as illustrated in Fig. 1.1. Since the decoder simply discards a received packet after a failed decoding attempt, we do not need to consider the case when a codeword is divided into several codeword trunks before transmission, because a codeword trunk will be discarded if itself cannot recover the original information. For example, if we divide a codeword with rate $\beta$ into two codeword trunks and transmit them one after the other, then the transmission is successful only when either one of codeword trunks itself can be decoded. Therefore, it will be equivalent to transmitting an entire codeword with rate $2\beta$. 

25
The following lemma shows that when the code rate $\beta$ is less than the channel capacity $\gamma$, after $n$ transmissions, the error exponent is $n\Lambda_1(\beta, \Pi)$.

**Lemma 2.4.1.**  
1. if $\beta > \gamma$, then

$$
\mathbb{P} [N_f > n | L_c = l_c] = \left(1 - e^{-l_c\Lambda_1(\beta, \Pi)(1+g(l_c))}\right)^n,
$$

where $g(l_c) \in o(1)$ as $l_c \to \infty$.

2. if $\beta < \gamma$, then

$$
\mathbb{P} [N_f > n | L_c = l_c] = e^{-nl_c\Lambda_1(\beta, \Pi)(1+s(l_c))},
$$

where $s(l_c) \in o(1)$ as $l_c \to \infty$.

**Proof.** see Appendix A.3. $\square$

### 2.4.1 Codeword with infinite support

Interestingly, we observe an intriguing threshold phenomenon. We show that when the codeword length distribution is light-tailed and has an infinite support, the transmission delay is light-tailed (exponential) only if $\gamma > \beta$, and heavy-tailed (power law) if $\gamma < \beta$.

**Theorem 2.4.1** (Threshold phenomenon). In the case when the decoder does not use memory and the codeword has variable length $L_c$, we get

1. if $\beta > \gamma$, then

$$
\lim_{n \to \infty} \frac{\log \mathbb{P} [N_f > n]}{\log n} = \lim_{t \to \infty} \frac{\log \mathbb{P} [T_f > t]}{\log t} = -\frac{\lambda}{\Lambda_1(\beta, \Pi)}.
$$
2. if $\beta < \gamma$, then

$$
\lim_{t \to \infty} \frac{\log \mathbb{P}[T_f > t]}{t} = -\min \{\lambda, \Lambda_1(\beta, \Pi)\}.
$$

The definition of $\Lambda_1(\beta, \Pi)$ can be found in Notation 2.2.3.

Proof. see Appendix A.7. \qed

Remark 2.4.1.1. The tail distribution of the transmission delay can follow either power law or exponential distribution, depending on the relationship between code-rate $\beta$ and channel capacity $\gamma$. When $\beta > \gamma$ and $\lambda/\Lambda_1(\beta, \Pi) < 1$, the system even has a zero throughput.

2.4.2 Codeword with finite support

Under the heavy-tailed delay case when $\beta > \gamma$, we can further show that if the codeword length is upper bounded, the delay distribution still has a heavy-tailed main body, although it eventually becomes light-tailed.

Theorem 2.4.2. In the case when the decoder does not use memory and the codeword has variable length $L_c(b)$, if $\beta > \gamma$, for any $\eta > 0$, we can find $n(\eta) > 0$ and $b(\eta) > 0$ such that for any $b > b(\eta)$ we have, 1) $\forall n \in [n(\eta), n_b]$ and $\forall t \in [n(\eta)b, n_bb]$, 

$$
1 - \eta \leq -\frac{\log \mathbb{P}[N_f(b) > n]}{\log n} \frac{\Lambda_1(\beta, \Pi)}{\lambda} \leq 1 + \eta,
$$

$$
1 - \eta \leq -\frac{\log \mathbb{P}[T_f(b) > t]}{\log t} \frac{\Lambda_1(\beta, \Pi)}{\lambda} \leq 1 + \eta.
$$

where

$$
n_b = (\mathbb{P}[N_f = 1|L_c = b])^{-1},
$$

(2.4.1)
and the definition of $\Lambda_1(\beta, \Pi)$ can be found in Notation 2.2.3. 2)

\[
\lim_{n \to \infty} \frac{\log \mathbb{P}[N_f(b) > n]}{n} = \log (\mathbb{P}[N_f > 1|L_c = b]),
\]
\[
\lim_{n \to \infty} \frac{\log \mathbb{P}[T_f(b) > t]}{t} = \frac{1}{b} \log (\mathbb{P}[T_f > 1|L_c = b]) .
\]

Proof. see Appendix A.8. \qed

Remark 2.4.2.1. From Equation (2.4.1) and by Lemma 2.4.1, we can obtain

\[
\lim_{b \to \infty} \frac{\log n_b}{b} = \Lambda_1(\beta, \Pi),
\] (2.4.2)

which implies that $n_b$ increases exponentially fast with the increase of the maximum codeword length $b$. Since the waist of the heavy-tailed main body of the delay distribution is $n_b b$, we know that the waist also increases exponentially fast as we increase the maximum codeword length $b$.

From Theorem 2.4.2 we know that even if the codeword length is bounded, the heavy-tailed main body could still play a dominant role. From Theorem 2.4.1 we know that when $\lambda < \Lambda_1(\beta, \Pi)$ and $\beta > \gamma$, the throughput will vanish to zero as $b$ approaches infinity. Now we explore how fast the throughput vanishes to zero as $b$ increases.

Let $\{L_i\}_{i \geq 1}$ be the i.i.d. sequence of codeword lengths with distribution $L_c(b)$. Denote $T_i$ as the transmission delay of $L_i$. The throughput of this system is defined as $\Delta(b) = \lim \sum_{i=1}^{n} \beta L_i / \sum_{i=1}^{n} T_i$.

**Theorem 2.4.3 (Throughput).** In the case when the decoder does not use memory and the codeword has variable length $L_c(b)$, if $\beta > \gamma$ and $\lambda < \Lambda_1(\beta, \Pi)$, we have

\[
-\limsup_{b \to \infty} \frac{\log \Delta(b)}{b} \geq \Lambda_1(\beta, \Pi) - \lambda.
\]

28
The definition of $\Lambda_1(\beta, \Pi)$ can be found in Notation 2.2.3.

Proof. see Appendix A.9. \qed

Remark 2.4.3.1. Theorem 2.4.3 indicates that when code-rate $\beta$ is greater than channel capacity $\gamma$ and $\lambda < \Lambda_1(\beta, \Pi)$, as the maximum codeword length $b$ increases, the throughput vanishes to 0 at least exponentially fast with rate $\Lambda_1(\beta, \Pi) - \lambda$.

2.5 Simulations

In this section, we conduct simulations to verify our main results. As is evident from the following figures, the simulations match theoretical results well.

Example 2.5.1. In this example, we study the case when the decoder uses memory and the codeword length has infinite support. We assume that the channel is i.i.d. ($k = 0$). As shown in Theorem 2.3.1, under the above assumptions, the delay distribution is always light-tailed. In order to verify this result, we assume that $L_c$ is geometrically distributed with mean 100 ($\lambda = 0.01$), and choose code-rate $\beta = 0.25$ and channel capacity $\gamma = 0.50$. By Theorem 2.3.1 we know that when $r = 1$, the decay rate of the delay is $\min\{\Lambda_1^0, \lambda\} = 0.0025$; when $r = 3$, the decay rate of the delay is $\min\{\Lambda_1^0, \Lambda_3^0\} = \min\{\Lambda_2^0, \Lambda_3^0\} = 0.0037$; when $r = 5$, the decay rate of the delay is $\min\{\Lambda_1^0, \Lambda_5^0\} = \min\{\Lambda_2^0, \Lambda_3^0\} = 0.0042$. From Fig. 2.1 we can see that the decay rate of the delay increases when $r$ increases from 1 to 5, and the theoretical result is quite accurate.

This example shows that when the code rate is less than the channel capacity, the delay distribution can benefit from an increased rate of feedback. In other words, by increasing the number of codeword trunks $r$, the tail of delay distribution can become lighter.
Example 2.5.2. In this simulation, we study the case when the decoder uses memory and the codeword length has a finite support. We assume that the channel is i.i.d. ($k = 0$), code-rate $\beta = 0.75$, $\lambda = 0.01$, $r = 1$, and channel capacity $\gamma = 0.1$. From these system parameters we can calculate $n_1^* = 14$ and $\Lambda^b = \min\{\Lambda_1^*, \lambda\} = 7.1429 \times 10^{-4}$. We choose four sets of maximum codeword length $b$ as 200, 400, 600, 800. Theorem 2.3.2 indicates that the delay distribution has a light-tailed main body with decay rate $\Lambda^b = 7.1429 \times 10^{-4}$ and waist $n_6^* b = 14 \times b$. In Fig. 2.2 we plot the delay distributions when $b = 200, 400, 600, 800$ together with the infinite support case when $b = \infty$, and we use a short solid line to indicate the waist of the light-tailed main body. As we can see from Fig. 2.2, the theoretical waists of the main bodies, which are $n_6^* b = 14 \times b = 2800, 5600, 8400, 11200$, are close to the simulation results.
Example 2.5.3. Now we use simulations to verify Theorem 2.4.2. Theorem 2.4.2 says that when the decoder does not use memory, if code rate $\beta$ is greater than channel capacity $\gamma$ and the codeword length has a finite support, the distribution of delay as well as the distribution of number of retransmissions have a heavy-tailed main body and an exponential tail. The waist of the main body increases exponentially fast with the increase of maximum codeword length $b$. In this experiment, we set code-rate $\beta = 0.25$, channel capacity $\gamma = 0.20$, $k = 0$, and $\lambda = 0.01$. From these parameters we can get $\Lambda_1(\beta, \Pi) = 0.0074$. We choose four sets of maximum codeword length $b$ as 200, 400, 600, 800. As Equation (2.4.2) indicates, the waist of the heavy-tailed main bodies of the number of retransmissions is $n_b \approx e^{b \Lambda_1(\beta, \Pi)} = 4.3772, 19.1595, 83.8641, 367.0865$. In Fig. 2.3, we plot the distribution of the number of retransmissions when $b = 200, 400, 600, 800$ together with the infinite support case when $b = \infty$, and we use a short solid line to indicate the waist of the heavy-tailed
main body. As can be seen from Fig. 2.3, the simulation matches with our theoretical result.

The above two examples show that when the decoder does not use memory, the choice of the code rate has a dramatic effect on the delay distribution, even if the codeword length is upper-bounded. More specifically, as we increase the code rate, the main body of the tail of the distribution of the delay could change from an exponential decay function which spans just proportionally to the maximum codeword size, to a much slower power-law decay function which spans exponentially to the maximum codeword size. This emphasizes the importance of rate adaptation when the decoder does not use memory.

Figure 2.3: Simulation results for Example 2.5.3
2.6 Conclusion

In this chapter, we characterize the delay distribution in a point-to-point Markovian modulated binary erasure channel with variable codeword length. Erasure codes are used to encode information such that a fixed fraction of bits in the codeword can recover the information. We use a general coding framework called incremental redundancy codes. In this framework, the codeword is divided into several codeword trunks and these codeword trunks are transmitted one at a time to the receiver. Therefore, after each transmission, the receiver gains extra information, which is called incremental redundancy. At the receiver end, we investigate two different scenarios, namely decoder that uses memory and decoder that does not use memory.

In the decoder that uses memory case, the decoder caches all previously successfully transmitted bits. In the decoder that does not use memory case, received bits are discarded if the corresponding information cannot be decoded. In both cases, we first assume that the distribution of the codeword length is light-tailed and has an infinite support. Then, we consider a more realistic case when the codeword length is upper bounded.

Our results show the following. The transmission delay can be dramatically reduced by allowing the decoder to use memory. This is true because the delay is always light-tailed when the decoder uses memory, while the delay can be heavy-tailed when the decoder does not use memory. Secondly, analogously to the non-coding case, the tail effect of the delay distribution persists even if the codeword length has a finite support. When the codeword length is upper bounded, the light-tailed delay distribution will turn into a delay distribution with light-tailed main body whose decay rate is similar to that in the infinite support scenario. Further, we show that the waist of this main body scales linearly with respect to the increase of the maximum
codeword length; the heavy-tailed delay distribution will turn into a delay distribution with heavy-tailed main body, whose waist scales exponentially with the increase of the maximum codeword length. Our results provide a benchmark for quantifying the tradeoff between system complexity (which is determined by code-rate $\beta$, number of codeword trunks $r$, maximum codeword length $b$ and whether to use memory at the receiver or not) and the distribution of the delay.
3.1 Introduction

In this chapter, we study the throughput of a wireless broadcast network with \( n \) receivers using rateless codes. In this broadcast network, channels between the transmitter and the receivers are modeled as packet erasure channels where transmitted packets may either be erased or successfully received. This model describes a situation where packets may get lost or are not decodable at the receiver due to a variety of factors such as channel fading, interference or checksum errors. We assume that the underlying channel is a Markov modulated packet erasure channel that is \( i.i.d. \) across users, but can be correlated in time. We let \( \gamma \) denote the steady state probability that a packet is transmitted successfully on the erasure channel.

Instead of transmitting the broadcast data packet one after another through feedback and retransmissions, we investigate a class of coding schemes called rateless codes (or fountain codes). In this coding scheme, \( K \) broadcast packets are encoded together prior to transmission. \( K \) is called the coding block size. A rateless encoder views these \( K \) packets as \( K \) input symbols and can generate an arbitrary number of output symbols (which we call coded packets) as needed until the coding block is decoded. Although some coded packets may get lost during the transmission, rateless
decoder can guarantee that any $K(1 + \varepsilon)$ coded packets can recover the original $K$ packets with high probability, where $\varepsilon$ is a positive number that can be made arbitrarily small at the cost of coding complexity. Examples of rateless erasure codes include Raptor codes [16], LT Codes [15] and random linear codes [22], where the former two are more computationally efficient when $K$ is very large and random linear codes have less communication overhead when $K$ is relatively small and the field size of packets is large. The best encoding and decoding complexity of rateless codes (e.g. Raptor codes) increase linearly as the coding block size $K$ increases. Further, increasing the coding block size can result in large delays and large receiver buffer size. Therefore, real systems always have an upper bound on the value of $K$.

We consider broadcast traffic and a discrete time queueing model, where the numbers of packet arrivals over different time slots are independent and identically distributed and the packet length is a fixed value. We let $\lambda$ denote the packet arrival rate and assume that the encoder waits until there are at least $K$ packets in the queue and then encodes the first $K$ of them as a single coding block. In this case, the largest arrival rate that can be stabilized is equal to the average number of packets that can be transmitted per slot, which we call the throughput. Therefore, we only need to characterize the throughput that can be achieved using rateless codes under parameters $K$ and $n$. As described in Figure 3.1, the channel dynamics for the $i^{th}$ receiver is denoted by a stochastic process $\{X_{ij}\}_{j \in \mathbb{N}}$, where $j$ is the index of the time slot in which one packet can be transmitted and $X_{ij}$ is the channel state of $i^{th}$ receiver during the transmission of the $j^{th}$ packet. We capture a fairly general correlation structure by letting the current channel state be impacted by the channel states in previous $l$ time slots, where $l$ can be any number. As the number of receivers $n$ approaches
infinity, we show that the throughput is nonzero only if the coding block size \( K \) increases at least as fast as \( \log n \). In other words, if 
\[
c \triangleq \lim_{n \to \infty} \frac{K}{\log n},
\]
the asymptotic\(^1\) throughput is positive whenever \( c > 0 \). In Theorem 3.3.1, by utilizing large deviation techniques, we give an explicit expression for the asymptotic throughput, which is a function of \( K, n, \gamma \) and the channel correlation structure.

![Figure 3.1: Broadcast with discrete time queueing model](image)

To study the non-asymptotic behavior of the system, we make a more restrictive channel assumption that the current channel state is impacted by only the channel state in previous 1 time slot, which is the so called Gilbert-Elliott erasure channel model. In this case, for any finite \( K \) and \( n \), we find a lower bound on the throughput in terms of the transmission time of a system with larger \( K \) and \( n \). As a special case when

\(^1\)The asymptotic is with respect to increasing the number of receivers \( n \)
the channels are memoryless, if $\frac{K}{\log n}$ is kept constant, this lower bound reveals that the throughput will follow a decreasing pattern as the number of receivers $n$ increases. By combining this result with the characterization of the asymptotic throughput, we are able to provide a lower bound on the maximum achievable throughput for any finite values of $K$ and $n$. This lower bound captures the asymptotic throughput in the sense that when $n$ approaches infinity, it coincides with the asymptotic throughput.

3.1.1 Related Work

Among the works that investigate the throughput over erasure channels, [23], [24], [25] and [26] are the most relevant to this chapter. In [25], the authors investigate the asymptotic throughput as a function of $n$ and $K$ and also show that the asymptotic throughput will be non-zero only if $K$ at least scales with $\log n$. However, they only consider the channel correlation model with $l = 1$ and use a completely different proof technique. Moreover, no explicit expression on the asymptotic throughput is provided. In [23] and [24], two closed-form lower bounds on the maximum achievable rate $\lambda$ are provided. However, their bound does not converge to the asymptotic throughput when $n$ approaches infinity. Moreover, our bound is shown to be better in a variety of simulation settings with finite $K$ and $n$, as will be showed in Section 3.5. In [26], the authors consider the case when instantaneous feedback is provided from every user after the transmission of each decoded packets, while we only assume that feedback is provided after the entire coding block has been decoded. In [27], the authors derived the capacity of general wireless erasure networks, while we only focus on a broadcast erasure network with no intermediate nodes, where the capacity can be approached by using rateless codes with $K$ approaches infinity.
3.1.2 Key Contributions

The main contributions of this chapter are summarized as follows:

- We give an explicit expression for the asymptotic throughput of the system when the number of receivers $n$ approaches infinity for any values of $K$ as a function of $n$ under the erasure channel with any levels of correlation. (Theorem 3.3.1)

- Under the Gilbert-Elliott erasure channel model ($l = 1$), for any finite $K$ and $n$, we find a lower bound on the throughput in terms of the transmission time of a system with larger $K$ and $n$. As a special case, when channels are memoryless ($l = 0$), this lower bound reveals that when $K$ grows with $n$ in a way that the ratio $\frac{K}{\log n}$ is kept constant, the throughput follows a decreasing pattern as $n$ increases. (Theorem 3.4.1)

- We provide an asymptotically tight lower bound on the maximum achievable throughput for any values of $K$ and $n$ under the Gilbert-Elliott erasure channel model ($l = 1$) and show that its performance is significantly better than the previously known bounds in [23] and [24]. (Theorem 3.4.2)

The rest of this chapter is organized as follows. In Section 3.2 we describe our model and assumptions. In Section 3.3 we give the characterization of the asymptotic throughput. In Section 3.4 we provide a lower bound on the maximum achievable throughput for any finite values of $K$ and $n$. In Section 3.5 we use simulations to verify our theoretical results. Finally, in Section 3.6, we conclude the chapter. Detailed proofs on all the theorems can be found in Appendix B.
3.2 System Model

We consider a broadcast channel with \( n \) receivers. Time is slotted, and the numbers of broadcast packet arrivals over different time slots are i.i.d. with finite variance. We denote the expected number of packet arrivals per slot as the packet arrival rate \( \lambda \). The transmission starts when there are more than \( K \) packets waiting in the incoming queue intended for all the receivers. Instead of transmitting these packets one after another using feedback and retransmissions, we view each data packet as a symbol and encode the first \( K \) of them into an arbitrary number of coded symbols as needed using rateless codes (For example, Raptor codes [16] or random linear codes [22]) until the coding block is decoded. These \( K \) packets together form a single coding block with \( K \) being called block size. During the transmission, the coded symbols are transmitted one after another.

Each receiver sends an ACK feedback signal after it has successfully decoded the \( K \) packets. We assume that the ACK signal is transmitted instantaneously and received without error. In the following context, the term packet and symbol are used interchangeably.

We model the broadcast channel as a slotted broadcast packet erasure channel where one packet can be transmitted per slot. The channel dynamics can be represented by a stochastic process \( \{ X_{ij} \}_{1 \leq i \leq n, j \in \mathbb{N}} \), where \( X_{ij} \) is the state of channel between transmitter and the \( i \)th receiver during the transmission of \( j \)th packet (we also call it the \( j \)th time slot in the \( i \)th channel), which is given by

\[
X_{ij} = \begin{cases} 
1 & \text{\( j \)th packet in the \( i \)th channel is successfully received,} \\
0 & \text{otherwise.}
\end{cases}
\]

We assume that the dynamics of the channels for different receivers are independent.
and identical. More precisely, for all \(1 \leq i \leq n\), \(\{X_{ij}\}_{j \geq 1}\) are independent and identical processes.

Since, in practice, the channel dynamics are often temporarily correlated, we investigate the situation where the current channel state distribution depends on the channel states in the preceding \(l\) time slots. More specifically, for \(\mathcal{F}_{im} = \{X_{ij}\}_{j \leq m}\) and fixed \(l\), we define \(\mathcal{H}_{im} = \{X_{im}, \ldots, X_{i(m-l+1)}\}\) for \(m \geq l \geq 1\) with \(\mathcal{H}_{im} = \{\emptyset, \Omega\}\) for \(l = 0\), and assume that \(\mathbb{P}[X_{i(m+1)} = 1|\mathcal{F}_{im}] = \mathbb{P}[X_{i(m+1)} = 1|\mathcal{H}_{im}]\) for all \(m \geq l\). To put it another way, when \(l \geq 1\), the state \((X_{im}, \ldots, X_{i(m-l+1)})\), \(m \geq l\), forms a Markov chain. Denote by \(\Pi\) the transition matrix of the Markov chain \(\{(X_{im}, \ldots, X_{i(m-l+1)})\}_{m \geq l}\), where

\[
\Pi = [\pi(s, u)]_{s, u \in \{0, 1\}^l},
\]

with \(\pi(s, u)\) being the one-step transition probability from state \(s\) to state \(u\). Throughout this chapter, we assume that \(\Pi\) is irreducible and aperiodic, which ensures that this Markov chain is ergodic [19]. Therefore, for any initial value \(\mathcal{H}_l\), the parameter \(\gamma_i\) is well defined and given by

\[
\gamma_i = \lim_{m \to \infty} \mathbb{P}[X_{im} = 1],
\]

and, from the ergodic theorem [19] we know

\[
\mathbb{P}\left[\lim_{m \to \infty} \frac{\sum_{j=1}^{m} X_{ij}}{m} = \gamma_i\right] = 1. \quad (3.2.1)
\]

Since \(\{X_{ij}\}_{j \geq 1}\) for all \(1 \leq i \leq n\) are i.i.d., we denote \(\gamma = \gamma_i\), for all \(1 \leq i \leq n\).

Using near optimal rateless codes, such as Raptor Codes [16], LT Codes [15] and random linear codes [22], only slightly more than \(K\) coded symbols are needed to
decode the whole coding block. For simplicity, here we assume that any combination of $K$ coded symbols can lead to a successful decoding of the $K$ packets.

According to the above system model, we have the following definitions:

**Definition 3.2.1.** The number of time slots (number of transmitted coded symbols) needed for user $i$ to successfully decode $K$ packets is defined as

$$T_i(K) = \min \left\{ m \left| \sum_{j=1}^{m} X_{ij} \geq K \right. \right\}.$$

**Definition 3.2.2.** The number of time slots (number of transmitted coded symbols) needed to complete the transmission of a single coding block to all the receivers is defined as

$$T(n, K) = \max \{ T_i(K), i = 1, 2, \ldots, n \}.$$

**Definition 3.2.3 (Initial State).** Since the current channel state depends on the channel states in the previous $l$ time-slots, for each receiver $i$, by assuming that the system starts at time slot 1, we define the initial state of receiver $i$ as

$$\mathcal{E}_i = [X_{i(-l+1)}, X_{i(-l+2)}, \ldots, X_{i0}] \in \{0, 1\}^l.$$

The initial state for all the receivers is then denoted as $\mathcal{E} \triangleq [\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n]$.

**Definition 3.2.4 (Throughput).** For a system with an infinite backlog of packets, we define throughput $\eta(n, K)$ as the long term average number of packets that can be transmitted per slot. More precisely,

$$\eta(n, K) = K \times \lim_{t \to \infty} \frac{R(t)}{t},$$
where $R(t)$ is the number of successfully transmitted coding blocks in $t$ time slots. For any finite values of $K$ and $n$, it is easy to check that $\{E^h, T^h(n, K)\}_h$ is a finite-state ergodic Markov renewal process, where $E^h$ and $T^h(n, K)$ denote the initial state and the transmission time of the $h$th coding block, respectively. Then, $R(t)$ is the total number of state transitions that occur in $t$ time slots of this Markov renewal process. Therefore, from [28] we know that

$$\eta(n, K) \overset{a.s.}{=} \frac{K}{\mathbb{E}[T(n, K)]} = \frac{K}{\mathbb{E}[\mathbb{E}[T(n, K)|E]]},$$

(3.2.2)

where the outer expectation in the last term denote the expectation with respect to the steady state distribution of the embedded Markov chain $\{E^h\}_h$.

3.2.1 Large Deviation Principle for finite state Markov chains

We use the following result from Large Deviation Theory [20] in our proofs. Let $\{Y_i\}$ be a finite state Markov chain possessing an irreducible transition matrix $\Phi$, and let $f(.)$ be a deterministic function that maps the state space of $\{Y_i\}$ into $\mathbb{R}$. For every $\beta \in \mathbb{R}$, define

$$I(\beta) \triangleq \sup_{\theta \in \mathbb{R}} \{\theta \beta - \log \rho(\Phi_{\theta})\},$$

where $\Phi_{\theta}(A, B) = \Phi(A, B)e^{\theta f(B)}$ with $A$ and $B$ being two states in $\{Y_k\}$, and $\rho(\Phi_{\theta})$ denotes the Perron-Frobenious eigenvalue of $\Phi_{\theta}$ (See Theorem 3.1.1 in [20]), which is the largest eigenvalue of $\Phi_{\theta}$. Here $I(.)$ is called a large deviation rate function. It is a non-negative convex function that attains its minimum in any closed set (See Theorem 3.1.2 in [20]).
Then from Theorem 3.1.2 in [20], for any closed set $\Gamma \subseteq \mathbb{R}$, and any initial state $Y_0$,

$$\lim_{m \to \infty} \frac{1}{m} \log P \left[ \sum_{i=1}^{m} f(Y_i) m \in \Gamma \right] = -\inf_{\beta \in \Gamma} I(\beta).$$

### 3.3 Asymptotic Throughput

Before presenting the main results, we need to introduce some necessary definitions. First, define a mapping $f$ from the state space of the Markov chain $\{0, 1\}^l$ to $\{0, 1\}$ as

$$f((X_{im}, \ldots, X_{i(m-l+1)})) = X_{im}.$$ 

In the degenerate case when $l = 0$, simply let $f(X_{ij}) = X_{ij}$. Then, given a real number $\theta$, we define a matrix $\Pi_\theta$ as

$$\Pi_\theta = \begin{cases} 
\left[ \pi(s, u) e^{\theta f(u)} \right]_{s, u \in \{0, 1\}^l} & \text{when } l \geq 1 \\
\left[ \pi(s, u) e^{\theta f(u)} \right]_{s, u \in \{0, 1\}} & \text{when } l = 0
\end{cases}.$$ 

Last, define a standard large deviation rate function $\Lambda(\beta, \Pi)$ as

$$\Lambda(\beta, \Pi) = \sup_{\theta} \{ \theta \beta - \log \rho(\Pi_\theta) \}. \tag{3.3.1}$$

Then from Section 3.2.1 we know that

$$\lim_{m \to \infty} \frac{1}{m} \log P \left[ \sum_{j=1}^{m} X_{ij} \frac{m}{m} \in \Gamma \right] = -\inf_{\beta \in \Gamma} \Lambda(\beta, \Pi). \tag{3.3.2}$$
for any closed set $\Gamma \subseteq \mathbb{R}$ and any initial state $\mathcal{E}_i$. By letting $\Gamma = [\gamma]$ and combining Equation (3.2.1), we know that $\Lambda(\gamma, \Pi) = 0$ and $\Lambda(\beta, \Pi) > 0$ for any $\beta \neq \gamma$. Also, since $\mathbb{P}[^{\sum_{j=1}^{m}X_{ij}}{m} \leq \beta]$ is an increasing function of $\beta$ in $[0, 1]$ and becomes 0 only if $\beta < 0$, we know that $\inf_{\lambda \in [-\infty, \beta]} \Lambda(\lambda, \Pi)$ is a finite decreasing function of $\beta$ in $[0, 1]$ and only becomes infinity for $\beta < 0$, which, by combining the fact that $\Lambda(\beta, \Pi)$ is a convex function of $\beta$ and $\Lambda(\gamma, \Pi) = 0$, implies that $\Lambda(\beta, \Pi)$ is a finite decreasing function in $[0, \gamma]$, and

$$\inf_{\lambda \in [-\infty, \beta]} \Lambda(\lambda, \Pi) = \Lambda(\beta, \Pi) 1(\beta < \gamma), \quad (3.3.3)$$

where $1(\beta < \gamma)$ is an indicator function that equals 1 when $\beta < \gamma$ and 0 otherwise.

To sum up, the rate function $\Lambda(\beta, \Pi)$ has the following properties:

1. $\Lambda(\beta, \Pi) = \infty$ if $\beta < 0$.
2. $\Lambda(\beta, \Pi) = 0$ only if $\beta = \gamma$.
3. $\Lambda(\beta, \Pi)$ is a finite non-negative decreasing function of $\beta$ in $[0, \gamma]$.

Now we are ready to present our main results. The asymptotic throughput for any values of $K$ as a function of $n$ is characterized by the theorem below:

**Theorem 3.3.1.** Assume that $K$ is a function of $n$ and the value of $\lim_{n \to \infty} \frac{K}{\log n}$ exists (can be infinity), which we denote by $c \triangleq \lim_{n \to \infty} \frac{K}{\log n}$, then we have

$$\lim_{n \to \infty} \eta(n, K) = \sup \left\{ \beta \left| \frac{c}{\Lambda(\beta, \Pi)}, 0 \leq \beta < \gamma \right\} \right.. \quad (3.3.4)$$

**Proof.** see Section B.1. \hfill \Box

From Theorem 3.3.1, we know that, if the coding block size $K$ is set to be a function of the network size $n$, then we can characterize the asymptotic throughput.
when \( n \) approaches infinity in an explicit form. Equation (3.3.4) implies that the asymptotic throughput is a function of \( \gamma, \lim_{n \to \infty} K/\log n \) and the channel correlation structure indicated by \( \Pi \).

By Theorem 3.3.1, the asymptotic throughput in the special cases when \( K \in o(\log n) \) and \( K \in \omega(\log n) \) are given in the following corollary.

**Corollary 3.3.1.1.** Assume that \( K \) is a function of \( n \). We then have

1. if \( K \in o(\log n) \), then\(^2\)

\[
\lim_{n \to \infty} \eta(n, K) = 0.
\]

2. if \( K \in \omega(\log n) \), then

\[
\lim_{n \to \infty} \eta(n, K) = \gamma.
\]

*Proof.* 1) If \( K \in o(\log n) \), then \( c = \lim_{n \to \infty} \frac{K}{\log n} = 0 \) and we have

\[
\{ \beta \mid c \geq \frac{\beta}{\Lambda(\beta, \Pi)}, 0 \leq \beta < \gamma \} = \{0\}.
\]

The above equation is true because \( \Lambda(\beta, \Pi) \) is a finite function on \([0, \gamma)\). Then according to Theorem 3.3.1, we get

\[
\lim_{n \to \infty} \eta(n, K) = \sup\{0\} = 0.
\]

\(^2\)We use standard notations: \( f(n) = o(g(n)) \) if \( \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0 \) and \( f(n) = \omega(g(n)) \) if \( \lim_{n \to \infty} \frac{f(n)}{g(n)} \) diverges.
2) If $K \in \omega(\log n)$, then $c = \lim_{n \to \infty} \frac{K}{\log n} = \infty$ and we have

$$\left\{ \beta \mid c \geq \frac{\beta}{\Lambda(\beta, \Pi)}, 0 \leq \beta < \gamma \right\} = [0, \gamma).$$

The above equation is true because $\Lambda(\beta, \Pi)$ is a non-negative function that attains 0 only when $\beta = \gamma$. Then according to Theorem 3.3.1, we get

$$\lim_{n \to \infty} \eta(n, K) = \sup [0, \gamma) = \gamma.$$

\[\square\]

Corollary 3.3.1.1 says that the throughput will vanish to 0 as $n$ becomes large, when $K$ does not scale as fast as $\log n$. Whereas when $K$ scales faster than $\log n$ (Or more specifically, when $K \in \omega(\log n)$), throughput approaches the capacity $\gamma$ of the system in the limit. It should be noted that Theorem 3.3.1, together with Corollary 3.3.1.1, are a generalized version of Theorem 1 in [25], which only consider the case when $l = 1$ and does not give an explicit expression for the asymptotic throughput.

As a special case when the channels are memoryless ($l = 0$), we can express $\Lambda(\beta, \Pi)$ in a closed form, as shown in the corollary below.

**Corollary 3.3.1.2.** Assume that $K$ is function of $n$ and the channels are memoryless ($l = 0$), we have if $\lim_{n \to \infty} \frac{K}{\log n} = c$, where $c$ is a positive constant, then

$$\lim_{n \to \infty} \eta(n, K) = \sup \left\{ \beta \mid \log \frac{\beta}{\gamma} + \frac{1 - \beta}{\beta} \log \frac{1 - \beta}{1 - \gamma} \geq \frac{1}{c}, 0 \leq \beta < \gamma \right\}.$$

(3.3.5)
Proof. When \( l = 0 \), the channel is memoryless with \( \pi(0,1) = \pi(1,1) = \gamma \) and \( \pi(1,0) = \pi(0,0) = 1 - \gamma \). Then we have

\[
\Pi_{\theta} = \left[ \pi(s, u)e^{\theta f(u)} \right]_{s, u \in \{0, 1\}} = \begin{bmatrix} 1 - \gamma & \gamma e^{\theta} \\ 1 - \gamma & \gamma e^{\theta} \end{bmatrix}.
\]

It is easy to see that \( \rho(\Pi_{\theta}) = 1 - \gamma + \gamma e^{\theta} \). Therefore we have, according to Equation (3.3.1),

\[
\Lambda(\beta, \Pi) = \sup_{\theta \in \mathbb{R}} \left\{ \theta \beta - \log \left( 1 - \gamma + \gamma e^{\theta} \right) \right\}
= \beta \log \frac{\beta}{\gamma} + (1 - \beta) \log \frac{1 - \beta}{1 - \gamma},
\]

where the supremum is attained when \( \theta = \log \frac{\beta(1 - \gamma)}{\gamma(1 - \beta)} \). \( \square \)

### 3.4 Throughput lower bound for finite \( K \) and \( n \)

For rateless coding schemes, the best encoding and decoding complexity increases linearly in \( K \) (e.g., Raptor codes [16]), the size of the coding block. Moreover, the value of \( K \) determines the receiver buffer size. Therefore, in reality, the value of \( K \) is often limited by the decoder buffer size or the computational power of both sender and receiver. We then have to consider the case when \( K \) is finite and need to answer the following questions: For a given number of receivers \( n \), channel statistics, and a maximum available coding block size \( K \), what is maximum packet arrival rate \( \lambda \) that can be supported by this system? For a specific number of receivers and channel statistics, if we are given a target packet arrival rate \( \lambda \), how can we design the value of \( K \) in the system such that the target arrival rate can be supported?

In order to answer these questions, we make a more restrictive channel assumption
that the current channel state is impacted by only the channel state in the previous 1 time slot. In other words, we have the Gilbert-Elliott erasure channel model. Under this model, for any receiver $1 \leq i \leq n$, the channel states $\{X_{ij}\}_{j \in \mathbb{N}}$ evolve with $j$ according to a two state Markov chain as illustrated in Figure 3.2. Here, $p_{01}$ and $p_{10}$ are the transition probabilities between state 0 and state 1, and system capacity $\gamma = p_{01}/(p_{10} + p_{01})$. Based on this model, in the theorem below, we find a lower bound on the throughput for any $n$ and $K$ in terms of the expected transmission time of a system with larger $n$ and $K$.

![Figure 3.2: Gilbert-Elliott Erasure Channel Model](image)

**Theorem 3.4.1.** Under the Gilbert-Elliott erasure channel model ($l = 1$) as illustrated in Figure 3.2, for any $n \in \mathbb{N}$ $K \in \mathbb{N}$ and $\alpha \in \mathbb{N}$, we have $^3$

$$\eta(n, K) > \frac{\alpha K}{\mathbb{E}[T(n^\alpha, \alpha(K + K_0))|\mathcal{E} = 1_n^\alpha]},$$

where

$$K_0 = \min \left\{ m \geq 0 \left| \sum_{d=0}^{m} (1 - p_{10})^d p_{10} + p_{01} \geq 1 \right. \right\}.$$

$^3$We denote an all-one vector with dimension $m$ as $1_m$
Remark 3.4.1.1. Observe that $K_0$ is independent of the choice of $n$ and $K$ and is only a function of channel dynamics. $K_0 = 0$ if and only if $p_{01} + p_{10} \geq 1$.

Proof. see Section B.5. □

When the channels are memoryless, the above theorem reduces to a simpler form, as shown in the following corollary.

Corollary 3.4.1.1. When the channels are memoryless ($l = 0$), for any $n \in \mathbb{N}, K \in \mathbb{N}$ and $\alpha \in \mathbb{N}$, we have

$$\eta(n, K) > \eta(n^\alpha, \alpha K).$$

Remark 3.4.1.2. While Theorem 3.3.1 tells us that in order to achieve nonzero throughput, we can double the coding block size $K$ for every quadratic increase of $n$, which is to make $K/\log n$ a fixed value, it does not tell us anything about how the throughput will converge as $n$ approaches infinity. This corollary indicates that under the memoryless channel assumption, if we adapt the coding block size $K$ with the increase of network size $n$ in a way that $K/\log n$ is kept as a fixed value, then the throughput will follow a decreasing pattern before it reaches the asymptotic throughput.

Proof. The memoryless channels can be considered as a special case of the Gilbert-Elliott erasure channels when $p_{10} = 1-\gamma, p_{01} = \gamma$ and $K_0 = 0$, then by Theorem 3.4.1,

$$\eta(n, K) > \frac{\alpha K}{\mathbb{E}[T(n^\alpha, \alpha K)|\mathcal{E} = 1_{n^\alpha}]} = \frac{\alpha K}{\mathbb{E}[T(n^\alpha, \alpha K)]} = \eta(n^\alpha, \alpha K).$$

The last equation follows by noting that $T(n^\alpha, \alpha K)$ is independent of initial state $\mathcal{E}$ when the channels are memoryless. □
By the help of the Theorem 3.3.1 and Theorem 3.4.1, we can get a lower bound on
the maximum stable throughput that can be achieved for any finite values of coding
block size $K$ and network size $n$, as shown in the theorem below.

**Theorem 3.4.2.** For a broadcast network with $n$ receivers, and coding block size
$K$, under the Gilbert-Elliott erasure channel model ($l = 1$), the throughput is lower
bounded by

$$\eta(n, K) > \frac{K}{K + K_0} \mathcal{R} \left( \frac{K + K_0}{\log n} \right),$$

and thus the system with packet arrival rate $\lambda$ is stable if

$$\lambda \leq \frac{K}{K + K_0} \mathcal{R} \left( \frac{K + K_0}{\log n} \right),$$

where

$$\mathcal{R} (r) = \sup \left\{ \beta \bigg| r \geq \frac{\beta}{\Lambda(\beta, \Pi)}, 0 \leq \beta < \gamma \right\},$$

and

$$K_0 = \min \left\{ m \geq 0 \bigg| \sum_{d=0}^{m} (1 - p_{10})^d p_{10} + p_{01} \geq 1 \right\}.$$

**Remark 3.4.2.1.** As a special case, when the channels are memoryless, we have
$p_{10} = 1 - \gamma$ and $p_{01} = \gamma$. It is easy to obtain that $K_0 = 0$. Therefore,

$$\eta(n, K) > \sup \left\{ \beta \bigg| \log \frac{\beta}{\gamma} + \frac{1 - \beta}{\beta} \log \frac{1 - \beta}{1 - \gamma} \geq \frac{\log n}{K}, 0 \leq \beta < \gamma \right\},$$
and the system with packet arrival rate $\lambda$ is stable if

$$\lambda \leq \sup \left\{ \beta \left| \log \frac{\beta}{\gamma} + \frac{1 - \beta}{\beta} \log \frac{1 - \beta}{1 - \gamma} \geq \frac{\log n}{K}, 0 \leq \beta < \gamma \right. \right\}.$$ 

**Proof.** From Lemma B.1.1 in Section B.1, we can see that when $K$ and $n$ are finite, the transmission time of a coding block $T(n, K)$ is light-tail distributed, meaning that it has finite variance. Then according to [29], using Lyapunov method we know that the queue will be stable if the traffic intensity of this queue, which is defined as the packet arrival rate $\lambda$ over the service rate, is less than 1. Therefore, the queue will be stable if the arrival rate $\lambda$ satisfies

$$\lambda < \sup \left\{ \mu \left| \frac{\mu}{K/E[T(n, K)]} < 1 \right. \right\}$$

$$= \eta(n, K).$$

(3.4.1)

By Theorem 3.4.1 we know that, for any integer values of $\alpha$

$$\eta(n, K) > \frac{\alpha K}{E[T(n^\alpha, \alpha(K + K_0))))|E = 1_{n^\alpha}],}$$

implying that

$$\eta(n, K)$$

$$> \lim_{\alpha \to \infty} \frac{K}{K + K_0} \frac{\alpha(K + K_0)}{E[T(n^\alpha, \alpha(K + K_0))]|E = 1_{n^\alpha}].}$$

(3.4.2)

Since $\alpha(K + K_0)/ \log n^\alpha = (K + K_0)/ \log n$ for any value of $\alpha$, then by Equation (B.1.6)
in the proof of Theorem 3.3.1, we get

\[
\lim_{\alpha \to \infty} \frac{\alpha(K + K_0)}{E[T(n^\alpha, \alpha(K + K_0)) | E = 1_n^\alpha]}
= \sup \left\{ \frac{K + K_0}{\log n} \geq \frac{\beta}{\Lambda(\beta, \Pi)}, 0 \leq \beta < \gamma \right\}
= R\left(\frac{K + K_0}{\log n}\right),
\]

which, by combining Equation (3.4.1) and Equation (3.4.2), completes the proof. □

In order to compare this lower bound on the maximum achievable rate with the existing bounds given in [23] and [24], we restate Theorem 2 in [23] and Theorem 7 in [24] as the following.

**Theorem 3.4.3** (Theorem 2 in [23] and Theorem 7 in [24]). In a broadcast network with \( n \) receivers, coding block length \( K \) and packet arrive rate \( \lambda \),

1) when the channels are memoryless \((l = 0)\) with erasure probability \(1 - \gamma\) and \( K > 16 \), the system is stable if

\[
\lambda < \frac{(1 - \gamma)K}{K + (\log n + 0.78)\sqrt{K} + 2.61}.
\]

2) For Gilbert-Elliott erasure channels \((l = 1)\) with state transition probability \( p_{10} \) and \( p_{01} \), when \(1 - p_{10} - p_{01} \geq 0\) and \( K \geq 21\log n - 4 \), the system is stable if

\[
\lambda < \frac{p_{01}K}{K + 2\sqrt{(0.78K + 3.37)\log n} + 2.61}.
\]

For ease of notation let us denote the bounds given in Theorem 3.4.3 as the CSE bound 1 and CSE bound 2 respectively using the initials of the authors’ last name.

Note that both CSE bound 1 [23] and CSE bound 2 [24] have simply closed-form expressions which give insight into how the change of system parameters would affect
the achievable throughput, while our bound in Theorem 3.4.2 cannot be reduced to a closed-form expression, and is expressed in terms of a supremum involving a large deviation rate function. However, our bound has several advantages over the CSE bounds. Firstly we should note that the CSE bounds are only valid when $K$ and $n$ satisfy certain conditions, while our bound is valid for any finite values of $K$ and $n$. Secondly, our bound converges to the asymptotic throughput in the sense that as $n$ approaches infinity while keeping $K/\log n$ as a constant $c$, our bound on the maximum achievable rate will converge to the asymptotic throughput with parameter $c$. Or more specifically,

$$\lim_{n \to \infty} \frac{K}{K + K_0} R \left( \frac{K + K_0}{\log n} \right) = R \left( \lim_{n \to \infty} \frac{K + K_0}{\log n} \right) = R(c) = \lim_{n \to \infty} \eta(n, K),$$

(3.4.3)

which can be seen from Theorem 3.3.1 and Theorem 3.4.2. However, the CSE bounds are not asymptotically tight. When we keep the ratio $K/\log n$ to be a constant $c$, as $n$ or $K$ approaches infinity, CSE bound 1 even becomes trivial (approaches 0), which can be seen from the equation below.

$$\lim_{K \to \infty} \frac{(1 - \gamma)K}{K + (\log n + 0.78)\sqrt{K} + 2.61}$$

$$= \lim_{K \to \infty} \frac{(1 - \gamma)}{1 + (1/c + 0.78/K)\sqrt{K} + 2.61/K}$$

$$= 0.$$ 

(3.4.4)

Next, in Section 3.5, we show that our bound outperforms the CSE bounds under various simulation settings.
3.5 Simulation

In this Section, we conduct simulation experiments to verify our main results.

Example 3.5.1. This example verifies Theorem 3.3.1, Corollary 3.4.1.1, and Theorem 3.4.2. We choose a memoryless channel with $\gamma = 0.5$. By keeping $K/\log n$ as a constant $15/\log 2$, we change $K$ from 5 to 300 and calculate the maximum achievable rate, which is $\eta(n, K)$, through simulations for each pair of $(K, n)$. Since the value of our bound is a function of the ratio $K/\log n$, in this case, it is a constant for all $K$ and is equal to the asymptotic throughput with parameter $15/\log 2$. From Figure 3.3 we can see that as $K$ approaches infinity, the maximum achievable rate converges to our lower bound (which is also the asymptotic throughput in this case) in a decreasing manner, which validates Theorem 3.3.1, Corollary 3.4.1.1 and Theorem 3.4.2.
Example 3.5.2. In order to verify Theorem 3.4.2 under the Gilbert-Elliott erasure channel model, we choose the state transition probability $p_{10} = p_{01} = 0.4$. It is easy to obtain that $\gamma = 0.5$ and $K_0 = 1$. By keeping $K/\log n$ as a constant $5/\log 2$ and changing $K$ from 5 to 100, we plot in Figure 3.4 both the simulation result of the maximum achievable rate and our lower bound shown in Theorem 3.4.1. Again, as we can see from the figure, the lower bound becomes tighter as $K$ increases. Eventually the maximum achievable rate will converge to the lower bound as shown in Equation (3.4.3). However, neither of the CSE lower bounds is valid under this system setting.

Example 3.5.3. In this example, we conduct three set of experiments under the
memoryless channel assumption with different values of $K$ as a function of $n$, and show that our bound outperforms the CSE bound in all these simulation settings.

In the first case, we set the coding block size $K$ to be the same as the network size $n$ and change $n$ from 5 to 300. We plot the simulation result of the maximum achievable rate as well as our bound and the CSE bound in Figure 3.5a, since in this case $K$ scales faster than $\log n$, the achievable rate will approach system capacity $\gamma$ as the network size $n$ grows.
In the second case, we assume that the number of receivers is fixed to be 10 and we increase coding block size $K$ from 5 to 300. The simulations result, together with the two bounds, are plotted in Figure 3.5b. In this case, the achievable rate will also approach system capacity $\gamma$ as $n$ increases.

In the final case, as shown in Figure 3.5c, we keep the coding block size to be a constant 80 and increase the number of receivers from 5 to 100. Since $K$ does not increase with $\log n$ at all, the achievable rate will vanish to 0 as $n$ grows.

From Figures 3.3, 3.5a, 3.5b and 3.5c, we can see that our bound obtained by Theorem 3.4.2 is significantly better than the lower bounds achieved in [23] and [24] in all these four different cases.

3.6 Conclusion

In this chapter, we characterize the throughput of a broadcast network using rateless codes. The broadcast channels are modeled by Markov modulated packet erasure channels, where the packet can either be erased or successfully received and for each receiver the current channel state distribution depends on the channel states in previous $l$ packet transmissions.

We first characterize the asymptotic throughput of the system when $n$ approaches infinity for any values of the coding block size $K$ as a function of number of receivers $n$ in an explicit form. We show that as long as $K$ scales at least as fast as $\log n$, we can achieve a non-zero asymptotic throughput. Under the more restrictive Gilbert-Elliott erasure channel model ($l = 1$), we study the case when $K$ and $n$ are finite. For any $K$ and $n$, we find a lower bound on the throughput in terms of the transmission time of a system with larger $K$ and $n$. As a special case when channels are memoryless, this result shows that, by keeping the ratio $K/\log n$ to be a constant, the system throughput will converge to the asymptotic throughput in a decreasing manner as $n$
grows. By the help of these results, under the Gilbert-Elliott erasure channel model, we are able to give a lower bound on the maximum achievable throughput (maximum achievable rate), which is a function of $K$, $n$ and state transition probabilities $p_{01}$ and $p_{10}$. In contrast to the state-of-the-art, we analytically show that our bound is asymptotically tight when $K/\log n$ is fixed as $n$ approaches infinity. Further, through numerical evaluations, we show that our bound is significantly better than existing results.
CHAPTER 4
CHARACTERIZING THE ACHIEVABLE THROUGHPUT
IN WIRELESS NETWORKS WITH TWO ACTIVE RF
CHAINS

4.1 Introduction

Characterizing the throughput region of a general wireless network has been an open problem for a long time, partially because the intrinsic broadcast nature of wireless communication leads to a very complex interference coupling among simultaneous transmissions. The de facto method to manage interference coupling and coordinate wireless transmissions is to treat interference simply as noise and to avoid it by scheduling links that do not cause serious interference with each other. Generally speaking, under scheduling, a group of wireless links can be activated at the same time only when they all satisfy their SINR constraints. Recent developments in physical layer wireless technologies show that by dealing with interference in a smart way rather than simply treating it as noise and avoiding it, e.g., by using the interference alignment technique [30], or noisy network coding [31], a higher network throughput can be obtained versus traditional interference avoidance techniques. However, almost all of these techniques require extensive knowledge of the entire network state information, and it is still an open problem regarding how to incorporate them into the scheduling problem under a general wireless network.
Therefore, in this chapter, we restrict our focus to the case when interference in the wireless network is treated as noise and the channel state information is known only at the receiver, and leave the accommodation of more advanced physical layer techniques such as interference alignment [30] as future work. To characterize the interference relationship between the nodes, we assume a binary interference channel, where any two nodes in a wireless network either interfere with each other, or do not interfere at all. At any scheduled time-slot, the rate of a link is simply represented by the number of data streams that are transmitted on that link, which means that the rate of a link is either 0, 1, or 2. Henceforth, whenever we refer to the optimal throughput region or the achievable throughput it is with respect to this interference model.

It should be noted that even with this simplified 0-1 interference model and simplified throughput definition, the comparison of the throughput regions achieved by different multi-RF chain techniques is a challenging task. The fundamental reason is that different transmission modes leave different interference footprints on the rest of the network. Consider a simple scenario shown in Fig. 4.1. Assume that node A and node B lie in a wireless network where every node has two set of RF chains and can operate in either MIMO or full-duplex mode, and the circle around these two nodes indicates their interference regions. If there is a bidirectional transmission between node A and node B, as shown in Fig. 4.1(a), then any node within their interference region cannot be transmitting, since both nodes A and B are utilizing both of their two RF chains and no extra interference can be decoupled using multiple receive RF chains. At the same time, if a node lies in and only in one of node A and node B’s interference region, then it cannot be receiving more than one data stream, because its two RF chains can at most resolve two spatially multiplexed streams, one of which is the interference from either node A or node B. On the other hand, when
Nodes in this region can neither transmit any data flows nor receive more than 1 flow.
Nodes in this region cannot transmit any data flows

(a) Bidirectional transmission
(b) Multiplexing transmission

Figure 4.1: The interference footprint of the transmission between node A and node B in a wireless network where each node has two active RF chains and can enable either full-duplex or MIMO transmission. The circle around node A/B indicates its interference range. In (a), both node A and node B are operating in full-duplex, and a bidirectional transmission is established between the two nodes; in (b), two spatially multiplexed streams are sent from node A to node B.

If node A is transmitting two spatially multiplexed data streams to node B, as shown in Fig. 4.1(b), then it is easy to see that any node within A’s interference region cannot be receiving any number of data streams, because every node has only two RF chains and, therefore, cannot decode more than two spatially multiplexed streams. For the same reason, any node within B’s interference region cannot be transmitting any number of data streams.

The intellectual merit of this chapter lies in answering the following two key questions:

1. In a general wireless network with at least two antennas and at most two active
RF chains at every node, for any two different RF hardware structures, what is the relationship between their achievable throughput regions?

2. For any two different RF hardware configurations, under what exact network scenario can one achieve a strictly larger throughput region than the other?

Given that there are four different RF hardware configurations as shown in Fig. 1.3, the above two questions can be broken down into several smaller questions, such as: Is there a network scenario under which full-duplex can achieve a better throughput region than MIMO? Does MIMO alone achieve all the throughput gains brought by full-duplex in any scenario? When MIMO and full-duplex coexist in a network, can we achieve a better throughput region than MIMO or full-duplex alone? If so, under what network scenario? The rest of this chapter answers these questions and provides a very clear guideline on which network topologies and traffic patterns could result in improvement for one technology over another.

4.2 System Model

We consider a wireless network where each wireless node has two sets of RF chains and there is no channel state information at the transmitter. We assume a binary symmetric interference structure between wireless nodes, where the transmission of a node either interferes with the reception of another node, or does not interfere at all, and when node \( A \) interferes with node \( B \), node \( B \) also interferes with node \( A \). Note that this binary interference model is widely adopted in the scheduling literature \([32-35]\), and serves as an important first step towards scheduling under the more complex, SINR interference model \([36,37]\).

Let \( \mathcal{G} = \{\mathcal{N}, \mathcal{E}\} \) denote the undirected network topology graph, which captures the interference relationship between the wireless nodes. Here \( \mathcal{N} \) is the set of wireless
nodes, and $\mathcal{E}$ is the set of edges between the wireless nodes. For $A, B \in \mathcal{N}$, $\{A, B\} \in \mathcal{E}$ if and only if $A$ and $B$ interfere with each other. Among these edges, we further denote $\mathcal{E}_L$ as the set of edges where a direct wireless data-link can be established between the two end nodes. Then the set of data links can be denoted as

$$\mathcal{L} = \{(A, B), (B, A)\mid \{A, B\} \in \mathcal{E}_L\}.$$ 

Since every node has two antennas, for each data link, there can either be one data stream, or a pair of multiplexed data streams, enabled by MIMO multiplexing. Therefore, we further denote the set of all possible data streams as

$$\mathcal{L}_{12} = \{(A, B)^1, (A, B)^2\mid (A, B) \in \mathcal{L}\},$$

where $(A, B)^1$ stands for a single data stream from node $A$ to node $B$, and $(A, B)^2$ stands for two multiplexed data streams from node $A$ to node $B$. In the rest of the chapter we make a simplified assumption that $\mathcal{E}_L = \mathcal{E}$. However, our analysis can be readily extended to the case when $\mathcal{E}_L \subset \mathcal{E}$.

### 4.2.1 Constraint set

As described in the Introduction, when a node has at least two antennas and can simultaneously activate two RF chains, it can have at least three different transceiver structures as shown in Fig. 1.3. Given our assumption that the channel state information is known only at the receiver, it is easy to see that a wireless node has the potential to operate in either one of the ten modes shown in Fig. 4.2. Among these ten modes, mode 1 and 2 are SISO modes, mode 3 and 4 are full-duplex modes, mode 5 and 6 are MIMO multiplexing modes, and mode 7 through 10 are multi-user MIMO modes.
As we have discussed in the introduction, which modes can be activated (or not) depends on the node’s hardware configuration. Given that there are four RF architectures, as shown in Fig. 1.3, we can set up the following four constraint sets which restrict the modes a node can operate in, based on its RF configuration.

1) Single-input Single-output (SS) constraint set: In this constraint set, each node has only one set of RF chains with no cancellation module, as shown
in Fig. 1.3(a). As a result, each node can only operate in SISO modes, which are mode 1 and mode 2 in Fig. 4.2. Note that this constraint set does not allow the simultaneous activation of two RF chains, and it serves only as a baseline for showing the throughput gains of two active RF chains.

2) Full-duplex (FD) constraint set: Nodes in this constraint set have one complete set of RF chains plus a cancellation module, as shown in Fig. 1.3(b), where one antenna is designated as a transmit antenna and the other as a receive antenna. The cancellation module cancels the self-interference from the transmit antenna to the receive antenna [7–9]. Therefore, the node can operate in both SISO and full-duplex modes, which are modes 1-4 in Fig. 4.2.

3) MIMO (MM) constraint set: In this set, each node has two complete sets of RF chains, as shown in Fig. 1.3. Hence, it can operate in MIMO communication modes, meaning that a node can either transmit or receive two data streams simultaneously. In Fig. 4.2, modes 5-10 form MIMO modes, where modes 5 and 6 correspond to MIMO multiplexing, and modes 7-10 represent multi-user MIMO. In order to differentiate between MIMO multiplexing and multi-user MIMO, we define two sub-constraint-sets, namely MMa and MMb constraint sets. In the MMa set, multi-user MIMO is not allowed and all the nodes can operate in one of modes 1, 2, 5, and 6. In MMb set, both MIMO multiplexing and multi-user MIMO are allowed, meaning that all the nodes can operate in one of modes 1, 2, and 5-10.

4) Full-duplex MIMO (FM) constraint set: In this set, each node has two complete sets of RF chains and a cancellation module, which serves as the self-interference cancellation circuit. Nodes in this mode can operate in either full-duplex modes or MIMO modes. Similarly, as in the previous constraint set, to differentiate between MIMO multiplexing and multi-user MIMO, we define two sub-constraint-sets, namely FMa and FMb set. In the FMa set, nodes can operate in either one
of mode 1 to mode 6, while in FMb set, nodes can operate in any of the ten modes shown in Fig. 4.2.

A subset of data links $S \subseteq L_{12}$ is called a \textit{feasible activation set} under a constraint set if the links within that subset can be activated at the same time without violating the modes of transmission defined by that constraint set. We denote the set of all feasible activation sets under a certain constraint set as $S_{\text{Constr}}(G)$. For example,

$$S_{\text{SS}}(G) = \{ S \subseteq L_{12} | S \text{ is a feasible activation set in topology } G \text{ under SS constraint set} \}.$$

We also call $S \in S_{\text{SS}}(G)$ an SS-feasible schedule. We use a vector in $\{0, 1, 2\}^{L}$ to denote a schedule $S$ such that the $k^{th}$ element is set to 1 if a single data stream is scheduled on link $k \in L$, set to 2 if two multiplexed streams are scheduled on link $k \in L$, and set to 0 otherwise. We denote the vector representing schedule $S$ as $\vec{R}(S)$, then it is formally defined as follows:

$$\vec{R}(S) = \left[ R_{(A,B)}(S), (A,B) \in L \right], \text{ and}$$

$$R_{(A,B)}(S) = \begin{cases} 
1 & \text{if } (A,B)^1 \in S \\
2 & \text{if } (A,B)^2 \in S \\
0 & \text{otherwise}
\end{cases} \quad (4.2.1)$$

$\vec{R}(S)$ counts the number of independent data streams that are packed into schedule $S$, hence we also call it the rate vector of schedule $S$.

Given the set of all activation sets $S_{\text{Constr}}(G)$, we denote the set of all corresponding rate vectors as $R_{\text{Constr}}(G) = \{ \vec{R}(S) | S \in S_{\text{Constr}}(G) \}$, and let $\text{Co}(R_{\text{Constr}}(G))$ denote its
convex hull, where the convex hull $\text{Co}(A)$ of a set $A$ is defined as

$$\text{Co}(A) \triangleq \left\{ \sum_i w_i \alpha_i \mid w_i \geq 0, \sum w_i = 1, \alpha \in A \right\}.$$  

The *throughput region* under a given scheduling policy is defined as the set of vectors $\bar{\lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_{|L|}\}$ for which the long term average throughput achieved over link $i$ can be greater than or equal to $\lambda_i$ for every $i$. In this sense, the *optimal throughput region* that can be achieved by scheduling is known to be [38]

$$\mathbb{D}_{\text{Constr}}(\mathcal{G}) = \left\{ \bar{\lambda} \mid \bar{\lambda} \preceq \bar{\theta}, \text{ for some } \bar{\theta} \in \text{Co}(\mathbb{R}_{\text{Constr}}(\mathcal{G})) \right\},$$

which also equals to $\text{Co}\{\mathbb{R}_{\text{Constr}}(\mathcal{G}), 0\}$. A summary of all the notations can be found in Table 4.1. *Without causing confusion, in the rest of the chapter, we will abuse notation slightly by writing $S_{\text{Constr}}(\mathcal{G})$, $\mathbb{R}_{\text{Constr}}(\mathcal{G})$ and $\mathbb{D}_{\text{Constr}}(\mathcal{G})$ as $S_{\text{Constr}}$, $\mathbb{R}_{\text{Constr}}$, and $\mathbb{D}_{\text{Constr}}$, respectively.*

Table 4.1: Notations (network graph $\mathcal{G}$ with SS constraint set)

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{L}_{12}$</td>
<td>The set of all data streams (include single stream and multiplexed stream) in $\mathcal{G}$.</td>
</tr>
<tr>
<td>$S \subset \mathcal{L}_{12}$</td>
<td>A subset of data streams, also called an activation set, or a schedule.</td>
</tr>
<tr>
<td>$\bar{R}(S)$</td>
<td>The rate vector of schedule $S$ defined in Equation (4.2.1).</td>
</tr>
<tr>
<td>$S_{\text{SS}}(\mathcal{G})$</td>
<td>The set of all feasible schedules under SS constraint set.</td>
</tr>
<tr>
<td>$\mathbb{R}_{\text{SS}}(\mathcal{G})$</td>
<td>The set of rate vectors of all the schedules in $S_{\text{SS}}(\mathcal{G})$.</td>
</tr>
<tr>
<td>$\mathbb{D}_{\text{SS}}(\mathcal{G})$</td>
<td>The optimal throughput region of $\mathcal{G}$ under SS constraint set.</td>
</tr>
</tbody>
</table>
Figure 4.3: Throughput regions of a five-node cyclic network under SS, FD, MMa/MMb, and FMa/FMb constraint sets. Here we only focus on the data links from A to E, from D to C and from B to A. Each dot in the throughput region plot represents a feasible schedule under the corresponding constraint set. The red arrows on each network graph illustrate one such feasible schedule, while the parentheses after each node represents its communication mode, and these feasible schedules correspond to the big red dots on the throughput region plot. (a) When every node has only one active RF chain at any time, no more than one data stream in these three links can be simultaneously activated. (b) When every node has full-duplex capability, then a data stream from B to A and a data stream from A to E can be activated together. (c,d,e) Under MMa/MMb/FMa constraint set, at any time we cannot support more than 2 simultaneous data streams in the three links. (f) When every node have both multi-user MIMO and full-duplex capabilities, three data streams, one for each data link, can be activated together. The comparison between (b,d) and (f) shows that a strictly larger throughput region can be achieved when each node can choose to operate in either MIMO or full-duplex mode, than the case when only MIMO, or only full-duplex is supported.
4.3 Optimal throughput region

Given the six constraint sets and the definition of the optimal throughput region that can be achieved by scheduling, a natural question to ask is what relationship exists between the achievable throughput regions under different constraint sets. In other words, does an upgrade in the RF hardware bring additional throughput gain in the wireless network when scheduling is used? If the throughput region achieved by constraint set A is not always the same as that achieved by constraint set B for any wireless network, under what network condition can one of constraint sets A and B provide a strictly better throughput region than the other? This section answers these questions.

While it is easy to compare the throughput achieved by different constraint sets under an elementary wireless network, when we consider a more general wireless network, it is not clear if some modes in Fig. 4.2 are always better than some others, because different modes of transmission of a node leave different interference footprints on the rest of the network, as discussed in the introduction.

The theorem below characterizes the relationship between the throughput regions that are achieved by the six constraint sets for any network graph.

**Theorem 4.3.1.** For any network topology graph $\mathcal{G}$, the throughput regions under $SS$, $FD$, $MMa/MMb$, and $FMa/FMb$ constraint set have the following relationship:

\[
D_{MMa} = 2D_{SS} \tag{4.3.1}
\]

\[
D_{SS} \subseteq D_{FD} \subseteq D_{MMa} \subseteq D_{FMa} \subseteq D_{MMb} \subseteq D_{FMb} \tag{4.3.2}
\]

**Remark 4.3.1.1.** Equation (4.3.1) is obvious because by using 2 by 2 MIMO multiplexing allowed in $MMa$ constraint set, each link in a $SS$-feasible schedule can transmit two independent data streams simultaneously instead of one, and therefore achieving
twice the throughput. Part (a), (d), and (e) in Equation (4.3.2) are also straightforward, since the available modes under SS, MMa, and MMb constraint set are included in FD, MMb, and FMb constraint set, respectively.

**Remark 4.3.1.2.** Part (b) in Equation (4.3.2) says that if the two antennas on each node can only be used to enable full-duplex transmission, i.e., each node has the RF structure of Fig. 1.3(b), then the achievable throughput region, for any network graph, is always a subset of that achieved in the case when each node can operate in MIMO multiplexing modes, enabled by two complete sets of RF chains. Part (c) implies that, when multi-user MIMO is not allowed, then upgrading the RF hardware from that in Fig. 1.3(c) to that in Fig. 1.3(d) does not increase the achievable throughput region, for any network graph. Fig. 4.3 gives an illustration of the relationship between the different throughput regions in a five-node cyclic network graph.

**Remark 4.3.1.3.** In general, establishing the relationship of these throughput regions for a network on case by case basis seems very hard because it depends on network topologies and traffic patterns. However, by creating the notion of the 10 modes, in the proof, we are able to circumvent this inherent difficulty and decouple the problem from a network back to what happens at the end nodes of a link belonging to each activation set.

**Proof.** We focus on the proof of parts (b) and (c) in Equation (4.3.2). Since the available modes in FD constraint set is a subset of that in FMa constraint set, for every $S_1 \in S_{FD}$, $S_1 \in S_{FMa}$. Therefore, to prove $D_{FD} \subseteq D_{MMa} = D_{FMa}$, it suffices to show that for every $S_1 \in S_{FMa}$, we can construct $S_2, S_3 \in S_{MMa}$ such that $2\tilde{R}(S_1) = \tilde{R}(S_2) + \tilde{R}(S_3)$.

Let us start with an activation set $S_1 \in S_{FMa}$. From the definition of the FMa constraint set, we know that each link in $S_1$ must fall into one of the following four categories based on the operation modes of its two end nodes. (i) The transmitter is
in mode 1 and the receiver is in mode 2. (ii) Both the transmitter and the receiver are in mode 4. (iii) Either the transmitter or the receiver is in mode 3. (iv) The transmitter is in mode 5 and the receiver is in mode 6.

Then, let us construct two activation sets $S_2$ and $S_3$ based on $S_1$ by going through links in all these four different categories. Initially set both $S_2$ and $S_3$ as the empty set.

(i). If link $(A, B)^1 \in S_1$ falls into the first category, then add $(A, B)^2$ to $S_2$.

(ii). If link $(A, B)^1$ and $(B, A)^1$ are both in $S_1$, then we know that they both fall into the second category. Add either $(A, B)^2$ or $(B, A)^2$ to $S_2$ and the remaining one to $S_3$.

(iii). If link $(A, B)^1$ falls into the third category, and $B$ is working in mode 3, then add $(A, B)^2$ to $S_2$. Otherwise if $A$ is working in mode 3, add $(A, B)^2$ to $S_3$.

(iv). If link $(A, B)^2$ is in $S_1$, we know that it falls into the fourth category. Add $(A, B)^2$ to both $S_2$ and $S_3$.

It is clear, then, that $2\bar{R}(S_1) = \bar{R}(S_2) + \bar{R}(S_3)$, since for every $(A, B)^1 \in S_1$, $(A, B)^2$ is in either $S_2$ or $S_3$, and for every $(A, B)^2 \in S_1$, $(A, B)^2$ is in both $S_2$ and $S_3$.

Next we need to show that $S_2, S_3 \in S_{MMa}$. Assume to the contrary that $S_2 \notin S_{MMa}$. This means that there exist two links $(A, B)^2$ and $(C, D)^2$ in $S_2$ such that $C$ interferes with $B$ (more accurately, $\{C, B\} \in \mathcal{E}$), or $B = C$. However, if $B = C$, meaning that $B$ is operating in full-duplex modes (mode 3 and 4) under schedule $S_1$, $(A, B)^2$ and $(C, D)^2$ cannot be both in $S_2$, according to the way $S_2$ and $S_3$ are constructed in the second and third category; if $\{C, B\} \in \mathcal{E}$, then we know that $(A, B)^x$ and $(C, D)^y$ cannot be scheduled simultaneously under FMa constraint set for any $x, y \in \{1, 2\}$. Thus, we have a contradiction, so $S_2 \in S_{MMa}$. Similarly we can show that this is also true for $S_3$. $\square$
Note that in both the MMa/b and FMa/b constraint set, each node has two complete sets of RF chains, and both MIMO multiplexing modes and multi-user MIMO modes can fully exploit the potential of the two complete sets of RF chains. Then, one may reach the conjecture that part (d) and (e) in Equation (4.3.2) are actually equality signs, meaning that $D_{FMa} = D_{MMb} = D_{FMb}$.

However, quite the contrary, Fig. 4.3 provides an interesting counterexample in which the throughput region achieved by the FMb constraint set is strictly larger than that achieved by MMb constraint set. This observation is somewhat surprising in that although Theorem 4.3.1 shows that the throughput region achieved by full-duplex communication is only a subset of that achieved by MIMO communication, when every node can choose to activate either MIMO communication or full-duplex communication, a larger throughput region may be achieved.

Also, Fig. 4.4 provides another simple counterexample in which the throughput region achieved by MIMO multiplexing is strictly a subset of that achieved by MMb constraint set, where every node can operate in either MIMO multiplexing, or multi-user MIMO modes.

Therefore, the next important question is, can we characterize all the network conditions under which we have $D_{FMa} \subset D_{MMb}$ and $D_{MMb} \subset D_{FMb}$. Theorem 4.3.2 and Theorem 4.3.3 below answer this question, where both necessary and sufficient conditions are given for the two strict inclusion relations to hold.

**Theorem 4.3.2.** $D_{MMa} \subset D_{MMb}$ if and only if the network topology $G$ contains a cycle with $2N_l$ nodes, where $N_l$ is an odd number, such that if we label the nodes in that cycle sequentially from 0 to $2N_l - 1$, then there is no chord between a node with even label and a node with odd label.

**Proof.** Before presenting the proof, let us first introduce some necessary notations.
Figure 4.4: Throughput region of a six-node cyclic network under MMa/MMb constraint set. Here we only focus on three links: B to A, F to E, and D to C. Each dot in the throughput region plot represents a feasible schedule under the corresponding constraint set. The red arrows on each network graph illustrate one such feasible schedule, with the parentheses after each node represents its communication mode, and these feasible schedules correspond to the big red dots on the throughput region plot. (a) With only MIMO multiplexing enabled, no more than two data streams can be scheduled, but with multiuser MIMO capability at each node, three data streams, one for each link, can be activated simultaneously, as can be seen in (b).

For any $S_1 \in \mathcal{S}_{\text{FMb}}$, let $N_R(S_1) \triangleq \{ A \in \mathcal{N} | A \text{ is receiving under schedule } S_1 \}$. Then $N_R(S_1)$ is the set of nodes that operate in either mode 2, mode 3, mode 4, mode 6, mode 8, mode 9, or mode 10 under schedule $S_1$. Since FMb constraint set contains every possible modes a node can operate in, $N_R(S_1)$ is well-defined for $S_1$ being a feasible schedule under any other constraint sets.

For any $A, B \in N_R(S_1)$, we say $A$ affects $B$ under schedule $S_1$ when a flow received
by node $A$ is either transmitted by $B$ or seen by node $B$ as interference. Define

$$E_R(S_1) \triangleq \{(A, B) | A, B \in N_R(S_1), \quad A \text{ affects } B \text{ under schedule } S_1\}.$$ 

Further denote the directed graph formed by $N_R(S_1)$ and $E_R(S_1)$ as $G_R(S_1) \triangleq \{N_R(S_1), E_R(S_1)\}$. Assume, without loss of generality, that there are $M$ maximal weakly connected\(^1\) subgraphs in $G_R(S_1)$. Denote the $m^{th}$ such subgraph as $G_R(S_1^{(m)})$, where $S_1^{(m)}$ is the set of data links associated with the $m^{th}$ subgraph, with $\cup_{m=1}^{M} S_1^{(m)} = S_1$. For any directed graph $G$, let $\tilde{G}$ denote the corresponding undirected underlying graph.

Since each node in $N_R(S_1)$ is a receiver node, for any $A \in N_R(S_1)$, there is at most one node in $N_R(S_1)$ that affects $A$. Otherwise, if there are two different nodes that affect node $A$, then node $A$ will see at least two interference flows and cannot be a receiver. This means, in the directed graph $G_R(S_1)$, the indegree of each node cannot be more than one.

The rest of the proof is done in the following two steps, where the details can be found in Section C.6.

**Step 1:** we show that for any network topology $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, $\mathcal{D}_{\text{MMA}}(\mathcal{G}) \subset \mathcal{D}_{\text{MMM}}(\mathcal{G})$ if and only if there exists $S_1 \in \mathcal{S}_{\text{MMM}}(\mathcal{G})$ such that $G_R(S_1)$ contains an odd length cycle.

**Step 2:** we show that for any network topology $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, there exists $S_1 \in \mathcal{S}_{\text{MMM}}(\mathcal{G})$, such that $G_R(S_1)$ contains a cycle with odd length $N_i$ if and only if $\mathcal{G}$ contains a cycle with $2N_i$ nodes such that if we label the nodes on that cycle

---

\(^1\)A directed graph is called weakly connected if replacing all of its directed edges with undirected edges produces a connected (undirected) graph.
sequentially from 0 to \(2N_l - 1\), then there is no chord between a node with even label and a node with odd label.

\[\square\]

**Theorem 4.3.3.** \(\mathcal{D}_{MMb} \subset \mathcal{D}_{FMb}\) if and only if the network topology contains a cycle with \(2N_l + N_f\) nodes, where \(N_l + N_f\) is an odd number, and \(0 < N_f < N_l\), such that we can divide the \(2N_l + N_f\) nodes into two groups, group X with \(2N_l\) nodes and group Y with \(N_f\) nodes that satisfy the following conditions: 1) If we label the nodes in group X sequentially from 0 to \(2N_l - 1\), then there is no chord between a node with even label and a node with odd label. 2) The path on the cycle that connects any two nodes in group Y has even number of nodes in group X. 3) There is no chord connecting any node in group Y with the rest of nodes in the cycle.

**Proof.** Similarly as that in the proof of Theorem 4.3.2, this proof comprises of the following two steps, where the details can be found in Section C.7.

**Step 1:** we show that for any network topology \(G = \{N, \mathcal{E}\}\), \(\mathcal{D}_{MMb}(G) \subset \mathcal{D}_{FMb}(G)\) if and only if there exists \(S_1 \in \mathcal{S}_{FMb}\) such that \(G_R(S_1)\) contains a cycle with odd length and at least one node in that cycle is operating in mode 3.

**Step 2:** we show that there exists \(S_1 \in \mathcal{S}_{FMb}\) such that \(G_R(S_1)\) contains a odd length with \(N_l + N_f\) nodes, and \(N_f > 0\) nodes in the cycle are operating in mode 3 if and only if \(G\) contains a cycle with \(2N_l + N_f\) nodes, where \(N_l + N_f\) is an odd number, and \(N_f < N_l\), such that we can divide the \(2N_l + N_f\) nodes into two groups, group X with \(2N_l\) nodes and group Y with \(N_f\) nodes that satisfy the following conditions: 1) If we label the nodes in group X sequentially from 0 to \(2N_l - 1\), then there is no chord between a node with even label and a node with odd label. 2) The path on the cycle that connects any two nodes in group Y has even number of nodes in group X. 3) There is no chord connecting any node in group Y with the rest of nodes in the cycle.

\[\square\]

76
Corollary 4.3.3.1. In a tree network topology, $\textit{FM}_a/\textit{FM}_b$ constraint set and $\textit{MM}_a/\textit{MM}_b$ constraint set lead to the same throughput region. In other words,

$$D_{\textit{MM}_a} = D_{\textit{FM}_a} = D_{\textit{MM}_b} = D_{\textit{FM}_b},$$

when the network topology does not contain any cycle.

Remark 4.3.3.1. This Corollary follows directly from Theorem 4.3.2 and Theorem 4.3.3. The result implies that when the network topology formed by the interference relationship between wireless nodes does not contain any cycles, MIMO multiplexing alone can support the largest throughput region. In other words, neither multi-user MIMO nor full-duplex communication can provide any throughput gain when MIMO multiplexing is supported in every nodes.

Remark 4.3.3.2. It should be noted that Theorems 1-3 establish the complete characterization of the relationship between full duplex and MIMO, and provide us with a very clear guideline on which architectures and traffic patterns could result in improvement for one technology over another.

4.4 Experimental Verification

In the previous sections, for analytical tractability, we focused on a binary interference model. This section presents results from an experimental testbed with software-defined radios for two purposes; (i) to verify if the throughput region claims are true in practice, and (ii) to present throughput comparison results as the SNR between nodes varies. Our results, indeed, corroborate the findings from the previous sections.
4.4.1 Experimental setting

Setup

The nodes in our experiments are equipped with NI PXIe 1082 chassis [39] with two RF chains. Each RF chain consists of an XCVR 2450 (RF front end), an NI-5781 (data converter module) and an NI PXIe-7965R (Xilinx Virtex-5 FPGA). The nodes are capable of communicating at 2.4GHz wide-band. The supported carrier-level synchronization between RF chains from this platform helps with the MIMO and full-duplex implementation. We implement the OFDM transceivers for each RF chain. The communication system supports QPSK modulation with various coding rates (convolutional code 1/2, 2/3, 3/4).

The full-duplex implementation follows the design from [7], and the MIMO multiplexing implementation is based on the standard 802.11n protocol. For multi-user MIMO implementation, since the channel coefficient is not available at the transmitters, our system is effectively a multi-user MIMO system with the channel state information at the receiver (CSIR). Carrier-level synchronization is the only requirement in this case. LTE systems achieve this synchronization by having the receivers broadcast poll messages to which the transmitters synchronize [40]. However, this poll-based technique is not enough for our system since the synchronization has to be achieved across multiple collision domains. For instance, in the example shown in Fig. 4.3(f), nodes A and D need to be synchronized for the multi-user MIMO decoding at node E. At the same time, synchronization is also required for nodes B and D at receiver C. Thus, a receiver-initiated synchronization would result in a conflict at node D, which receives this poll from both E and C. To address this problem, one extra node in the experiment is assigned as reference node. It broadcasts, at a higher power, a known sequence as a reference signal. All the other nodes adjust their frequency offset and starting time according to this broadcast signal. Note that there
are many ways to achieve this synchronization, e.g., by using GPS signals or cellular control signals. Our goal is to compare different multi-RF chain configurations without introducing implementation specific biases.

Environment
Our results are evaluated in an indoor environment. Specifically, the experiment is conducted inside our Computer Science department building, where there are multiple metallic cubicles. As the transmission is closely related to the conflict relationship of all the involved nodes, we use low transmission power (around -20dBm) and use walls made out of metal and bricks as blocks among nodes to create the desired interference map.

We operate at 2.437Ghz center frequency at night, as there is less interference. During the experiment, we find out that except for some periodical broadcast messages from cohabiting WiFi APs, nearly all of the interference signals are from within our experimental testbed.

Method
To match the channel condition with available coding rates, we have a pre-calibration phase before the execution of experiment.

Initial Calibration For each connected link in the network, we use the lowest available data rate (1/2 QPSK) under basic SISO transmission while keeping all the other links silent. We adjust the transmission power to the value where the packet reception ratio is just about to decline. The borderline transmission power is referred to as the link’s reference power. Instead of choosing the optimal coding scheme for the observed SNR, this calibration allows us to pick the SNR for which the chosen coding scheme is optimal.
**Execution**  For each feasible schedule, we conduct our experiment under a set of 30 different channel conditions, where each channel condition is created by randomly and independently choosing a transmit power for each link with a value anywhere between its reference power and reference power + 10dB. For every link, each channel condition and each available channel coding rate, 100 packets are transmitted. For any specific link, the coding rate that maximizes the throughput will be adopted as the coding setting for the link. Following the above steps it can be guaranteed that the selected coding rate matches the channel condition.

**Metric**

We omit carrier sensing and MAC overheads in the experiment as they are MAC protocol specific. All of the transmissions are unidirectional for a certain link. For each link, we define the link-rate under a certain schedule as the product of PER (packet error rate) and physical layer data-rate on that link, and the network throughput is defined as the sum of the long term average link throughput on each link.

4.4.2 Throughput result

We verify the throughput relationship between different constraint sets in a real experiment setup as described. Specifically, we build the network with topology shown in Fig. 4.3 here.

For every set of channel conditions, we conduct the experiment for four constraint sets: SS, FD, MMA, and FMb under two different scheduling configurations. The two scheduling configurations is shown in Fig. 4.5:

- **Config 1:** In this configuration, all the three links \((D, C), (B, A)\) and \((A, E)\) are guaranteed to transmit for the same amount of time.
Figure 4.5: Scheduling decision of two configurations for four different constraint sets: SS, FD, MMa, and FMb.

- **Config 2**: In this configuration, we pick a fixed schedule for each constraint set, as indicated by the red solid dot in Fig. 4.5.

Figs. 4.6a and 4.6b present the CDFs of the network throughput with respect to the set of 30 different channel conditions under the two scheduling configurations, respectively. There are some points we want to highlight here:

- The full-duplex constraint set makes the major difference for the two figures. This is expected theoretically. As shown in Fig. 4.5, the average number of data streams of full-duplex constraint set is 1.5 for config 1 while the one for config 2 is 2.
Figure 4.6: CDFs of the network throughput w.r.t. 30 different channel conditions. (a) and (b) are the results corresponding to config 1 and 2 shown in Fig. 4.5, respectively.

- In both Figs. 4.6a and 4.6b, the throughput relationship supports our theoretical result: The throughput of FMb is on average 27% higher than MMa constraint set while the theoretical result shows a 50% difference. This gap is mainly due to the residual signal after full-duplex cancellation and the inaccuracy of the synchronization regarding the multi-user MIMO communications. At the same time, note that when the SINR is high, the ratio of the throughput between MMa and FMb is approaching 1.5, which is close to the theoretical result.

4.5 Conclusion

In this chapter, we investigated the achievable throughput performance in wireless networks where each node can simultaneously activate two RF chains with no channel state information at the transmitter. There are three wireless technologies that can take advantage of the simultaneous activation of two RF chains, namely MIMO
multiplexing, multi-user MIMO, and wireless full-duplex communication. We compared the throughput region achieved by some combinations of the three wireless technologies for a general network topology under a binary interference model. Our results provide a clear guideline on which architecture and traffic pattern could result in throughput improvement for one technology over another.

There are many interesting future directions that one can investigate: 1) Instead of considering the simplified throughput definition and the idealized binary interference model, we would like to extend the analysis to accommodate the more accurate SINR-based interference model. 2) Investigate the throughput performance in a general network where channel state information is provided at the transmitter. Here, a node in multi-user MIMO broadcasting can precode its two transmitted streams such that only one receive antenna (one active receive RF chain) is required at the two receivers each instead of two. 3) Generalize our results to the case when each node can have more than two active RF chains and different nodes potentially have different number of active RF-chains. 4) Instead of only focusing on the throughput region achieved by pure scheduling, i.e., interference avoidance, we would like to incorporate our analysis with more advanced physical layer technologies such as interference alignment [30] and cut-through transmission [9].
CHAPTER 5
SCHEDULING IN WIRELESS NETWORKS WITH FULL-DUPLEX CUT-THROUGH TRANSMISSION

5.1 Introduction

Recent development in wireless radio technology shows that by using advanced signal processing techniques together with new RF circuit designs, a wireless device can transmit and receive on the same frequency band, achieving full-duplex transmission [41–43]. The key idea that enables this full-duplex capability is that a node has complete knowledge of the digital packet it is transmitting, and therefore, it could potentially predict and actively cancel the impact of the transmitted signal onto its own receive antenna.

The full-duplex technology eliminates the conventional constraint in wireless networking that a node can either transmit or receive at any time but not both for any frequency band, and as a result it enlarges the capacity region of wireless networks. Take the five node tandem network shown in Figure 5.1 as an example. Assume that the nodes interfere only with their closest neighbors, and there exists a single flow from node A to node E. Under the half-duplex constraint, each link along the route of the flow can be activated only one third of the time. However, when the nodes are capable of canceling their self interference, i.e., they are full-duplex enabled, then each link can be activated half of the time, as shown in Figure 5.1(a). Further, since
there is only a single traffic flow from node $A$ to node $E$, we know that any packet transmitted by node $C$ is previously received by $C$ from its upstream node $B$. If we assume that node $B$ keeps a copy for every packet that it has received, then it has the potential to decode any packet that is collided with the transmission from node $C$.

We use the term *cross interference cancellation* to describe the case when a node can withstand a stream of interference that carries a known packet. From Figure 5.1(b) we can see that each link can be activated half of the time if the nodes are capable of performing cross interference cancellation.

Interestingly, as is evident from Figure 5.1(c), if the nodes are capable of canceling both their self interference and a stream of known cross interference, then all the link can be activated at the same time, effectively forming a *cut-through* route with each node along the route simultaneously receiving a new packet from its upstream node and forwarding a previously received packet to its downstream node. The concept
of wireless cut-through transmission was first envisioned in [41], in which it was argued that as a full-duplex node starts to receive a packet it can simultaneously start to forward it without having to decode the entire packet, while the downstream interference can be digitally canceled since the interference is simply a delayed version of the received packet. A preliminary implementation is presented in [44]. In this chapter, we make a somewhat less restrictive assumption that a full-duplex node can only start to forward a packet after the packet is completely decoded, and that it has the capability to cancel a single stream of interference from one of its immediate neighbors given that the neighbor is forwarding a packet previously received from it.

While wireless cut-through transmissions is a promising way to increase the capacity of the network, as we will see shortly, it also introduces new challenges in the design of MAC layer dynamic scheduling policies.

In a multi-hop wireless network, it is critical to have a resource allocation algorithm that efficiently utilizes the limited wireless spectrum. The seminal work of [10] develops a joint dynamic routing and scheduling algorithm, namely back-pressure, which is proven to be throughput-optimal, i.e., it can stabilize any network load that can be stabilized by some joint routing and scheduling algorithm. Later, through a utility maximization framework [11], it is shown that the wireless resource allocation problem can be optimally decomposed into three parts: transport layer rate control, network layer routing, and MAC layer scheduling, with minimal coupling among the layers. This cross-layer decomposition suggests that the MAC layer scheduling component is the bottleneck of this problem, as it requires solving a difficult combinatorial optimization problem which is NP-complete in general. There has been a plethora of work that focus on devising low-complexity and/or distributed scheduling algorithms. At the same time, with the continuous evolution of wireless physical layer technologies, many new transmission schemes emerge, such as wireless full-duplex,
interference alignment, distributed multi-user MIMO, noisy network coding, etc. On the one hand, these new schemes keep breaking conventional transmission constraints and expanding the rate region of wireless networks. On the other hand, they inherently come with more sophisticated interference relationship among wireless links and bring many challenges in the design of scheduling algorithms. Regarding full-duplex cut-through transmission, the challenges are the following: (i) the cut-through route naturally involves links that are multiple hops away from each other, which makes it hard to design scheduling algorithms based only on local information at each node. (ii) the MAC layer scheduling decision becomes closely coupled with the network layer routing decision, as a cut-through route is usually formed to serve a specific flow.

Our contributions are as follows:

- We introduce a novel way to model the interference relationship in wireless networks with full-duplex cut-through capability, which decouples the routing decision from the scheduling decision in a scalable and efficient manner.

- A queue-length based CSMA-type algorithm, similar to the one in [45], is proposed, which can dynamically form/change full-duplex cut-through routes in the network and achieve throughput-optimality.

The rest of the chapter is organized as follows: In Section 5.2, we introduce the network model, review the back-pressure algorithm under half-duplex or full-duplex networks, and then explain the difficulty in devising a scheduling algorithm for networks with cut-through capability. In Section 5.3, we propose a new and efficient method of modeling the interference relationship for cut-through transmission. In Section 5.4, we develop a throughput-optimal queue-length based CSMA-type algorithm that leverages the full-duplex cut-through capability. The algorithm is evaluated in Section 5.5 and the chapter is concluded in Section 5.6.
5.2 System Model

5.2.1 Network model and half-duplex/full-duplex constraint

We consider a multi-hop wireless network that can be described by a network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ together with an interference graph $\mathcal{G}_I = (\mathcal{V}, \mathcal{E}_I)$, where $\mathcal{V}$ denotes the set of wireless nodes, $\mathcal{E}$ denotes the set of wireless links, and $\mathcal{E}_I$ denotes the interference relationship between wireless nodes. For any two nodes $A, B$ in $\mathcal{V}$, $(A, B) \in \mathcal{E}_I$ if the transmission of node $A$ interferes with the reception of node $B$, while $(A, B) \in \mathcal{E}$ if a direct data-link can be established from node $A$ to node $B$. In other words, $\mathcal{E}$ captures the communication region of every node in the network and $\mathcal{E}_I$ captures the interference region the nodes in the network. We assume that $\mathcal{E}$ is a subset of $\mathcal{E}_I$, and the edges in both $\mathcal{G}$ and $\mathcal{G}_I$ are bidirectional. An example network is shown in Figure 5.2

![Interference graph and Data-link graph](image)

Figure 5.2: The interference graph $\mathcal{G}_I$ and the data link graph $\mathcal{G}$ of an example wireless network with five nodes. $\mathcal{I}(B) = \{A, C, D\}$.

For any node $A$ in $\mathcal{V}$, we denote $\mathcal{I}(A) \triangleq \{B | (A, B) \in \mathcal{E}_I\}$ as the set of nodes that

---

1The use of an interference graph gives a more accurate modeling of the interference relationship between wireless nodes, compared with the conventional link-centric hop-count based interference model
interfere with node \( A \). For any subset \( S \) of data-links in \( E \), we denote \( R(S) \) and \( T(S) \) as the set of receiver nodes and the set of transmitter nodes under the subset of links \( S \), respectively. More precisely, \( R(S) = \{ B | (A, B) \in S \} \) and \( T(S) = \{ A | (A, B) \in S \} \). We also call a subset of data links \( S \) a schedule.

Given these notations, we can describe the half-duplex interference constraint in a wireless network as follows:

**Definition 5.2.1** (Half-duplex feasibility conditions). A subset \( S \) of links in \( E \) is half-duplex feasible if for any link \((A, B)\) in \( S \), all of the following conditions hold:

1. \( I(A) \cap R(S) = \{ B \} \)
2. \( I(B) \cap T(S) = \{ A \} \)
3. \( A \notin R(S) \) and \( B \notin T(S) \)

We denote the set of all half-duplex feasible schedules in the network as \( S_{HD} \). For any \( S \in S_{HD} \) and any link \((A, B)\) in \( S \), the first condition in the above definition implies that there is no receiver node within node \( A \)'s interference range, other than node \( A \)'s intended receiver \( B \). Similarly, the second condition says that there is no transmitter node within node \( B \)'s interference range, except its intended transmitter \( A \). While the first two constraints guarantee that there is no cross-interference between links, the last constraint makes sure that there is no self-interference in \( S \), i.e., any node under \( S \) cannot be both a transmitter and a receiver.

From the half-duplex feasibility conditions, it is straightforward to derive the full-duplex feasibility conditions, which are described in Definition 5.2.2, since we only need to remove the last condition in Defintion 5.2.1 that forbids the existence of self-interference in a schedule.

**Definition 5.2.2** (Full-duplex feasibility conditions). A subset \( S \) of links in \( E \) is full-duplex feasible (denote as \( S \in S_{FD} \)) if for any link \((A, B)\) in \( S \), all of the following conditions hold:
\[ I(A) \cap R(S) = \{ B \} \quad \text{•} \quad I(B) \cap T(S) = \{ A \} \]

### 5.2.2 Traffic model

Let \( \mathcal{F} \) denote the set of data flows in the network. For any flow \( f \) in \( \mathcal{F} \), we further denote the source and the destination node of that flow as \( f_s \) and \( f_d \), respectively. We assume that each node maintains a set of next-hop nodes for each flow, in such a way that there is no loop for any flow\(^2\). Based on this routing table, for each data link \( (A, B) \in \mathcal{E} \), we define \( \mathcal{F}_{(A,B)} \) as the set of flows that it carries. In other words, \( \mathcal{F}_{(A,B)} \) denotes the set of flows that have node \( B \) as a valid next-hop from node \( A \).

We assume a time-slotted system, where each data-link, if scheduled at a certain time-slot, can transmit a single packet. We denote \( a^f[t] \) as the number of packets that arrive at the source node of flow \( f \) at time-slot \( t \), \( S[t] \) as the set of scheduled links at time-slot \( t \), and \( f_{(A,B)}[t] \) as the index of the flow that link \( (A, B) \) chooses to serve if it is scheduled at time-slot \( t \). We also assume that each node keeps a queue for each flow, and denote \( Q^f_A[t] \) as the queue length of flow \( f \) right before the start of the \( t \)th time-slot. Based on the model, we know that the queue \( Q^f_A \) evolves as

\[
Q^f_A[t] = Q^f_A[t - 1] + a^f[t - 1]1(f_s = A) + \sum_{B: (B,A) \in S[t-1]} 1(f = f_{(B,A)}[t - 1]) - \sum_{B: (A,B) \in S[t-1]} 1(f = f_{(A,B)}[t - 1])
\]

if \( f_d \neq A \), and \( Q^f_A[t] = 0 \) otherwise, where \( 1(.) \) is the indicator function.

Let us assume that the arrival process is i.i.d. across different flows, and denote the arrival rate of flow \( f \) as \( \lambda_f \). The capacity region of the network is then defined as

\(^2\)The assumption that there is no loop in the routing table does not preclude the possibility for there to be cycles in the network graph...
the set of rate vectors $\bar{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_{|F|}]$ under which all the queues in the network can be stabilized\footnote{We assume that the queueing dynamic can be captured by a Markovian process and stability refers to the Markov chain being positive recurrent \cite{46}} by some scheduling policy. An algorithm is called \textit{throughput-optimal} if it can stabilize the queues in the network for any arrival rates within the capacity region. It is well known from the seminal work \cite{10} that a joint routing and scheduling algorithm, called back-pressure algorithm, is throughput optimal. The algorithm is restated below:

\textbf{Routing:} $f_{(A,B)}[t] = \arg \max_{f \in \mathcal{F}_{(A,B)}} \left( Q^f_A[t - 1] - Q^f_B[t - 1] \right)$

$W_{(A,B)}[t] = \max_{f \in \mathcal{F}_{(A,B)}} \left( Q^f_A[t - 1] - Q^f_B[t - 1] \right)$

\textbf{Scheduling:} $S[t] = \arg \max_{S \in S} \sum_{(A,B) \in S} W_{(A,B)}[t], \quad (5.2.1)$

where $S$ can either be $S_{HD}$ or $S_{FD}$. A nice feature of the above algorithm is that there exists only a loose coupling between the routing decision and the scheduling decision: the scheduling component only needs to obtain the value of the maximum queue differential $W$, and operates irrespective of which flow gets served on each link. The fundamental reason for the loose coupling is that the routing decision does not affect the physical layer interference relationship of the data-links in the network, and therefore does not alter the MAC layer rate region of the network. While this is true for both half-duplex and full-duplex wireless network, this claim no longer holds for wireless networks with full-duplex cut-through capability. Indeed, there exists a direct coupling between the routing decision and the physical layer capability of the network.
To support our claim in the previous paragraph, let us look at the network shown in Figure 5.3(a). There are two flows running on the network, where flow 1 runs from node $A$ to node $E$, and flow 2 runs from node $D$ to node $E$. Now assume that the two links $(A, B)$ and $(B, D)$ are already activated to serve flow 1, in which case node $B$ is in full-duplex mode, then, whether link $(D, E)$ can be activated together with $(A, B)$ and $(B, D)$ or not depends on which flow it chooses to serve: if it picks flow 1, then node $B$ is capable of canceling the cross-interference from node $D$ and form a cut-through route, since the packet transmitted by node $D$ is previously received by node $B$. If it, on the other hand, picks flow 2, then the link cannot be activated together with $(A, B)$. Therefore, whether a link can be scheduled or not depends on which flow it chooses to serve, leading to a direct coupling between the routing decision and the scheduling decision. Furthermore, if the flows have multiple routes,

![Figure 5.3](image)

Figure 5.3: There are two flows that run on the network shown in Figure 5.2. The links $(A, B)$ and $(B, D)$ are already scheduled to serve flow 1. (a) Both flow 1 and flow 2 have only a single path. (b) Flow 1 has two alternative paths.
then whether a link can be scheduled or not may even depends on which packet it chooses to transmit. For example, in the scenario shown in Figure 5.3(b) where there are two paths for flow 1, link \((D, E)\) can be activated simultaneous with \((A, B)\) and \((B, D)\) only when it chooses to serve flow 1 with a packet it received previous from node \(B\).

Given the inherent direct-coupling of routing and scheduling decisions and the complicated flow-dependent interference relationship between data-links in the network, two natural questions arise: (i) Is there a way to efficiently model the interference relationship among data-links in wireless networks with full-duplex cut-throughput transmission capability, and at the same time circumvent the problem of the coupling between routing and scheduling? (ii) How to devise a distributed algorithm that dynamically forms/changes cut-through routes based on the patterns and the arrival-rates of traffic flows? We answer these two key questions in the rest of the chapter.

5.3 Modeling of interference relationship with full-duplex cut-through capability

In the previous section, we showed that cut-through transmission introduces a direct-coupling of the routing decision with the schedule decision. From the point of view of the back-pressure algorithm, this coupling means that the set of all feasible schedules \(\mathcal{S}\) in Equation (5.2.1) at time-slot \(t\) is a function of the set of routing decisions \(\{f_{(A,B)}[t]\}_{(A,B)\in \mathcal{E}}\).

In this section, we will focus on developing feasibility conditions that incorporate the routing decision on each link. The attempt is to restore the decoupled structure shown in Equation (5.2.1) by revising the state-space of feasible schedules. We will
first discuss two straightforward, yet inefficient and unscalable, methods, and then propose a new and efficient way to describe the cut-through interference relationship using a \textit{neighbor-expanded graph}.

5.3.1 Modeling using per-flow virtual links

From the previous section, we showed that even under the case when each flow has a single route, the physical layer interference relationship among data-links is dictated not only by their topological relationship in the network graph, but also by which flow each link chooses to serve. Then, a straightforward way to construct the interference relationship is to extend each data-link into a number of virtual links, with each virtual-link designated to a single specific flow. More precisely, we can denote the virtual-link of $(A, B)$ that is committed to a flow $f$ as a triplet $(A, B, f)$, and thus the set of data-links is extended to the set of virtual links $\{(A, B, f)\mid (A, B) \in \mathcal{E}, f \in \mathcal{F}_{(A,B)}\}$, based on which a feasibility condition of cut-through transmission can be derived.

The drawbacks of this approach are clear. First, this approach works only when each flow has only a single fixed route. Second, the state space of the virtual links can be very large, depending on the number of flows that get admitted. Third, the interference relationship among virtual links has to be re-derived each time new traffic flows enter the system.

5.3.2 Modeling using per-route virtual links

This approach exploits the fact that cut-through transmission can activate a group of consecutive data-links to serve a flow as if the whole group of data-links is a single link. We can exhaustively find all possible routes from any node to any other node
in the network, and view each route we find as a per-route virtual link. A per-
route virtual link can be denoted as a sequence of nodes that the route it represents
traverses. For example, we can denote the route that corresponds to the path 1 of
flow 1 in Figure 5.3(b) as a virtual link \((A, B, D, E)\).

The drawbacks of this approach are also clear. First, an exhaustive search of
all possible routes in the network is required every time there is a topology change.
Second, the number of all possible routes in a network increases exponentially as
the network size grows, which makes this approach unscalable and only suitable for
small networks. Third, since links several hops away may form a single virtual link
and have to be activated simultaneous when that virtual link is scheduled, this per-
route virtual link is not suitable for distributed implementations of the scheduling
algorithm, as a message passing mechanism has to be added to make sure that two
ends of a virtual link coordinates with each other.

5.3.3 Modeling using neighbor-expanded graph

Now we propose a novel and scalable way to describe the interference relationship in
wireless networks with cut-through capability. The key is to realize that full-duplex
cut-through capability is no more than the capability for each node to cancel its self
interference and a stream of cross interference, if that cross interference carries a
packet the node has previously received. Given our assumption that there is no loop
in the routing table for any routes, we observe that node \(A\) can cancel a stream of
cross-interference from node \(B\), if \(B\) transmits a packet that it has previously received
from node \(A\). Based on this observation, we can think of node \(B\) as a cluster of sub-
nodes denoted as \(\{B_C\mid (C, B) \in \mathcal{E} \text{ or } C = B\}\), where sub-node \(B_A\) stores the packets
node \(B\) has obtained directly from node \(A\) (\(B_B\) stores the packets that exogenously
arrived at node \(B\)). In this case, node \(A\) can cancel a stream of cross interference
from node $B$ only if $B_A$ is the activated sub-node among all the sub-nodes of node $B$. According to the definition of sub-node, if link $(A, B) \in E$ is scheduled, $B_A$ is the activated receiver sub-node.

Given the concept of per-neighbor sub-node, we can construct an extended data-link graph, which we call the neighbor-expanded graph. Specifically, we denote the neighbor-expanded graph as $\hat{G} = (\hat{V}, \hat{E})$, where

$$\hat{V} = \{A_C | (C, A) \in E \text{ or } C = A\},$$
$$\hat{E} = \{(A_C, B_A) | A_C \in \hat{V}, \text{ and } (A, B) \in E\}.$$

Figure 5.4 shows the neighbor-expanded graph of the network shown in Figure 5.2. Note, that the wireless network can now be fully represented by $G_I$ and $\hat{G}$. For the rest of the chapter, we reserve the term schedule and the notation $S$ only to refer to a subset of links in $\hat{E}$. For any schedule $S$, we define, similar as before,

$$R(S) = \{B | (A_C, B_A) \in S \text{ for some } C, A \in V\}$$
$$T(S) = \{A | (A_C, B_A) \in S \text{ for some } C, B \in V\}$$
$$\hat{T}(S) = \{A_C | (A_C, B_A) \in S \text{ for some } B \in V\}.$$

By the help of the neighbor-expanded graph, we can derive the feasibility condition under full-duplex cut-through transmission as the following.

**Definition 5.3.1** (Cut-through feasibility conditions). A subset $S$ of links in $\hat{E}$ is cut-through feasible (denote as $S \in S_{CT}$) if for any link $(A_C, B_A)$ in $S$, all of the following conditions hold:

(C1) $I(A) \cap R(S) = \{B\}$ or $I(A) \cap R(S) = \{B, C\}$ with $I(C) \cap T(S) = \{A\}$

(C2) $I(B) \cap T(S) = \{A\}$ or $I(B) \cap T(S) = \{A, D\}$ for some $D$ with $D_B \in \hat{T}(S)$. 

96
Figure 5.4: Neighbor-expanded graph of the data-link graph shown in Figure 5.2(b).

(C3) \( (A_D, B_A) \notin S \) for any \( D \in V \setminus \{C\} \)

By comparing the above definition with Definition 5.2.2, we can see that the full-duplex feasibility conditions are a subset of the cut-through ones. Condition (C1) in the above definition implies that \( (A_C, B_A) \) may be activated even when \( C \) is a receiver node, given that \( A \) is the only transmitter among all \( C \)'s neighbors. Condition (C2) says that \( (A_C, B_A) \) may be activated even when one of \( B \)'s neighbors \( D \) is a transmitter node, given that \( D_B \) is the activated sub-node (which means the packet that \( D \) transmits is received from node \( B \)). The last condition implies that there cannot exist two simultaneously activated transmitter sub-nodes that belong to the same node.

Based on Definition 5.3.1, in Figure 5.4, it is easy to check that the set of links \{\( (A_A, B_A), (B_A, D_B), (D_B, E_D) \)\} can be activated simultaneously, while \{\( (A_A, B_A), (B_A, D_B), (D_D, E_D) \)\} and \{\( (A_A, B_A), (B_A, D_B), (D_C, E_D) \)\} are not feasible schedules. This can be mapped back to our previous examples in Figure 5.3, where we claim that

97
link \((D, E)\) can be activated together with \((A, B)\) and \((B, D)\) only when \(D\) serves a flow 1 packet that it previous obtained from node \(B\).

There are two advantages in using neighbor-expanded graph to characterize the interference relationship, compared with the other two approaches. (i) since the number of sub-nodes that a node has in the neighbor-expanded graph equals the degree of that node in the data-link graph, we know that the size of \(\hat{V}\) and \(\hat{E}\) is bounded by the size of \(V\) and \(E\) times the maximum degree of the data-link graph, which is quite scalable as the network size grows. (ii) the feasibility constraint is not a function of the routing decision, nor does it involve links that are more than 2-hop away from each other, which not only restores the decoupled structure of the back-pressure algorithm, but also permits the development of distributed scheduling algorithms.

Since the minimal scheduling entity is now the sub-node, in order to apply the back-pressure algorithm in Equation (5.2.1), for every sub-node \(A_C\) and any flow \(f\), we need to keep a packet queue \(Q^f_{A_C}\). For the rest of the chapter, we will use \((A_C, B_A)\) and \(l\) interchangeably to index the links in \(\hat{E}\).

### 5.4 Queue-length based Cut-through CSMA algorithm

In this section, we develop a queue-length based CSMA algorithm, similar to the one proposed in [45], that can achieve throughput optimality in wireless networks with cut-through capability.

We should point out that there exists an important difference between the interference model adopted in [45] and the interference relationship we have derived in Definition 5.3.1 for full-duplex cut-throughput enabled network, which prohibits a direct application of the algorithm in [45] onto the cut-through scenario. The dynamic CSMA algorithm developed in [45] depends on the assumption that the interference
relationship between different links can be captured by an conflict graph\(^4\) with each independent set being a feasible schedule. However, under our cut-through feasibility conditions, such a conflict graph cannot be formed. In other words, given three links \(l_1, l_2, \) and \(l_3,\) the assumption that any two links can be activated together does not necessarily imply that the three links form a feasible schedule. For example, in Figure 5.5, among the three links \((A_A, B_A), (C_B, D_C),\) and \((E_B, F_E),\) any two can be activated at the same time, however, a schedule that include all three of them would violate condition (C2) in Definition 5.3.1, since node \(B_A\) can only cancel one stream of cross-interference. We circumvent this obstacle by introducing the concept of a trimmed decision schedule, whose definition is provided in Definition 5.4.2.

\[\text{Figure 5.5: An example neighbor-expanded graph.}\]

To differentiate between different cut-through feasible schedules in \(S_{\text{CT}},\) we use a subscript to denote the index of a specific schedule. For example, \(S_{(x)}\) is the \(x^{\text{th}}\) schedule in \(S_{\text{CT}}.\)

**Definition 5.4.1** (Decision schedule). For any \(M \subseteq \hat{E},\) we say that \(M\) is a valid decision schedule if for any \((A_C, B_A) \in M,\) all of the following conditions hold:

\(^4\text{In a conflict graph (also called link-contention graph), the vertices represent data-links and an edge between two vertices indicates that the two data-links cannot be activated simultaneously.}\)
• $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{M}) = \{B\}$  
  $$\mathcal{I}(B) \cap \mathcal{T}(\mathcal{M}) = \{A\}$$

• $(A_D, B_A) \notin \mathcal{M}$ for any $D \in V\setminus\{C\}$

**Definition 5.4.2** (Trimmed decision schedule). For any cut-through feasible schedule $\mathcal{S}_x \in \mathcal{S}_{CT}$ and any valid decision schedule $\mathcal{M}$, we define a new schedule $\mathcal{M}_x = \mathcal{M}_1(x) \cup \mathcal{M}_2(x) \cup \mathcal{M}_3(x) \cup \mathcal{M}_4(x)$, where

\[
\mathcal{M}_1(x) = \left\{(A_C, B_A) \in \mathcal{M} \cap \mathcal{S}_x \mid \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_x) = \{B\}\right\},
\]

\[
\mathcal{M}_2(x) = \left\{(A_C, B_A) \in \mathcal{M} \cap \mathcal{S}_x \mid \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_x) = \{B, C\}, \mathcal{I}(C) \cap \mathcal{T}(\mathcal{M}) = \{A\}\right\},
\]

\[
\mathcal{M}_3(x) = \left\{(A_C, B_A) \in \mathcal{M} \setminus \mathcal{S}_x \mid \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_x) = \emptyset, \mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_x) = \emptyset \text{ or } \{D\} \text{ for some } D \text{ with } D_B \in \hat{T}(\mathcal{S}_x)\right\},
\]

\[
\mathcal{M}_4(x) = \left\{(A_C, B_A) \in \mathcal{M} \setminus \mathcal{S}_x \mid \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_x) = \{C\}, |\mathcal{I}(C) \cap \mathcal{T}(\mathcal{S}_x)| = 1, \mathcal{I}(C) \cap \mathcal{T}(\mathcal{M}) = \{A\}, \mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_x) = \emptyset \text{ or } \{D\} \text{ for some } D \text{ with } D_B \in \hat{T}(\mathcal{S}_x)\right\}.
\]

Since $\mathcal{M}_x$ is a subset of $\mathcal{M}$ and also a function of $\mathcal{S}_x$, we say that $\mathcal{M}_x$ is the result of $\mathcal{M}$ trimmed by $\mathcal{S}_x$.
From the above definitions, we can see that whether a link in the decision schedule should be trimmed or not depends on the status of the links that are no more than two-hops away. Given the definitions of the decision schedule and trimmed decision schedule, we obtain the following two lemmas, whose proofs can be found in Section D.1 and D.2.

**Lemma 5.4.1.** For any cut-through feasible schedule $S(x) \in \mathcal{S}_{CT}$ and any valid decision schedule $M$, $S(x) \cup M(x)$ is a cut-through feasible schedule.

**Lemma 5.4.2.** For any two cut-through feasible schedules $S(y), S(z) \in \mathcal{S}_{CT}$ and any valid decision schedule $M$, if $S(y) \cap M(y) = S(z) \cap M(y)$, then $M(y) = M(z)$.

Now we are ready to introduce the algorithm. Let $\mathcal{M}$ be a set of valid decision schedules. At the beginning of each time-slot, the system chooses a decision schedule $M \in \mathcal{M}$ with probability $P_M$, where $P_M > 0$ for any $M \in \mathcal{M}$ and $\sum_{M \in \mathcal{M}} P_M = 1$. Assume, w.l.o.g., that the schedule used in the network at the $(t - 1)^{st}$ time-slot is $S[t - 1] = S(y)$ for some $y$. At the start of time-slot $t$, right after a decision schedule $M$ is picked, the system obtains a trimmed schedule $M(y)$ as a function of $M$ and $S(y)$. For any link $l \in M(y)$, it is included in the updated schedule $S[t]$ with probability $P_l$ and rejected with probability $1 - P_l$, while any link in $S(y) \cap M(y)$ is included in $S[t]$. From the Lemma 5.4.1 we know that $S[t]$ is a cut-through feasible schedule, since $S[t]$ is a subset of $S(y) \cup M(y)$. This algorithm is summarized in Algorithm 1. All the steps in the algorithm rely solely on local information and thus permit a distributed implementation.

Along with the same lines as in [45, 47], we can obtain the following proposition.
Algorithm 1: Basic Cut-through CSMA Algorithm

1 Assume that the schedule at time slot $t - 1$ is $S[t - 1] = S_{(y)}$

Before the start of time-slot $t$:
2 Randomly pick a decision schedule $M$ from $\mathbb{M}$ with probability $P_M$.
3 The picked decision schedule $M$ is then trimmed by $S_{(y)}$ to form a trimmed schedule $M_{(y)}$.

During time-slot $t$:
4 for any link $(A_C, B_A) \in \hat{E}$ do
5 If $(A_C, B_A)$ is in $M_{(y)}$, then with probability $P_{(A_C, B_A)}$, the link gets included in $S[t]$.
6 If $(A_C, B_A)$ is in $S_{(y)} \setminus M_{(y)}$, then the link is included in $S[t]$.

Proposition 5.4.1. \{$S[t]\}_t$ in Algorithm 1 evolves as an irreducible and aperiodic Markov chain with the state space being $S_{CT}$, if the set of decision schedules $\mathbb{M}$ satisfies

$$\bigcup_{M \in \mathbb{M}} \{(A_C, B_A) \in M | I(C) \cap T(M) = \{A\}\} = \hat{E}.$$ 

If the above condition is satisfied, then the stationary distribution of the Markov chain $\{S[t]\}_t$ is

$$\pi \left( S_{(x)} \right) \triangleq P \left( S[\infty] = S_{(x)} \right) = \frac{1}{Z} \left( \prod_{l \in S_{(x)}} P_l \right) \left( \prod_{l \notin S_{(x)}} (1 - P_l) \right),$$

where $Z = \sum_{S_{(y)} \in S_{CT}} \left( \prod_{l \in S_{(y)}} P_l \right) \left( \prod_{l \notin S_{(y)}} (1 - P_l) \right)$.

The stationary distribution of the cut-through feasible schedule in Proposition 5.4.1 is obtained by verifying the local balance equation of the Markov chain $\{S[t]\}_t$. The result in Lemma 5.4.2 is essential in guaranteeing that the Markov chain that describes Algorithm 1 is reversible, and thus has a product-form stationary distribution.

Given this product form distribution, the throughput-optimality of the algorithm
can be established using standard method. For example, we can set $P_i$ at time-slot $t$ to be $\frac{\exp(W_i[t])}{\exp(W_i[t]) + 1}$, and then prove the throughput-optimality by the aid of a time-scale separation assumption [45,47,48], or without such an assumption [49].

5.5 Numerical Simulation

In this section, we provide simulation results to (i) verify the effectiveness of using neighbor-expanded graph to capture the interference relationship in networks with cut-through capability; (ii) show the performance of the proposed distributed queue-length based CSMA algorithm.

Specifically, we study a 12-node ring network, where for each node in the network, we assume that it can communicate and interfere with only its two immediate neighbors, which makes the network graph $G$ and the interference graph $G_I$ to be the same, as depicted in Figure 5.6(a). There are four flows running in the network, with the source and destination node of each flow indicated in Figure 5.6(b). Since the network has full-duplex cut-through capability, it is not hard to see that we can divide the four flows into two groups \{flow 1, flow 3\} and \{flow 2, flow 4\}, where at each time-slot two cut-through routes can be formed to serve both flows in a single group without causing interference to each other. In other words, the network can simultaneously support the four flows each with a packet arrival rate of 0.5 packet/time-slot. In our simulation, we first obtain the neighbor-expanded graph of the 12-node ring network, and then apply the proposed distributed queue-length based CSMA algorithm. We choose $P_{\text{decision}}$ to be 0.2 and $P_i$ to be $\frac{\exp(0.2W_i[t])}{\exp(0.2W_i[t]) + 1}$. The packet arrivals to each flow follow a poisson process with the same rate $\lambda$. Packets in each node are routed towards the direction with the shortest path to its destination.

In Figure 5.7a we show the time-average of the sum-queue-length in the network (averaged across $10^5$ time-slots) as a function of the traffic rate $\lambda$. In Figure 5.7b
we focus on the case when $\lambda = 4.5$ packet/time-slot and plot the evolution of sum-queue-length in the network. From these figures we can see that, indeed, the proposed distributed CSMA algorithm can support any traffic rate $\lambda$ that is less than 0.5 packet/time-slot.

### 5.6 Conclusion

The full-duplex cut-through transmission technique in wireless networks introduces a direct coupling between the network-layer routing decision with the MAC-layer rate-region, leading to a complicated route/flow-dependent interference relationship between data-links in the network, which makes it hard to fully exploit the potential of cut-through transmission in an efficient way. In this chapter, we circumvent this difficulty by introducing the concept of neighbor-expanded network graph, which allows us to derive simple interference conditions that capture the full-duplex cut-through constraint in a scalable and low-complexity manner. The neighbor-expanded graph also enables us to devise algorithms that use only local information to form/change
(a) Average sum-queue-length vs. packet arrival rate $\lambda$.

(b) Evolution of sum-queue-length when $\lambda = 0.45$.

Figure 5.7: Simulation results

cut-through routes, with the proposed queue-length-based CSMA algorithm being an example.
CHAPTER 6
CONCLUSION

In this dissertation, we studied how the throughput in wireless systems is affected by the operations across different network layers, and showed how routing and scheduling algorithms could be redesigned to accommodate new wireless transmission techniques. Our research results enable a deeper understanding of how wireless systems could be designed to provide higher throughput.

First, we looked at the simple point-to-point wireless link and focused on the characterization of the distribution of the transmission delay, a metric that is closely related to the link-throughput. The transmission delay is determined jointly by the selection of physical-layer code-rates and link-layer retransmission schemes. We show that, in simple-ARQ scheme, without an appropriate level of coding redundancy, traditional retransmissions could incur very large (potentially heavy tail) delays that could substantially degrade the achievable throughput. However, when the receiver uses memory to cache all previously received undecodable packets, which is the case in the hybrid-ARQ scheme, a light-tailed packet length leads to a light-tailed delay, and the decay rate of the tail could be further improved by reducing the size of the packet-trunks that carry incremental information.

Next, we departed from the simple point-to-point channel and considered the use of rateless codes in broadcast channels, where the size of the network (the number of multi-cast receivers) can play an important role in shaping the link-throughput.
Using large deviation theory, we were able to derive a closed-form expression for
the broadcast throughput when the network size tends to infinity, for any mapping
between the network size and the coding block size. Then, by proving that for every
quadratic decrease in the number of receivers, we can have the coding block sized
halved without decreasing the achievable throughput, we related the non-asymptotic
throughput with the asymptotic throughput. Based on this result, we derived a lower
bound on the throughput for any finite coding block size and finite network size, which
is asymptotically tight.

Then, we shifted our focus from characterizing the link-layer throughput to the
characterization of the MAC-layer rate-region (also called throughput-region) in multi-
antenna multi-hop wireless networks. By assuming a simple binary interference
model, we were able to provide a thorough comparison of the rate-regions that can
be achieved by different combinations of multi-antenna/multi-RF-chain techniques
from a degree-of-freedom perspective. Given that each multi-antenna/multi-RF-chain
technique provides unique scheduling flexibility, our results answer the questions of
which combination of techniques could provide a larger throughput-region over an-
other, and under what condition can the gain in throughput-region be realized.

Finally, the MAC-layer scheduling component and the network-layer routing com-
ponent were considered. We showed that when the full-duplex capability meets
with the cross-interference cancellation capability, a powerful transmission scheme,
namely cut-through transmission, can be achieved. Then, we realized, from the
MAC/network layer control perspective, that cut-through transmission introduces
a special coupling between the MAC-layer and the network-layer decisions, which
brings challenges in the design of efficient scheduling and routing algorithms. By
introducing the concept of the neighbor-expanded graph, we were able to circum-
vent these challenges and enable a seamless adaptation of traditional half-duplex
routing/scheduling algorithms into wireless networks with full-duplex cut-through capabilities.

The investigations in this dissertation has demonstrated, as a first step, the potential of applying the Large Deviation Theory in the performance analysis of wireless systems. In many existing literatures that focus on the asymptotic performance analysis, often the Central Limit Theorem is invoked, which gives the approximate distribution of the scaled sum of a long sequence of identically distributed random variables. However, applying the Central Limit Theorem in the asymptotic analysis may lead to involved, or even non-closed-form results that are hard to evaluate, especially when the random variables of focus are correlated. For example, in [25], the authors consider the same problem as that in Section 3.3. By applying the Central Limit Theorem together with the Extreme Value Theory, they also obtain an expression of the asymptotic throughput, which, however, is not in a closed form. In contrast, in Chapter 3, by subtly combining the Extreme Value Theory with the Large Deviation Theory, we are able to find a closed-form expression for the asymptotic throughput under arbitrary channel correlation. Further, our large deviation analysis even leads to a lower bound on the non-asymptotic throughput, which performs significantly better than previous known ones. Our work in Chapter 2 is another evident on how Large Deviation Theory can help us gain understanding on the throughput/delay performance in wireless links. Since wireless systems have evolved to a stage where a large number of antennas and/or a large number of frequency channels can be utilized, we believe that our investigations in the first two chapters provide useful insights on how Large Deviation Theory could be used to characterize the system performance with increasing number of antennas and frequency channels.
In this dissertation, we compared the throughput of different multi-antenna techniques under the ideal binary interference model. To obtain more accurate throughput comparison, we need to take the aggregated interference into account. There are existing works that use stochastic geometry to characterize the achievable rate distribution in a multi-cell environment. It would be interesting to investigate how different multi-antenna techniques affect the distribution of the aggregated interference in the wireless system and in turn shape the distribution of achievable rates, by using the tool of stochastic geometry. From another perspective, besides providing gains in the network throughput-region, the scheduling flexibility brought by the multi-antenna techniques may also bring significant benefits in terms of reducing the packet queueing delay. An interesting direction is to compare the optimal packet evacuation time (the time for all existing packets to evacuate the system when there is no future arrival in the network) under different multi-antenna schemes. Moreover, one could extend our analyses with the two-antenna schemes to the case when each node has more than two antennas. It is of interest to understand how the network delay/throughput scales as the increase of the number of antennas.

Regarding the development of optimal resource control algorithms, there are also intriguing new directions. In this dissertation, we only considered developing optimal routing and scheduling algorithms for cut-through enabled network. Over the past few years, many other new physical-layer techniques have been proposed to combat inter-link interference and expand the network throughput-region, such as interference-alignment [50], network-MIMO, side-channel assisted full-duplex [51], to name a few. While the performance of these new schemes are well-studied in elementary scenarios, from general multi-hop networks’ perspective, these new techniques call for the redesign of network-wide resource allocation algorithms.
APPENDIX A: PROOFS FOR CHAPTER 2

A.1 Proof of Lemma 2.3.1

First we consider the case when $k \geq 1$. By Definition 2.2.1, we have

$$
P[N_{m}^{(1)} > n | L_c = l_c] = P \left[ \sum_{i=1}^{L_c} 1 \left( \sum_{j=1}^{n} X_{(j-1)L_c+i} \geq 1 \right) \leq \beta L_c | L_c = l_c \right]
$$

$$
= P \left[ \sum_{i=1}^{L_c} 1 \left( \sum_{j=1}^{n} X_{(j-1)L_c+i} \geq 1 \right) \leq \beta L_c | L_c = l_c \right]
$$

$$
= P \left[ \sum_{i=1}^{L_c} 1 \left( \sum_{j=1}^{n} X_{(j-1)L_c+i} = 0 \right) > (1 - \beta) L_c | L_c = l_c \right]
$$

$$
= E \left[ \left. \sum_{i=1}^{L_c} \prod_{j=1}^{n} 1 \left( X_{(j-1)L_c+i} = 0 \right) \right| L_c = l_c, \bigcup_{j=1}^{n} E_j \right], \quad (A.1.1)
$$

where $E_j \triangleq \{ X_{(j-1)L_c+1}, \ldots, X_{(j-1)L_c+k} \}, 1 \leq j \leq n$. Let $Y_{in} = [Y_i, Y_{L_c+i}, \ldots, Y_{(n-1)L_c+i}]$ and

$$
f_n (Y_{in}) = \prod_{j=1}^{n} f \left( Y_{(j-1)L_c+i} \right).
$$

If $L_c > k$, then given $\bigcup_{j=1}^{n} E_j$, $\{ Y_{in} \}_{k<i \leq L_c}$ forms a Markov chain with state space $\{0, 1\}^k$ and probability transition matrix $\Pi^\otimes n$. We further observe that if $L_c > k$,
we have the following relationship

\[
\left\{ \sum_{i=1+k}^{L_c} \prod_{j=1}^{n} 1(X_{(j-1)L_c+i} = 0) > (1 - \beta)L_c \right\} \\
\subseteq \left\{ \sum_{i=1}^{L_c} \prod_{j=1}^{n} 1(X_{(j-1)L_c+i} = 0) > (1 - \beta)L_c \right\} \\
\subseteq \left\{ \sum_{i=1+k}^{L_c} \prod_{j=1}^{n} 1(X_{(j-1)L_c+i} = 0) > (1 - \beta)L_c - k \right\}.
\]

Using the above observation, we can construct upper and lower bounds as follows.

\[
P\left[ \sum_{i=1}^{L_c} \prod_{j=1}^{n} 1(X_{(j-1)L_c+i} = 0) > (1 - \beta)L_c \left| L_c, \bigcup_{j=1}^{n} E_j \right. \right] \\
\geq \mathbb{P}\left[ \sum_{i=1+k}^{L_c} f_n(Y_{in}) > (1 - \beta)L_c \left| L_c, \bigcup_{j=1}^{n} E_j \right. \right], \\
P\left[ \sum_{i=1}^{L_c} \prod_{j=1}^{n} 1(X_{(j-1)L_c+i} = 0) > (1 - \beta)L_c \left| L_c, \bigcup_{j=1}^{n} E_j \right. \right] \\
\leq \mathbb{P}\left[ \sum_{i=1+k}^{L_c} f_n(Y_{in}) > (1 - \beta)L_c - k \left| L_c, \bigcup_{j=1}^{n} E_j \right. \right]. \tag{A.1.2}
\]

By a direct application of Theorem 3.1.2 in [20], we know that for a given \( \varepsilon > 0 \) and any values of \( \bigcup_{j=1}^{n} E_j \), we can find \( l_\varepsilon \) such that

\[
P\left[ \sum_{i=1+k}^{L_c} f_n(Y_{in}) > (1 - \beta)L_c \left| L_c = l_c \right. \right] \\
\geq e^{-\inf_{1-\omega>1-\beta} \Lambda_n(\omega, \Pi)(1+\varepsilon)l_c}, \tag{A.1.3}
\]

\[
P\left[ \sum_{i=1+k}^{L_c} f_n(Y_{in}) > (1 - \beta)L_c - k \left| L_c = l_c \right. \right] \\
\leq e^{-\inf_{1-\omega>1-\beta} \Lambda_n(\omega, \Pi)(1-\varepsilon)l_c}, \tag{A.1.4}
\]

111
whenever $l_c > l_e$. Since $\Lambda_n(\omega, \Pi)$ is a large deviation rate function, from [20] we know that

$$\inf_{1 - \omega > 1 - \beta} \Lambda_n(\omega, \Pi) = \begin{cases} 
\Lambda_n(\beta, \Pi) & \text{if } \mu_n < \beta \\
0 & \text{otherwise}
\end{cases}$$

$$= \Lambda_n(\beta, \Pi) \mathbb{1}(n \geq \alpha). \quad (A.1.5)$$

The upper and lower bounds in Equation (A.1.4) and (A.1.3), together with Equation (A.1.1), (A.1.2) and (A.1.5), imply that

$$- \lim_{l_c \to \infty} \inf \frac{\log \mathbb{P} \left[ N_m^{(1)} > n | L_e = l_c \right]}{l_c} \leq (1 + \varepsilon) \Lambda_n(\beta, \Pi) \mathbb{1}(n \geq \alpha),$$

$$- \lim_{l_c \to \infty} \sup \frac{\log \mathbb{P} \left[ N_m^{(1)} > n | L_e = l_c \right]}{l_c} \geq (1 - \varepsilon) \Lambda_n(\beta, \Pi) \mathbb{1}(n \geq \alpha),$$

which, with $\varepsilon \to 0$, completes the proof when $k \geq 1$.

Next, let us consider the case when $k = 0$.

In this memoryless channel case, for any single bit in the codeword, after the $n^{th}$ transmission, the probability that the bit is successfully received is $1 - (1 - \gamma)^n$. Therefore, for a certain packet, the probability of the number of transmissions being greater than $n$ is equal to the probability of the number of transmissions being greater than 1 under a memoryless erasure channel with erasure probability $(1 - \gamma)^n$.

Then, by a direct application of Gärtner-Ellis theorem (Theorem 2.3.6 in [20]),
we have, for any $n \geq 1$,
\[
\lim_{l_c \to \infty} \frac{\log \mathbb{P}[N_m^{(1)} > n|L_c = l_c]}{l_c} = -\Lambda_n(\beta, \Pi) 1(n \geq \alpha),
\]
where $\Lambda_n(\beta, \Pi)$ and $\alpha$ can be written in closed-forms, which are
\[
\Lambda_n(\beta, \Pi) = \sup_{\theta} \{\theta(1 - \beta) - \log (\mathbb{E} [X_i^\theta X_i])\}
= \sup_{\theta} \{\theta(1 - \beta) - \log (1 - (1 - \gamma)^n + (1 - \gamma)^n e^\theta)\}
= \beta \log \frac{\beta}{1 - (1 - \gamma)^n} + (1 - \beta) \log \frac{1 - \beta}{(1 - \gamma)^n},
\]
and $\alpha = \left\lceil \frac{\log(1 - \beta)}{\log(1 - \gamma)} \right\rceil$.

### A.2 Proof of Lemma 2.3.2

The first equation in Lemma 2.3.2 follows directly from the fact that
\[
\mathbb{P} [N_m^{(r)} > nr|L_c = l_c] = \mathbb{P} [N_m^{(1)} > n|L_c = l_c].
\]

Next, let us focus on the last two equations in Lemma 2.3.2. Denote
\[
Z_1(n, l_c, B) = \sum_{i=1}^B \sum_{l=(l_c/r)(i-1)+1}^{(l_c/r)i} 1 \left( \sum_{j=1}^{n+1} X_{(j-1)l_c+i} \geq 1 \right),
\]
\[
Z_2(n, l_c, B) = \sum_{i=B+1}^r \sum_{l=(l_c/r)(i-1)+1}^{(l_c/r)i} 1 \left( \sum_{j=1}^{n} X_{(j-1)l_c+i} \geq 1 \right).
\]
By Definition 2.2.1, it is easy to see that

\[ P[N^{(r)}_m > B| L_c = l_c] = P\left[ \frac{Z_1(0, l_c, B)}{l_c} < \beta \right], \]

and

\[ P[N^{(r)}_m > nr + B| L_c = l_c] = \sum_{z_1 \geq 0, z_2 \geq 0} P\left[ \frac{Z_1(n, l_c, B)}{l_c} = z_1, \frac{Z_2(n, l_c, B)}{l_c} = z_2 \right]. \]

Then based on the above two equations, the last two equations in Lemma 2.3.2 can be proven using the same technique as that used in the proof of Lemma 2.3.1.

### A.3 Proof of Lemma 2.4.1

From Definition 2.2.2 we know

\[ P[N_f > n| L_c] = E \prod_{j=1}^{n} P\left[ \sum_{i=(j-1)L_c+1}^{jL_c} X_i \leq \beta L_c \left| \bigcup_{i=1}^{n} \mathcal{E}_i, L_c \right. \right] \left| L_c \right. \right], \quad (A.3.1) \]
where $\mathcal{E}_j \triangleq \{X_{(j-1)L_c+1}, \ldots, X_{(j-1)L_c+k}\}$, $1 \leq j \leq n$. The last equation is due to the Markov property of the channel states. Observe that if $L_c > k$, for any $1 \leq j \leq n$,

$$
\left\{ \sum_{i=(j-1)L_c+1+k}^{jL_c} X_i \leq \beta L_c \right\} \subseteq \left\{ \sum_{i=(j-1)L_c+1}^{jL_c} X_i \leq \beta L_c \right\} \subseteq \left\{ \sum_{i=(j-1)L_c+1+k}^{jL_c} X_i \leq \beta L_c - k \right\},
$$

which further yields

$$
\Pr\left[ \sum_{i=(j-1)L_c+1+k}^{jL_c} X_i \leq \beta L_c - k \left| \bigcup_{i=1}^{n} \mathcal{E}_i, L_c \right. \right] \geq \Pr\left[ \sum_{i=(j-1)L_c+1}^{jL_c} X_i \leq \beta L_c \left| \bigcup_{i=1}^{n} \mathcal{E}_i, L_c \right. \right] \geq \Pr\left[ \sum_{i=(j-1)L_c+1+k}^{jL_c} X_i \leq \beta L_c \left| \bigcup_{i=1}^{n} \mathcal{E}_i, L_c \right. \right].
$$

Similarly as the proof of Lemma 2.3.1, by Theorem 3.1.2 in [20], we obtain, for any $1 \leq j \leq n$

$$
\lim_{L_c \to \infty} \frac{\log \Pr\left[ \sum_{i=(j-1)L_c+1}^{jL_c} X_i > \beta L_c \left| \bigcup_{i=1}^{n} \mathcal{E}_i, L_c = L_c \right. \right]}{L_c} = -\Lambda_1(\beta, \Pi) \mathbf{1}(\beta > \gamma),
$$

$$
\lim_{L_c \to \infty} \frac{\log \Pr\left[ \sum_{i=(j-1)L_c+1}^{jL_c} X_i \leq \beta L_c \left| \bigcup_{i=1}^{n} \mathcal{E}_i, L_c = L_c \right. \right]}{L_c} = -\Lambda_1(\beta, \Pi) \mathbf{1}(\beta < \gamma),
$$

115
which, by combining Equation (A.3.1), completes the proof.

### A.4 Proof of Lemma A.4.1

In order to prove the theorems, we need the following lemma.

**Lemma A.4.1.** Assume $b$ is a function of $t$, which satisfies $b \triangleq b(t) > \frac{t}{y}$. Then, for any $x, y \in \mathbb{N}$ we have

$$
\lim_{t \to \infty} \frac{\log \mathbb{P}[N_m^{(1)} > x, \frac{t}{y+1} < L_c(b) < \frac{t}{y}]}{t} = -\frac{\lambda + \Lambda_x(\beta, \Pi) 1(x \geq \alpha)}{y+1}.
$$

**Proof.** Observe that

$$
\mathbb{P} \left[ N_m^{(1)} > x, \frac{t}{y+1} < L_c(b) < \frac{t}{y} \right] = \sum_{l_c = \lceil t/(y+1) \rceil}^{\lfloor t/y \rfloor} \mathbb{P} \left[ N_m^{(1)} > x \middle| L_c(b) = l_c \right] \mathbb{P} \left[ L_c(b) = l_c \right]. \tag{A.4.1}
$$

Let us first consider the case when $x \geq \alpha$. From Lemma 2.3.1 we know that when $x \geq \alpha$, for $l_c$ large enough, $\mathbb{P}[N_m^{(1)} > x | L_c(b) = l_c]$ is a decreasing function of $l_c$. Therefore, for $t$ large enough,

$$
\mathbb{P} \left[ N_m^{(1)} > x, \frac{t}{y+1} < L_c(b) < \frac{t}{y} \right] \leq \mathbb{P} \left[ N_m^{(1)} > x \middle| L_c(b) = \left\lfloor \frac{t}{y+1} \right\rfloor \right] \mathbb{P} \left[ \frac{t}{y+1} < L_c(b) < \frac{t}{y} \right]. \tag{A.4.2}
$$
Since $b = b(t) > \frac{t}{y}$, by the definition of $L_c(b)$, we can easily obtain

$$
\lim_{t \to \infty} \log \frac{\mathbb{P}\left[ \frac{t}{y+1} < L_c(b) < \frac{t}{y} \right]}{t} = -\frac{\lambda}{y + 1}.
$$

(A.4.3)

Combining Equation (A.4.3) and (A.4.2), and by Lemma 2.3.1, we get, for $x \geq \alpha$

$$
\limsup_{t \to \infty} \frac{\log \mathbb{P}\left[ N_{m}^{(1)} > x, \frac{t}{y+1} < L_c(b) < \frac{t}{y} \right]}{t} \leq - (\lambda + \Lambda_x(\beta, \Pi))/(y + 1).
$$

The lower bound can be constructed in a similar manner by noting that

$$
\mathbb{P}\left[ N_{m}^{(1)} > x, \frac{t}{y+1} < L_c(b) < \frac{t}{y} \right] \geq \mathbb{P}\left[ N_{m}^{(1)} > x \left| L_c(b) = \left\lfloor \frac{t}{y+1} \right\rfloor \right. \right] \mathbb{P}\left[ L_c(b) = \left\lfloor \frac{t}{y+1} \right\rfloor \right].
$$

Next, let us consider the case when $x < \alpha$. From Lemma 2.3.1 we know that for any $\varepsilon > 0$, we can find a large enough $l_\varepsilon$ such that for all $l_c > l_\varepsilon$,

$$
1 - \varepsilon \leq \mathbb{P}\left[ N_{m}^{(1)} > x \left| L_c(b) = l_c \right. \right] \leq 1,
$$

which, by combining Equation (A.4.1) and (A.4.3), yields, for $x < \alpha$,

$$
\lim_{t \to \infty} \log \frac{\mathbb{P}\left[ N_{m}^{(1)} > x, \frac{t}{y+1} < L_c(b) < \frac{t}{y} \right]}{t} = -\lambda/(y + 1).
$$

□
A.5 Proof of Theorem 2.3.1

Observe that

\[
\mathbb{P}\left[ T_m^{(r)} > t \right] = \sum_{h=r}^{\infty} \mathbb{P}\left[ T_m^{(r)} > t, \frac{tr}{h+1} < L_c \leq \frac{tr}{h} \right] + \mathbb{P}\left[ T_m^{(r)} > t, L_c > t \right]
\]

\[
= \sum_{n=1}^{\infty} \sum_{h=nr}^{(n+1)r-1} \mathbb{P}\left[ N_m^{(r)} > h, \frac{tr}{h+1} < L_c \leq \frac{tr}{h} \right]
\]

\[
+ \mathbb{P}\left[ T_m^{(r)} > t, L_c > t \right]. \quad (A.5.1)
\]

Let us first focus on the first part of Equation (A.5.1). Denote

\[
\mathbb{P}_{ntr} = \sum_{h=nr}^{(n+1)r-1} \mathbb{P}\left[ N_m^{(r)} > h, \frac{tr}{h+1} < L_c \leq \frac{tr}{h} \right],
\]

then it is easy to check that

\[
\mathbb{P}_{ntr} \leq \mathbb{P}\left[ N_m^{(1)} > n, \frac{t}{n+1} < L_c \leq \frac{t}{n} \right],
\]

\[
\mathbb{P}_{ntr} \geq \mathbb{P}\left[ N_m^{(1)} > n+1, \frac{t}{n+1} < L_c \leq \frac{t}{n} \right],
\]

which, by Lemma A.4.1 in Appendix A.4, yields

\[
\limsup_{t \to \infty} \frac{\log \mathbb{P}_{ntr}}{t} \leq -\frac{\lambda + \Lambda_n(\beta, \Pi)1(n \geq \alpha)}{n+1},
\]

\[
\liminf_{t \to \infty} \frac{\log \mathbb{P}_{ntr}}{t} \geq -\frac{\lambda + \Lambda_{n+1}(\beta, \Pi)1(n \geq \alpha - 1)}{n+1}. \quad (A.5.2)
\]
Now let us focus on the second part of Equation (A.5.1). Observe that

\[
\mathbb{P} \left[ T^{(r)}_m > t, L_c > t \right] \\
= \sum_{n=1}^{r-1} \mathbb{P} \left[ T^{(r)}_m > t, \frac{rt}{n+1} < L_c \leq \frac{rt}{n} \right] + \mathbb{P} \left[ T^{(r)}_m > t, rt < L_c \right] \\
= \sum_{n=1}^{r-1} \mathbb{P} \left[ N^{(r)}_m > n, \frac{rt}{n+1} < L_c \leq \frac{rt}{n} \right] \\
+ \mathbb{P} \left[ N^{(r)}_m > 0, rt < L_c \right].
\]

From Lemma 2.3.2 we know that, for \( 1 \leq n \leq r - 1 \),

\[
- \lim_{l_c \to \infty} \frac{\log \mathbb{P} \left[ N^{(r)}_m > n | L_c = l_c \right]}{l_c} \\
= \Lambda_1 \left( \frac{\beta r}{n}, \Pi \right) 1 \left( \frac{\beta r}{n} < \gamma \right).
\]

By combining the above two equations, we get

\[
- \lim_{t \to \infty} \frac{\log \mathbb{P} \left[ T^{(r)}_m > t, L_c > t \right]}{t} \\
= - \lim_{t \to \infty} \frac{\log \mathbb{P} [L_c > t]}{t} = \lambda,
\]

when \( \beta > \gamma \), and

\[
- \lim_{t \to \infty} \frac{\log \mathbb{P} \left[ T^{(r)}_m > t, L_c > t \right]}{t} \\
= \min \left\{ - \lim_{t \to \infty} \frac{\log \mathbb{P} [L_c > tr / \left\lceil \frac{r \beta}{\gamma} \right\rceil]}{t}, \\
- \lim_{t \to \infty} \frac{\log \left( \sum_{n=1}^{r-1} \mathbb{P} \left[ N^{(r)}_m > n, \frac{rt}{n+1} < L_c \leq \frac{rt}{n} \right] \right)}{t} \right\} \\
= \min \left\{ \frac{\lambda r}{\left\lceil \frac{r \beta}{\gamma} \right\rceil}, \tilde{\Lambda}_3 \right\} = \Lambda_3^g,
\]

(A.5.4)
when $\beta < \gamma$.

Combining Equation (A.5.1), (A.5.2), (A.5.3), and (A.5.4), we get

$$\limsup_{t \to \infty} \frac{\log \mathbb{P}\left[T_m^{(r)} > t\right]}{t} = \max \left\{ \limsup_{t \to \infty} \frac{\log \sum_{n=1}^{\infty} \mathbb{P}_{ntr}}{t}, \right. \\
\left. \lim_{t \to \infty} \frac{\log \mathbb{P}\left[T_m^{(r)} > t, L_c > t\right]}{t} \right\}^{(a)} \leq \max \left\{ -\inf_n \frac{\lambda + \Lambda_n(\beta, \Pi) \mathbf{1}(n \geq \alpha)}{n+1}, -\Lambda_3^o \right\} \\
= -\min\{\Lambda_1^o, \Lambda_3^o\}. \quad (A.5.5)$$

The lower bound can be found in a similar manner. Notice that inequality (a) in the preceding equation is true because $\mathbb{P}_{ntr}$ is nonzero only for a finite number of $n$, due to the fact that $L_c$ cannot be less than 1.

In the special case when $r = 1$, by Lemma A.4.1 and the definition of $L_c$, we have

$$\lim_{t \to \infty} \frac{\log \mathbb{P}_{ntr}}{t} = -\frac{\lambda + \Lambda_n(\beta, \Pi) \mathbf{1}(n \geq \alpha)}{n+1},$$

$$\lim_{t \to \infty} \frac{\log \mathbb{P}\left[T_m^{(1)} > t, L_c > t\right]}{t} = -\lambda, \quad (A.5.6)$$

which, by combining Equation (A.5.1), completes the proof.
A.6 Proof of Theorem 2.3.2

From the definition of \( n_2^o, \Lambda_1^o \) and \( \Lambda_2^o \) in Notation 2.2.5 and by Lemma A.4.1, we can obtain, for any \( \delta > 0 \),

\[
\limsup_{t \to \infty} -\frac{\log \mathbb{P}[T_m^{(r)} > t, \frac{t}{n_2^o + 1} < L_c \left( \frac{t}{n_2^o} + \delta \right) \leq \frac{t}{n_2^o}]}{t} \leq \Lambda_2^o,
\]

\[
\liminf_{t \to \infty} -\frac{\log \mathbb{P}[T_m^{(r)} > t, \frac{t}{n_2^o + 1} < L_c \left( \frac{t}{n_2^o} + \delta \right) \leq \frac{t}{n_2^o}]}{t} \geq \Lambda_1^o.
\]

Then, for any \( \eta > 0 \) and for any \( \delta > 0 \), we can find \( t(\eta) \) such that

\[
-\frac{\log \mathbb{P}[T_m^{(r)} > t, \frac{t}{n_2^o + 1} < L_c \left( \frac{t}{n_2^o} + \delta \right) \leq \frac{t}{n_2^o}]}{t} \leq (1 + \eta)\Lambda_2^o,
\]

\[
-\frac{\log \mathbb{P}[T_m^{(r)} > t, \frac{t}{n_2^o + 1} < L_c \left( \frac{t}{n_2^o} + \delta \right) \leq \frac{t}{n_2^o}]}{t} \geq (1 - \eta)\Lambda_1^o,
\]

whenever \( t > t(\eta) \). We denote \( b(\eta) \triangleq \frac{t(\eta)}{n_2^o} + \delta \). In other words, for any \( b > b(\eta) \), whenever \( t \in [(b - \delta)n_2^o, bn_2^o] \),

\[
-\frac{\log \mathbb{P}[T_m^{(r)} > t, \frac{t}{n_2^o + 1} < L_c(b) \leq \frac{t}{n_2^o}]}{t} \leq (1 + \eta)\Lambda_2^o,
\]

\[
-\frac{\log \mathbb{P}[T_m^{(r)} > t, \frac{t}{n_2^o + 1} < L_c(b) \leq \frac{t}{n_2^o}]}{t} \geq (1 - \eta)\Lambda_1^o,
\]

which, by using the same technique as in Equation (A.5.5), completes the proof of the first part. The second part of Theorem 2.3.2 follows by noting that

\[
\lim_{t \to \infty} -\frac{\log \mathbb{P}[T_m^{(1)} > t, \frac{t}{n_1^o + 1} < L_c \left( \frac{t}{n_1^o} + \delta \right) \leq \frac{t}{n_1^o}]}{t} = \Lambda_1^o,
\]

where the definition of \( n_1^o \) can be found in Notation 2.2.5.
A.7 Proof of Theorem 2.4.1

1) If $\beta > \gamma$, by Lemma 2.4.1, for any $\varepsilon > 0$, we can find $l_\varepsilon$ such that

$$\mathbb{P}[N_f > n|L_c = l_c] \geq (1 - e^{-L_1(\beta, \Pi)(1-\varepsilon)})^n,$$

whenever $l_c > l_\varepsilon$. Then we have, for $n$ large enough,

$$\mathbb{P}[N_f > n] = \mathbb{E}[\mathbb{P}[N_f > n|L_c]]$$

$$\geq \mathbb{E}\left[\frac{\log n}{\Lambda_1(\beta, \Pi)(1-\varepsilon)} < L_c < \frac{\log n}{\Lambda_1(\beta, \Pi)(1-\varepsilon)} + z, \right.$$

$$\left. (1 - e^{-\Lambda_1(\beta, \Pi)(1-\varepsilon)L_c})^n \right]$$

$$\geq \mathbb{E}\left[\frac{\log n}{\Lambda_1(\beta, \Pi)(1-\varepsilon)} < L_c < \frac{\log n}{\Lambda_1(\beta, \Pi)(1-\varepsilon)} + z, \right.$$

$$\left. (1 - e^{-\Lambda_1(\beta, \Pi)(1-\varepsilon)L_c})^n \right]$$

$$\geq e^{-\lambda(1+\varepsilon)\frac{\log n}{\Lambda_1(\beta, \Pi)(1-\varepsilon)}} (1 - e^{-(\log n)})^n.$$ 

Taking logarithms on both sides of the preceding inequality, we get

$$\liminf_{n \to \infty} \frac{\log \mathbb{P}[N_f > n]}{\log n} \geq -\frac{\lambda(1+\varepsilon)}{\Lambda_1(\beta, \Pi)(1-\varepsilon)},$$

which, when $\varepsilon \to 0$, results in the lower bound.

Next, we prove the upper bound. Using the same technique as in the proof of the lower bound, and by the definition of $L_c$, we can find $l_\varepsilon$ such that
\[ \Pr[N_f > n] \leq \Pr[L_c > l_c, (1 - e^{-\Lambda_1(\beta, \Pi)(1+\epsilon)L_c})^n] \]
\[ + \Pr[N_f > n, L_c \leq l_c] \]
\[ \leq \sum_{l=l_c}^{\infty} (1 - e^{-\Lambda_1(\beta, \Pi)(1+\epsilon)t})^n \Pr[L_c = l] + O(e^{-\xi n}) \]
\[ \leq O\left(\int_{0}^{\infty} (1 - e^{-\Lambda_1(\beta, \Pi)(1+\epsilon)x})^n e^{-\lambda(1-\epsilon)x} \, dx\right) \]
\[ + O(e^{-\xi n}), \]
for some \( \xi > 0 \). Computing the integral in the preceding inequality, we obtain
\[ \limsup_{n \to \infty} \frac{\log \Pr[N_f > n]}{\log n} \leq -\frac{\lambda(1 - \epsilon)}{\Lambda_1(\beta, \Pi)(1 + \epsilon)}, \]
which, with \( \epsilon \to 0 \), proves the upper bound.

Now, we prove the result for \( \Pr[T_f > t] \). The upper bound follows by noting that, for any \( h > 0 \),
\[ \Pr[T_f > t] \leq \Pr[N_f L_c > t, L_c \leq h \log t] + \Pr[L_c > h \log t] \]
\[ \leq \Pr[N_f > t/(h \log t)] + \Pr[L_c > h \log t], \]
where \( \lim_{t \to \infty} \log \Pr[N_f > t/(h \log t)]/ \log t = -\lambda/\Lambda_1(\beta, \Pi) \), and \( \Pr[L_c > h \log t] = o(\Pr[N_f > t/(h \log t)]) \) for \( h \) large enough.

The lower bound follows by noting that, for some \( l_2 > l_1 > 0 \) with \( \Pr[l_1 < L_c < l_2] \)
$l_2 > 0,$

$$\mathbb{P}[T_f > t] \geq \mathbb{P}[N_f L_c > t, l_1 < L_c < l_2]$$

$$\geq \mathbb{P}[N_f > t/l_1] \mathbb{P}[l_1 < L_c < l_2].$$

2) Observe that

$$\mathbb{P}[T_f > t]$$

$$= \mathbb{P}[L_c > t] + \sum_{n=1}^\infty \mathbb{P} \left[ T_f > t, \frac{t}{n+1} < L_c \leq \frac{t}{n} \right]$$

$$= \mathbb{P}[L_c > t] + \sum_{n=1}^\infty \mathbb{P} \left[ N_f > n, \frac{t}{n+1} < L_c \leq \frac{t}{n} \right]. \quad (A.7.1)$$

By Lemma 2.4.1 we know that $\mathbb{P} [N_f > n| L_c = l_c]$ has a rate function of $n\Lambda_1(\beta, \Pi)$ with respect to the increase of $l_c$ when $\beta > \gamma$. By using the same technique as that used in the proof of Lemma A.4.1, we can easily get, when $\beta < \gamma,$

$$\lim_{t \to \infty} \frac{\log \mathbb{P} [N_f > n, \frac{t}{n+1} < L_c \leq \frac{t}{n}]}{t} = -\frac{n\Lambda_1(\beta, \Pi) + \lambda}{n+1},$$

$$\lim_{t \to \infty} \frac{\log \mathbb{P} [L_c > t]}{t} = -\lambda,$$

which, by combining Equation (A.7.1) and using the same technique as in Equation (A.5.5), yields

$$\lim_{t \to \infty} \frac{\log \mathbb{P}[T_f > t]}{t} = -\min \left\{ \inf_{n \in \mathbb{N}} \left\{ \frac{n\Lambda_1(\beta, \Pi) + \lambda}{n+1} \right\}, \lambda \right\}$$

$$= -\min\{\Lambda_1(\beta, \Pi), \lambda\}.$$
A.8 Proof of Theorem 2.4.2

1) By Lemma 2.4.1 we know that when $\beta > \gamma$

$$
\mathbb{P}[N_f > n | L_c = l_c] = \left(1 - e^{-l_c \Lambda_1(\beta, \Pi)(1+g(l_c))}\right)^n,
$$

with $g(l_c) \in o(1)$ as $l_c \to \infty$. Let us denote $l_n$ as the root of the function $l_n(1 + g(l_n)) = \log(1 + \frac{1}{n})$. In other words,

$$
l_n(1 + g(l_n)) = \frac{\log n}{\Lambda_1(\beta, \Pi)}.
$$

For any $b_0 > 0$, we have

$$
\mathbb{P}[N_f > n, l_n - z < L_c(l_n + b_0) \leq l_n] \geq \mathbb{P}[N_f > n | L_c = l_n] \mathbb{P}[l_n - z < L_c(l_n + b_0) \leq l_n]. \quad (A.8.1)
$$

Note that, by Lemma 2.4.1,

$$
\lim_{n \to \infty} \frac{\log \mathbb{P}[N_f > n | L_c = l_n]}{\log n} = \lim_{n \to \infty} \frac{\log \left(1 - e^{-l_n \Lambda_1(\beta, \Pi)(1+g(l_c))}\right)^n}{\log n} = \lim_{n \to \infty} \frac{n \log(1 + \frac{1}{n})}{\log n} = 0. \quad (A.8.2)
$$
Also, by the definition of $L_c(b)$,

\[
\lim_{n \to \infty} \frac{\log \mathbb{P}[l_n - z < L_c(l_n + b_0) \leq l_n]}{\log n} = \lim_{n \to \infty} \frac{\log \mathbb{P}[l_n - z < L_c(l_n + b_0) \leq l_n]}{l_n} \frac{l_n}{\log n} = -\lambda \frac{1}{\Lambda_1(\beta, \Pi)}. \tag{A.8.3}
\]

Combining Equation (A.8.1), (A.8.2) and (A.8.3), we get

\[
\lim_{n \to \infty} \frac{\log \mathbb{P}[N_f > n, l_n - z < L_c(l_n + b_0) \leq l_n]}{\log n} = -\frac{\lambda}{\Lambda_1(\beta, \Pi)}.
\]

Therefore, for any $\eta > 0$, we can find a $n_1(\eta)$ such that

\[
\lim_{n \to \infty} \frac{\log \mathbb{P}[N_f > n, l_n - z < L_c(l_n + b_0) \leq l_n]}{\log n} \geq -\frac{\lambda}{\Lambda_1(\beta, \Pi)}(1 + \eta), \tag{A.8.4}
\]

whenever $n > n_1(\eta)$. Also, by Theorem 2.4.1, we can find $n_2(\eta)$ such that

\[
\lim_{n \to \infty} \frac{\log \mathbb{P}[N_f > n]}{\log n} \leq -\frac{\lambda}{\Lambda_1(\beta, \Pi)}(1 - \eta). \tag{A.8.5}
\]

Let $n(\eta) \triangleq \max\{n_1(\alpha), n_2(\alpha)\}$ and $b(\eta) \triangleq l_n(\eta) + b_0$. By combining Equation (A.8.4) and (A.8.5), we know that for any $\eta > 0$, we can find $b(\eta)$ such that for any $b > b(\eta)$,

\[
\limsup_{n \to \infty} \frac{\log \mathbb{P}[N_f(b) > n]}{\log n} \leq \lim_{n \to \infty} \frac{\log \mathbb{P}[N_f > n]}{\log n} \leq -\frac{\lambda}{\Lambda_1(\beta, \Pi)}(1 - \eta),
\]

126
and

\[
\liminf_{n \to \infty} \frac{\log \mathbb{P} [N_f(b) > n]}{\log n} \\
\geq \frac{\log \mathbb{P} [N_f > n, l_n - z < L_c(l_n + b_0) \leq l_n]}{\log n} \\
\geq - \frac{\lambda}{\Lambda_1(\beta, \Pi)} (1 + \eta),
\]

whenever \( n \in [n(\eta), n_b] \), where \( n_b \) satisfies \( b(1 + g(b)) = \frac{\log n_b}{\Lambda_1(\beta, \Pi)} \). From Lemma 2.4.1, we know that

\[
n_b = e^{\Lambda_1(\beta, \Pi)b(1+g(b))} = \left( \mathbb{P} [N_f = 1 | L_c = b] \right)^{-1}. \tag{A.8.6}
\]

2) Note that

\[
\lim_{n \to \infty} \frac{\log \mathbb{P} [N_f(b) > n]}{n} \\
= \max_{l_c} \left\{ \lim_{n \to \infty} \frac{\log \mathbb{P} [L_c(b) = l_c, N_f(b) > n]}{n} \right\} \\
= \max_{l_c} \left\{ \lim_{n \to \infty} \frac{\log \mathbb{P} [L_c(b) = l_c]}{n}, \log \left( \mathbb{P} [N_f(b) > 1 | L_c(b) = b] \right) \right\} \\
= \log \left( \mathbb{P} [N_f(b) > 1 | L_c(b) = b] \right).
\]

The characterization of \( \mathbb{P} [T_f(b) > t] \) follows by noting that \( T_f(b) = N_f(b)L_c(b) \).
A.9 Proof of Theorem 2.4.3

Observe that

\[
\Delta(b) = \lim_{n \to \infty} \frac{\sum_{i=1}^{n} \beta L_i}{n} \frac{n}{\sum_{i=1}^{n} T_i} = \beta \frac{E[L_i]}{E[T_i]}.
\]

From Theorem 2.4.2 we know that for a given \( \eta > 0 \), we can find \( n(\eta) \) and \( b \) large enough such that

\[
E[T_i] \geq \int_{n(\eta)b}^{mb} P[T_f(b) > t]dt \\
\geq \int_{n(\eta)b}^{mb} t^{-\frac{\lambda}{\alpha(1+\eta)(1+\eta)}} dt,
\]

which, by combing the definition of \( n_b \) in Equation (A.8.6), yields

\[-\limsup_{b \to \infty} \frac{\log \Delta(b)}{b} = \liminf_{b \to \infty} \frac{\log E[T_i]}{b} \geq \Lambda_1(\beta, \Pi) - \lambda.\]
APPENDIX B: PROOF OF CHAPTER 3

B.1 Proof of Theorem 3.3.1

In order to prove Theorem 3.3.1, we first need the following lemmas: Lemma B.1.1, Lemma B.1.2 and Lemma B.1.3, whose proofs can be found in Section B.2, B.3 and B.4.

**Lemma B.1.1.** For any \( \beta \in (0, 1) \) and any values of \( E \), we have

\[
\mathbb{P} \left[ T(n, K) > \frac{k}{\beta} \, K = k, \mathcal{E} \right]
= 1 - \left( 1 - e^{-\frac{k}{\beta} A(\beta, \Pi) 1(\beta < \gamma) + g(\beta, k, E)} \right)^n,
\]

where

\[
g(\beta, k, E) \in \begin{cases}
  o(k) \text{ as } k \to \infty & \text{if } \beta < \gamma \\
  o(1) \text{ as } k \to \infty & \text{if } \beta > \gamma
\end{cases},
\]

and \( 1(\beta < \gamma) \) is an indicator function which equals 1 when \( \beta < \gamma \) and 0 otherwise.

**Lemma B.1.2.** Assume \( k \) is a function of \( n \) and denote \( k := k(n) \), and define

\[
f(k, \beta, \mathcal{E}) := e^{\frac{k}{\beta} A(\beta, \Pi) 1(\beta < \gamma) - g(\beta, k, E)},
\]

then we have
1. For a fixed $\beta \in (0, 1)$, if $\lim_{n \to \infty} \frac{n}{f(k(n), \beta, \mathcal{E})} = 0$, then

$$\lim_{n \to \infty} \mathbb{P} \left[ T(n, K) > \frac{k(n)}{\beta} \left| K = k(n), \mathcal{E} \right. \right] = 0. \quad \text{(B.1.1)}$$

2. For a fixed $\beta \in (0, 1)$, if $\lim_{n \to \infty} \frac{n}{f(k(n), \beta, \mathcal{E})} = \infty$, then

$$\lim_{n \to \infty} \mathbb{P} \left[ T(n, K) > \frac{k(n)}{\beta} \left| K = k(n), \mathcal{E} \right. \right] = 1. \quad \text{(B.1.2)}$$

**Lemma B.1.3.** Let $\{h_n(x)\}$ be a set of Lebesgue measurable functions defined on $[0, \infty)$ and $h_n(x)$ converges to $1(x < y)$ almost everywhere for some $y > 0$. If $h_n(x)$ is a decreasing function of $x$ and have the range $[0, 1]$ for any $n \in \mathbb{N}$, then $h_n(x)$ converges globally in measure to $1(x < y)$.

Since $K$ is assumed to be a function of $n$, we denote this function as $k(n)$. According to definition 3.2.4 we have,

$$\lim_{n \to \infty} (\eta(n, K))^{-1} = \lim_{n \to \infty} \frac{\mathbb{E} \left[ T(n, K) | \mathcal{E} \right]}{K}. \quad \text{(B.1.3)}$$

Next, we obtain the value of $\lim_{n \to \infty} \mathbb{E}[T(n, K) | \mathcal{E}] / K$ and show that it is independent of $\mathcal{E}$. Note that

$$\lim_{n \to \infty} \frac{\mathbb{E} \left[ T(n, K) | \mathcal{E} \right]}{K} = \lim_{n \to \infty} \int_{0}^{\infty} \mathbb{P} \left[ T(n, K) > \frac{s}{k(n)} \left| K = k(n), \mathcal{E} \right. \right] ds$$

$$= \lim_{n \to \infty} \int_{0}^{\infty} \mathbb{P} \left[ T(n, K) > \frac{k(n)u}{k(n)} \left| K = k(n), \mathcal{E} \right. \right] du. \quad \text{(B.1.4)}$$

130
According to the assumption that \( \lim_{n \to \infty} k(n)/\log(n) = c \), we have

\[
\lim_{n \to \infty} \frac{n}{e^{k(n)/\log(n)} - \Lambda(\beta, \Pi)1(\beta < \gamma) - g(\beta, k, E)} = \lim_{n \to \infty} e^{\log n - \frac{k(n)}{\log(n)} - \Lambda(\beta, \Pi)1(\beta < \gamma) - g(\beta, k, E)} = \left\{ \begin{array}{ll} 0 & c > \beta \\ \frac{\beta}{\Lambda(\beta, \Pi)1(\beta < \gamma)} & c < \beta \\ \infty & c < \beta \end{array} \right.,
\]

where the last equation follows from the fact that \( g(\beta, k, E) \in o(k) \) or \( o(1) \).

Since \( \Lambda(\gamma, \Pi) \) is a finite non-negative decreasing function in \([0, \gamma]\) that attains zero only at \( \beta = \gamma \), we know that \( \frac{\beta}{\Lambda(\beta, \Pi)1(\beta < \gamma)} |_{\beta=0} = 0 \), \( \lim_{\beta \to \gamma^-} \frac{\beta}{\Lambda(\beta, \Pi)1(\beta < \gamma)} = \infty \) and \( \frac{\beta}{\Lambda(\beta, \Pi)1(\beta < \gamma)} \) is a monotone increasing function on the domain \((0, \gamma)\). Therefore, the equation \( c = \frac{\beta}{\Lambda(\beta, \Pi)1(\beta < \gamma)} \) has only one solution of \( \beta \), which we denote as

\[
\beta_c = \sup \left\{ \beta \mid c \geq \frac{\beta}{\Lambda(\beta, \Pi)}, 0 \leq \beta < \gamma \right\}.
\]
Then, by Lemma B.1.2, we get

\[
\lim_{n \to \infty} \mathbb{P}[T(n, K) > k(n)u | K = k(n), \mathcal{E}]
\]

\[
= \begin{cases} 
1 & \text{if } c < \frac{1}{u} 
\frac{1}{u} \Lambda(1/u, \Pi) \mathbf{1}(1/u < \gamma) 
\frac{1}{u} \Lambda(1/u, \Pi) \mathbf{1}(1/u < \gamma) 
\end{cases}
\]

\[
= \begin{cases} 
1 & \text{if } \beta_c > \frac{1}{u} 
\frac{1}{u} \Lambda(1/u, \Pi) \mathbf{1}(1/u < \gamma) 
\frac{1}{u} \Lambda(1/u, \Pi) \mathbf{1}(1/u < \gamma) 
\end{cases}
\]

\[
= \begin{cases} 
1 & \text{if } u < \frac{1}{\beta_c} 
0 & \text{if } u > \frac{1}{\beta_c}.
\end{cases}
\]  \hspace{1cm} (B.1.5)

We let \( h_n(u) \triangleq \mathbb{P}[T(n, K) > k(n)u | K = k(n), \mathcal{E}] \). Equation (B.1.5) implies that \( h_n(u) \) converges to \( 1(u < 1/\beta_c) \) pointwisely, where \( 1(.) \) is the indicator function. Since \( h_n(u) \) is a decreasing function of \( u \) and has the range \([0, 1]\) for all \( n \), by Lemma B.1.3 we know that \( h_n(u) \) globally converges in measure to \( 1(u < 1/\beta_c) \). We also know that the set of function \( \{h_n(u)\} \) is uniformly bounded. Then we can apply Vitali convergence theorem to Equation (B.1.4) to exchange the limit and integral and obtain

\[
\lim_{n \to \infty} \mathbb{E} \left[ \frac{T(n, K)}{K} | \mathcal{E} \right]
= \int_0^\infty \lim_{n \to \infty} \mathbb{P}[T(n, K) > k(n)u | K = k(n), \mathcal{E}] \, du = \frac{1}{\beta_c}.
\]  \hspace{1cm} (B.1.6)

Note that the above result is independent of the choice of the initial state \( \mathcal{E} \). Since the cardinality of the state space of \( \mathcal{E} \) is finite for a finite value of \( n \), we can exchange the limit and expectation in Equation (B.1.3), which, after combining with the above equation, completes the proof.
B.2 Proof of Lemma B.1.1

From definition (3.2.1) and (3.2.2), we have, for any \( t \),

\[
\{T(n, K) \leq t, \mathcal{E}\} = \bigcap_{i=1}^{n} \{T_i(K) \leq t, \mathcal{E}_i\}.
\]

Therefore, we have

\[
\mathbb{P}[T(n, K) > t|K = k, \mathcal{E}] = 1 - \mathbb{P}[T(n, K) \leq t|K = k, \mathcal{E}]
\]

\[
= 1 - \prod_{i=1}^{n} (1 - \mathbb{P}[T_i(K) > t|K = k, \mathcal{E}_i]). \tag{B.2.1}
\]

Let \( t = \frac{k}{\beta} \), from definition 3.2.1 we can get, for any \( 1 \leq i \leq n \),

\[
\mathbb{P}\left[T_i(K) > \frac{k}{\beta} \bigg| K = k, \mathcal{E}_i\right] = \mathbb{P}\left[\sum_{j=1}^{k/\beta} X_{ij} \leq k \bigg| \mathcal{E}_i\right]
\]

\[
= \mathbb{P}\left[\frac{\sum_{j=1}^{k/\beta} X_{ij}}{k/\beta} \leq \beta \bigg| \mathcal{E}_i\right],
\]

which, by combining with Equation (3.3.2) and (3.3.3), yields

\[
\lim_{k \to \infty} \frac{\log \mathbb{P}\left[T_i(K) > \frac{k}{\beta} \bigg| K = k, \mathcal{E}_i\right]}{k/\beta} = -\inf_{\lambda \in (-\infty, \beta]} \Lambda(\lambda, \Pi).
\]

\[
= -\Lambda(\beta, \Pi) \mathbf{1}(\beta < \gamma).
\]

133
Note that the above equation is equivalent of

\[
\mathbb{P} \left[ T_i(K) > \frac{k}{\beta} \mid K = k, \mathcal{E}_i \right] = e^{-\Lambda(\beta, \Pi) 1(\beta < \gamma) \frac{k}{\beta} + g_i(\beta, k, \mathcal{E}_i)},
\]

for some

\[
g_i(\beta, k, \mathcal{E}_i) \in \begin{cases} 
    o(k) & \text{as } k \to \infty \text{ if } \beta < \gamma \\
    o(1) & \text{as } k \to \infty \text{ if } \beta > \gamma 
\end{cases}.
\]

Then Equation (B.2.1) can be rewritten as

\[
\mathbb{P} \left[ T(n, K) > \frac{k}{\beta} \mid K = k, \mathcal{E} \right] = 1 - \prod_{i=1}^{n} \left( 1 - e^{-\Lambda(\beta, \Pi) 1(\beta < \gamma) \frac{k}{\beta} + g_i(\beta, k, \mathcal{E}_i)} \right).
\]

Let us denote \( g_{\max}(\beta, k, \mathcal{E}) \triangleq \max_{1 \leq i \leq n} g_i(\beta, k, \mathcal{E}_i) \), and \( g_{\min}(\beta, k, \mathcal{E}) \triangleq \min_{1 \leq i \leq n} g_i(\beta, k, \mathcal{E}_i) \), then we have

\[
\left( 1 - e^{-\Lambda(\beta, \Pi) 1(\beta < \gamma) \frac{k}{\beta} + g_{\max}(\beta, k, \mathcal{E})} \right) \leq \prod_{i=1}^{n} \left( 1 - e^{-\Lambda(\beta, \Pi) 1(\beta < \gamma) \frac{k}{\beta} + g_i(\beta, k, \mathcal{E}_i)} \right) 
\leq \left( 1 - e^{-\Lambda(\beta, \Pi) 1(\beta < \gamma) \frac{k}{\beta} + g_{\min}(\beta, k, \mathcal{E})} \right) ^n.
\]

Since it is easy to show that both \( g_{\max} \) and \( g_{\min} \) belong to \( o(k) \) if \( \beta < \gamma \) and \( o(1) \) otherwise, by combining the above equation, we know there exists \( g_{\min}(\beta, k, \mathcal{E}) \leq \)
$g(\beta, k, \mathcal{E}) \leq g_{\text{max}}(\beta, k, \mathcal{E})$ such that

$$
\prod_{i=1}^{n} \left(1 - e^{-\Lambda(\beta, \Pi)1(\beta<\gamma)\frac{1}{\beta} + g_i(\beta, k, \mathcal{E})}\right)
= \left(1 - e^{-\Lambda(\beta, \Pi)1(\beta<\gamma)\frac{1}{\beta} + g(\beta, k, \mathcal{E})}\right)^n,
$$

where

$$
g(\beta, k, \mathcal{E}) \in \begin{cases} 
  o(k) & \text{as} \ k \to \infty \quad \text{if} \ \beta < \gamma \\
  o(1) & \text{as} \ k \to \infty \quad \text{if} \ \beta > \gamma
\end{cases}.
$$

### B.3 Proof of Lemma B.1.2

According to Lemma B.1.1 and the definition of $f(k(n), \beta, \mathcal{E})$, we have

$$
\mathbb{P} \left[ T(n, K) > \frac{k(n)}{\beta} \mid K = k(n), \mathcal{E} \right]
= 1 - \left(1 - \frac{1}{f(k(n), \beta, \mathcal{E})}\right)^n
= 1 - \left[\left(1 - \frac{1}{f(k(n), \beta, \mathcal{E})}\right)^{f(k(n), \beta, \mathcal{E})}\right]^{n/k(n, \beta, \mathcal{E})}.
$$

Since the function $\left(1 - \frac{1}{x}\right)^x$ with domain $(1, +\infty)$ is a bounded and strictly increasing function with region $(0, e^{-1})$ and the fact that $f(k, \beta) > 1$, we know that if
\[
\lim_{n \to \infty} \frac{n}{f(k(n), \beta, \mathcal{E})} = \infty, \text{ then}
\]

\[
\liminf_{n \to \infty} P \left[ T(n, K) > \frac{k(n)}{\beta} \bigg| K = k(n), \mathcal{E} \right] = 1 - \limsup_{n \to \infty} \left[ \left( 1 - \frac{1}{f(k(n), \beta, \mathcal{E})} \right)^{f(k(n), \beta, \mathcal{E})} \right]^{\frac{n}{f(k(n), \beta, \mathcal{E})}} \\
\geq 1 - \limsup_{n \to \infty} e^{-\frac{n}{f(k(n), \beta, \mathcal{E})}} = 1,
\]

which, together with the fact that \( P \left[ T(n, K) > \frac{k(n)}{\beta} \bigg| K = k(n), \mathcal{E} \right] \leq 1 \), yields Equation (B.1.2).

If \( \lim_{n \to \infty} \frac{n}{f(k(n), \beta, \mathcal{E})} = 0 \), then \( f(k(n), \beta, \mathcal{E}) \to \infty \) as \( n \to \infty \), which results in

\[
\lim_{n \to \infty} \left( 1 - \frac{1}{f(k(n), \beta, \mathcal{E})} \right)^{f(k(n), \beta, \mathcal{E})} = e^{-1}.
\]

Then we can obtain

\[
\limsup_{n \to \infty} P \left[ T(n, K) > \frac{k(n)}{\beta} \bigg| K = k(n), \mathcal{E} \right] = 1 - \liminf_{n \to \infty} \left[ \left( 1 - \frac{1}{f(k(n), \beta, \mathcal{E})} \right)^{f(k(n), \beta, \mathcal{E})} \right]^{\frac{n}{f(k(n), \beta, \mathcal{E})}} \\
= 1 - \liminf_{n \to \infty} \left[ \lim_{n \to \infty} \left( 1 - \frac{1}{f(k(n), \beta, \mathcal{E})} \right)^{f(k(n), \beta, \mathcal{E})} \right]^{\frac{n}{f(k(n), \beta, \mathcal{E})}} \\
= 1 - 1 = 0,
\]

which leads to Equation (B.1.1).
B.4 Proof of Lemma B.1.3

Choose $\varepsilon > 0$. Since $h_n(x)$ converges to $1(x < y)$ almost everywhere, for any $\delta > 0$, we can find $N \in \mathbb{N}$ such that for any $n > N$, we have

$$|h_n (y - \delta/2) - 1| < \varepsilon$$

$$|h_n (y + \delta/2) - 0| < \varepsilon.$$

Since $0 \leq h_n(x) \leq 1$ for any $x \in [0, \infty)$ and $h_n(x)$ is a decreasing function of $x$, we know that, for any $n > M$,

$$h_n(x) > 1 - \varepsilon \quad \forall x < y - \delta/2$$

$$h_n(x) < \varepsilon \quad \forall x > y + \delta/2.$$

Therefore, for any $n > N$,

$$\nu (\{|h_n(x) - 1(x < y)| > \varepsilon\})$$

$$< \nu([y - \delta/2, y]) + \nu([y, y + \delta/2]) = \delta,$$

where $\nu$ is the Lebesgue measure. Since $\varepsilon$ and $\delta$ are arbitrarily chosen, from the above inequality we know that $h_n(x)$ converges globally in measure to $1(x < y)$. 

137
B.5 Proof of Theorem 3.4.1.

Let us define two random variables $T^{(0)}$ and $T^{(1)}$ under the Gilbert-Elliott erasure channels as

\[
T^{(0)} = \min_m \left\{ m \left| \sum_{j=1}^m X_{1j} \geq 1, X_{10} = 0 \right. \right\}, \quad (B.5.1)
\]

\[
T^{(1)} = \min_m \left\{ m \left| \sum_{j=1}^m X_{1j} \geq 1, X_{10} = 1 \right. \right\}. \quad (B.5.2)
\]

In order to prove Theorem 3.4.1, we first need the following lemma, whose proof can be found in Section B.6.

Lemma B.5.1. Let \( \{T^{(1)}_d\}_{d \in \mathbb{N}} \) be i.i.d. random variables with the same distribution as $T^{(1)}$, then we have

\[
\sum_{d=1}^{K_0+1} T^{(1)}_d \geq T^{(0)},
\]

meaning that \( \sum_{d=1}^{K_0} T^{(1)}_d \) is stochastically greater than or equal to $T^{(0)}$, where

\[
K_0 = \min \left\{ m \geq 0 \left| \sum_{d=0}^m (1 - p_{10})^dp_{10} + p_{01} \geq 1 \right. \right\}. \quad (B.5.3)
\]

With Lemma B.5.1 established, we now turn to the proof of Theorem 3.4.1. Under the Gilbert-Elliott erasure channel assumption as illustrated in Figure 3.2, let $T(n, K, \mathcal{E})$ be defined as $T(n, K)$ with initial status $\mathcal{E}$. Then according to Definitions 3.2.1 and 3.2.2 and Equations (B.5.1) and (B.5.2) we can express $T(n, K, \mathcal{E})$
as

\[ T(n, K, \mathcal{E}) = \max_{1 \leq i \leq n} \left\{ T_{i1}^{(\mathcal{E}_i)} + \sum_{j=2}^{K} T_{ij}^{(1)} \right\}, \]

where \( \{T_{ij}^{(0)}\}_{i,j \in \mathbb{N}} \) are i.i.d. random variables with the same distribution as \( T^{(0)} \), and \( \{T_{ij}^{(1)}\}_{i,j \in \mathbb{N}} \) are i.i.d. random variables with the same distribution as \( T^{(1)} \). Similarly we can express \( T(n^\alpha, \alpha K, 1_{n^\alpha}) \) and \( T(n^\alpha, \alpha K, 1_{n^\alpha}) \) as

\[ T(n, K, 1_n) = \max_{1 \leq i \leq n} \left\{ \sum_{j=1}^{K} T_{ij}^{(1)} \right\}, \quad (B.5.4) \]

\[ T(n^\alpha, \alpha K, 1_{n^\alpha}) = \max_{1 \leq i \leq n^\alpha} \left\{ \alpha K \sum_{j=1}^{T_{ij}^{(1)}} \right\}. \quad (B.5.5) \]

First, by Lemma B.5.1 we know that, for any \( 1 \leq i \leq n \) and any initial status \( \mathcal{E} \),

\[ \sum_{j=1}^{K+K_0} T_{ij}^{(1)} \geq T_{i1}^{(\mathcal{E}_i)} + \sum_{j=2}^{K} T_{ij}^{(1)}, \]

implying that

\[ \max_{1 \leq i \leq n} \left\{ \sum_{j=1}^{K+K_0} T_{ij}^{(1)} \right\} \geq \max_{1 \leq i \leq n} \left\{ T_{i1}^{(\mathcal{E}_i)} + \sum_{j=2}^{K} T_{ij}^{(1)} \right\}, \]

which yields

\[ \mathbb{E}[T(n, K + K_0, 1_n)] \geq \mathbb{E}[T(n, K, \mathcal{E})]. \quad (B.5.6) \]

Next, we will show that

\[ \mathbb{E}[T(n^\alpha, \alpha K, 1_{n^\alpha})] \geq \alpha \mathbb{E}[T(n, K, 1_n)]. \]
Let us denote

\[ S^r_i = \sum_{j=1+(r-1)K}^{rK} T^{(1)}_{ij}. \]

Then we know that \{S^r_i\}_{i \in \mathbb{N}, r \in \mathbb{N}} are i.i.d. random variables. Equation (B.5.4) and (B.5.5) can be rewritten as

\[
T(n, K, 1^n) = \max_{1 \leq i \leq n} S^1_i \tag{B.5.7}
\]

\[
T(n^\alpha, \alpha K, 1^{n^\alpha}) = \max_{1 \leq i \leq n^\alpha} \sum_{r=1}^\alpha S^r_i. \tag{B.5.8}
\]

Instead of viewing Equation (B.5.8) as a 1-dimensional maximization over \(n^\alpha\) points, we can think of it as an \(\alpha\)-dimensional maximization over \(n^\alpha\) points where we can choose a coordinate from 1 to \(n\) on each dimension and therefore can further rewrite Equation (B.5.8) as

\[
T(n^\alpha, \alpha K, 1^{n^\alpha}) = \max_{1 \leq i_1 \leq n} \max_{1 \leq i_2 \leq n} \ldots \max_{1 \leq i_\alpha \leq n} \sum_{r=1}^\alpha S^r_{(i_1,i_2,\ldots,i_\alpha)}, \tag{B.5.9}
\]

where

\[
S^r_{(i_1,i_2,\ldots,i_\alpha)} = S^r_{\sum_{u=1}^\alpha n^{u-1}(i_u-1)+1}
\]

and \(i_u\) can be viewed as the coordinate in the \(u\)th dimension.

Next, we use Equation (B.5.9) to build a lower bound on the expectation of \(T(n^\alpha, \alpha K, 1^{n^\alpha})\).

For fixed values of \(i_2, i_3, \ldots, i_\alpha\), let us find a \(i_1^*\) such that

\[
i_1^*(i_2, \ldots, i_\alpha) = \arg \max_{1 \leq i_1 \leq n} S^1_{(i_1,i_2,\ldots,i_\alpha)}, \tag{B.5.10}
\]
which we denote as $i^*_1$ for short. Then according to Equation (B.5.9), we can find a lower bound for $\mathbb{E}[T(n^\alpha, \alpha K, \mathbf{1}_{n^\alpha})]$ by choosing $i_1 = i^*_1(i_2, \ldots, i_\alpha)$ for all possible values of $i_2, i_3, \ldots, i_\alpha$, which is

$$
\mathbb{E} [T(n^\alpha, \alpha K, \mathbf{1}_{n^\alpha})]
= \mathbb{E} \left[ \max_{1 \leq i_1 \leq n} \max_{1 \leq i_2 \leq n} \ldots \max_{1 \leq i_\alpha \leq n} \sum_{r=1}^\alpha S^r_{(i_1, i_2, \ldots, i_\alpha)} \right]
\geq (a) \mathbb{E} \left[ \max_{1 \leq i_1 \leq n} \ldots \max_{1 \leq i_\alpha \leq n} \sum_{r=1}^\alpha S^r_{(i^*_1, i_2, \ldots, i_\alpha)} \right]
= \mathbb{E} \left[ \max_{1 \leq i_2 \leq n} \ldots \max_{1 \leq i_\alpha \leq n} \left( \sum_{r=2}^\alpha S^r_{(i^*_1, i_2, \ldots, i_\alpha)} + S^1_{(i^*_1, i_2, \ldots, i_\alpha)} \right) \right].
$$

(B.5.11)

Since the choice of $i^*_1$ is only sub-optimal, the inequality (a) in Equation (B.5.11) should be strict inequality. Notice that according to Equation (B.5.10), for any values of $i_2, i_3, \ldots, i_\alpha$, we have

$$S^1_{(i^*_1, i_2, \ldots, i_\alpha)} = \max_{1 \leq i_1 \leq n} S^1_{(i_1, i_2, \ldots, i_\alpha)},$$

which, combining Equation (B.5.7) and the fact that $\{S^r_{i}\}$ are i.i.d. random variables, yields

$$
\mathbb{E} \left[ S^1_{(i^*_1, i_2, \ldots, i_\alpha)} \right] = \mathbb{E} \left[ \max_{1 \leq i_1 \leq n} S^1_{(i_1, i_2, \ldots, i_\alpha)} \right]
= \mathbb{E} \left[ \max_{1 \leq i_1 \leq n} S^1_{i} \right]
= \mathbb{E} \left[ \max_{1 \leq i_1 \leq n} S^1_{(i^*_1, i_2, \ldots, i_\alpha)} \right].
$$

(B.5.12)

As a second step, for any values of $i_3, i_4, \ldots, i_\alpha$, let us define $i^*_2$ as

$$i^*_2(i_1^*, i_3, \ldots, i_\alpha) = \arg \max_{1 \leq i_2 \leq n} S^2_{(i_1^*, i_2, \ldots, i_\alpha)}.$$
Then similarly as Equation (B.5.11), by fixing \( i_2 \) to be \( i_2^* \), we can obtain

\[
E\left[T(n^\alpha, \alpha K, 1_{n^\alpha})\right] > E \left[ \max_{1 \leq i_3 \leq n} \ldots \max_{1 \leq i_\alpha \leq n} \left( \sum_{r=3}^\alpha S_{(i_1^*, i_2^*, \ldots, i_\alpha)}^r + S_{(i_1^*, i_2^*, \ldots, i_\alpha)}^1 + S_{(i_1^*, i_2^*, \ldots, i_\alpha)}^2 \right) \right].
\]

Also, for any values of \( i_3, i_4, \ldots, i_\alpha \), we have

\[
E\left[S_{(i_1^*, i_2^*, \ldots, i_\alpha)}^2\right] = E \left[ \max_{1 \leq i_2 \leq n} S_{(i_1^*, i_2^*, \ldots, i_\alpha)}^1 \right] = E \left[T(n, K, 1_n)\right]. \tag{B.5.13}
\]

By defining \( i_3^*, \ldots, i_\alpha^* \) in a similar way

\[
i_u(i_1^*, \ldots, i_{u-1}^*, i_u+1, \ldots, i_\alpha)
= \arg \max_{1 \leq i_u \leq n} S_{(i_1^*, \ldots, i_{u-1}^*, i_u, \ldots, i_\alpha)}^u\]

and iterating the above step, we can get

\[
E\left[T(n^\alpha, \alpha K, 1_{n^\alpha})\right] > E \left[ S_{(i_1^*, i_2^*, \ldots, i_\alpha)}^1 + S_{(i_1^*, i_2^*, \ldots, i_\alpha)}^2 + \ldots + S_{(i_1^*, i_2^*, \ldots, i_\alpha)}^\alpha \right]
\overset{(b)}{=} \sum_{r=1}^\alpha E\left[S_{(i_1^*, i_2^*, \ldots, i_\alpha)}^r\right]
\overset{(c)}{=} \alpha E\left[T(n, K, 1_n)\right]. \tag{B.5.14}
\]

Equation (b) follows from the fact that \( \{S_{(i_1^*, i_2^*, \ldots, i_\alpha)}^r\}_{1 \leq r \leq \alpha} \) are independent random variables and equation (c) follows from Equations (B.5.12), (B.5.13), and iterative
steps. By combining Equations (3.2.2), (B.5.6), and (B.5.14), we have,

\[ \eta(n, K) = \frac{K}{\mathbb{E}[\mathbb{E}[T(n, K, \mathcal{E})]]} \]
\[ \geq \frac{K}{\mathbb{E}[T(n, K + K_0, 1_n)]} \]
\[ > \frac{\alpha K}{\mathbb{E}[T(n^{\alpha}, \alpha(K + K_0), 1_{n^\alpha})]} \]
\[ = \frac{K}{K + K_0} \frac{\alpha(K + K_0)}{\mathbb{E}[T(n^{\alpha}, \alpha(K + K_0), 1_{n^\alpha})]}, \]

which completes the proof.

**B.6 Proof of Lemma B.5.1**

First observe that \( \sum_{d=0}^{\infty} (1 - p_{10})^d p_{10} = 1 \), which makes sure that \( K_0 \) in Equation (B.5.3) is well defined.

Then according to the definition of \( T^{(0)} \) and \( T^{(1)} \) in Equations (B.5.1) and (B.5.2), we have, for any integer \( 1 \leq t \leq K_0 + 1 \),

\[ \mathbb{P} \left[ \sum_{d=1}^{K_0 + 1} T_d^{(1)} > t \right] = 1 \geq \mathbb{P} \left[ T^{(0)} > t \right], \quad (B.6.1) \]
and for any integer \( t > K_0 + 1 \),

\[
\mathbb{P}\left[ \sum_{d=1}^{K_0+1} T_d^{(1)} > t \right] \\
\geq \sum_{d=1}^{K_0+1} \mathbb{P} [T^{(1)} > 1] \mathbb{P} [T^{(1)} = 1]^{d-1} \mathbb{P} [T^{(0)} > t - d] \\
\geq \sum_{d=1}^{K_0+1} \mathbb{P} [T^{(1)} > 1] \mathbb{P} [T^{(1)} = 1]^{d-1} \mathbb{P} [T^{(0)} > t - 1] \\
= \sum_{d=1}^{K_0+1} p_{10}(1 - p_{10})^{d-1}(1 - p_{01})^{t-1} \\
\geq (1 - p_{01})(1 - p_{01})^{t-1} = (1 - p_{01})^t = \mathbb{P} [T^{(0)} > t],
\]

with the last inequality followed by the definition of \( K_0 \) in Equation (B.5.3). The above equation, together with Equation (B.6.1), completes the proof.
C.1 Supporting Lemmas

In order to prove Theorem 4.3.2 and Theorem 4.3.3, we need the following four lemmas, whose proofs can be found in Section C.2, C.3, C.4, and C.5.

**Lemma C.1.1.** For any $S_1 \in S_{FMb}$, the underlying undirected graph of any of the weakly connected subgraphs of $G_R(S_1)$ is a planar graph. More accurately, $\forall S_1 \in S_{FMb}$ and $\forall 1 \leq m \leq M$, $\tilde{G}_R(S_1^{(m)})$ is planar.

**Lemma C.1.2.** $\forall S_1 \in S_{FMb}$ and $\forall 1 \leq m \leq M$, the following statements are true: 1) $G_R(S_1^{(m)})$ cannot have more than one node that has an indegree of zero. 2) $\tilde{G}_R(S_1^{(m)})$ can at most have one chordless cycle. 3) $\tilde{G}_R(S_1^{(m)})$ contains a chordless cycle if and only if all nodes in $G_R(S_1^{(m)})$ have indegree of one.

**Lemma C.1.3.** $\forall S_1 \in S_{FMb}$ and $\forall 1 \leq m \leq M$, $\tilde{G}_R(S_1^{(m)})$ contains an odd length cycle only if all the nodes in $N_R(S_1^{(m)})$ is operating in either mode 3 or mode 9.

**Lemma C.1.4.** For any $S_1 \in S_{FMb}$, if $G_R(S_1)$ does not contain any odd length cycle, then we can find $S_2, S_3 \in S_{MMa}$, such that $2\tilde{R}(S_1) = \tilde{R}(S_2) + \tilde{R}(S_3)$.

C.2 Proof of Lemma C.1.1

For any $S_1 \in S_{FMb}$ and any $1 \leq m \leq M$, it is easy to see that the total number of edges in $\tilde{G}_R(S_1^{(m)})$ equals to the sum of indegree of nodes in $N_R(S_1^{(m)})$, which,
by combing the fact that the maximal indegree of every node in $N_R(S_1^{(m)})$ is 1 (See Fig. C.1), implies that the total number of edges in $G_R(S_1^{(m)})$ is no larger than the total number of vertexes in $G_R(S_1^{(m)})$.

Given that in either $K_{3,3}^1$ or $K_5^2$, the total number of edges is at least one more than the total number of vertexes, we know that $G_R(S_1^{(m)})$ can never contain a subgraph that is a subdivision$^3$ of $K_5$ or $K_{3,3}$. Therefore, by Kuratowski’s theorem [52], $G_R(S_1^{(m)})$ is a planar graph.

C.3 Proof of Lemma C.1.2

From Lemma C.1.1 we know that $G_R(S_1^{(m)})$ is a planar graph for any $S_1 \in S_{FMB}$ and $1 \leq m \leq M$, then by Euler’s formula [52], the number of chordless cycles in $G_R(S_1^{(m)})$, which is equal to one fewer the number of faces in the graph, is equal to

$$1 - |N_R(S_1^{(m)})| + |E_R(S_1^{(m)})|,$$

which, by combing with the fact that the maximum indegree of every nodes in $G_R(S_1^{(m)})$ is 1, implies that $G_R(S_1^{(m)})$ cannot have more than one chordless cycle, and it has one chordless cycle if and only if every node in $G_R(S_1^{(m)})$ has an indegree of one.

On the other hand, since $G_R(S_1^{(m)})$ is a connected graph for each $m$, we know that $|E_R(S_1^{(m)})| \geq |N_R(S_1^{(m)})| - 1$, otherwise $G_R(S_1^{(m)})$ would not be a connected graph. Since $|E_R(S_1^{(m)})|$ is equal to the sum of indegree of nodes in $N_R(S_1^{(m)})$, we know that

---

$^1$ $K_{3,3}$ is the complete bipartite graph on six vertices, three of which connect to each of the other three, also known as the utility graph.

$^2$ $K_5$ is the complete graph on five vertices.

$^3$ A subdivision of a graph results from inserting vertices into edges zero or more times.
the number of nodes in \( N_R(S_1^{(m)}) \) that have an indegree of zero cannot be more than one.

### C.4 Proof of Lemma C.1.3

From Lemma C.1.2 we know that \( \tilde{G}_R(S_1^{(m)}) \) contains a cycle if and only if every node in \( G_R(S_1^{(m)}) \) has an indegree of one, i.e., if and only if every node in \( G_R(S_1^{(m)}) \) is operating in either mode 3, 4, 9, or 10. However, from Fig. C.1 we can see that nodes in mode 4 or mode 10 affect each other in pairs, meaning that any subgraph in \( G_R(S_1) \) that contains a node in mode 4 or 10 contains a cycle with length two. Therefore, \( \tilde{G}_R(S_1^{(m)}) \) contains an odd length cycle only if all the nodes in \( N_R(S_1^{(m)}) \) is operating in either mode 3 or mode 9.

---

<table>
<thead>
<tr>
<th>Receiver modes</th>
<th>Indegree</th>
<th>Outdegree</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode 2</td>
<td>0</td>
<td>≥0</td>
</tr>
<tr>
<td>mode 3</td>
<td>1</td>
<td>≥0</td>
</tr>
<tr>
<td>mode 4</td>
<td>1</td>
<td>≥1</td>
</tr>
<tr>
<td>mode 6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mode 8</td>
<td>0</td>
<td>≥0</td>
</tr>
<tr>
<td>mode 9</td>
<td>1</td>
<td>≥0</td>
</tr>
<tr>
<td>mode 10</td>
<td>1</td>
<td>≥1</td>
</tr>
</tbody>
</table>

Figure C.1: The summary of the indegree and outdegree of nodes in graph \( G_R(S_1) = \{N_R(S_1), E_R(S_1)\} \) with different modes, for any FMb-feasible schedule \( S_1 \). Observe that if the indegree of a node is 1, then the node is the intended receiver of a single data flow.
C.5 Proof of Lemma C.1.4

For a fixed $S_1 \in S_{\text{FMb}}$, if a maximal weakly connected subgraph $G_R(S_1^{(m)})$ does not contain any odd length cycle, then from Lemma C.1.2 we know that $G_R(S_1^{(m)})$ either contains an even length cycle or does not contain any cycle. Initially set $S_2^{(m)}$ and $S_3^{(m)}$ as empty sets for any $m$. We differentiate the rest of the proof between these two cases.

a) From Lemma C.1.2 we know that the maximal weakly connected subgraph $G_R(S_1^{(m)})$ has no cycle if and only if there is exactly one node in $N_R(S_1^{(m)})$ that has zero indegree. From Fig. C.1 we can see that a node has a zero indegree in $G_R(S_1^{(m)})$ if and only if it is in either mode 2, 6, or 8.

a.1) If a node in $G_R(S_1^{(m)})$ is in mode 2, then it is clear that all the nodes in $G_R(S_1^{(m)})$ are the intended receiver of only a single data flow. Since there is no cycle in $G_R(S_1^{(m)})$, we can color the nodes in $N_R(S_1^{(m)})$ as either white or black in such a way that the neighboring nodes in $G_R(S_1^{(m)})$ have different colors. Then if $B \in N_R(S_1^{(m)})$ is a black node with $(A, B)^1 \in S_1^{(m)}$, add $(A, B)^2$ to $S_2^{(m)}$. Otherwise if $B \in N_R(S_1^{(m)})$ is a white node with $(A, B)^1 \in S_1^{(m)}$, add $(A, B)^2$ to $S_3^{(m)}$.

a.2) If a node in $G_R(S_1^{(m)})$ is in mode 6, then we know that the graph is a singleton, since a node in mode 6 has both indegree and outdegree of zero (see Fig. C.1). Therefore, $S_1^{(m)} = \{(A, B)^2\}$ for some $A, B \in \mathcal{N}$, and we let $S_2^{(m)} = S_3^{(m)} = \{(A, B)^2\}$.

a.3) If a node in $G_R(S_1^{(m)})$ is in mode 8, then denote the one node that operates in mode 8 as $A$, and the two independent data streams transmitted to node $A$ as $(B, A)^1$ and $(C, A)^1$. Let us add $(B, A)^2$ to $S_2^{(m)}$ and $(C, A)^2$ to $S_3^{(m)}$. Since node $A$ is the only node in $N_R(S_1^{\text{cycle}})$ that has an indegree of zero, it is clear that all the other nodes in $G_R(S_1^{(m)})$ besides node $A$ is the intended receiver of only a single data flow. Color the nodes in $N_R(S_1^{(m)})$ that see $(B, A)^1$ as interference, if there is any, as white, and color the nodes in $N_R(S_1^{(m)})$ that see $(C, A)^1$ as interference, if
there is any, as black. Then color the rest of the nodes in \( N_R(S_1^{(m)}) \setminus \{A\} \) as either black or white in such a way that the neighboring nodes have different colors. Then, if \( D \in N_R(S_1^{(m)}) \setminus \{A\} \) is a black node with \((E, D)^1 \in S_1^{(m)}\), add \((E, D)^2\) to \(S_2^{(m)}\). Otherwise if \( D \in N_R(S_1^{(m)}) \setminus \{A\} \) is a white node with \((E, D)^1 \in S_1^{(m)}\), add \((E, D)^2\) to \(S_3^{(m)}\).

b) If \( G_R(S_1^{(m)}) \) has an even length cycle, then from Lemma C.1.2 we know that every node in \( N_R(S_1^{(m)}) \) is the intended receiver of only one data flow. Similarly as that in case a.1, we can color the nodes in \( N_R(S_1^{(m)}) \) as either black or white in such a way that the neighboring nodes in \( G_R(S_1^{(m)}) \) have different colors, and construct \( S_2^{(m)} \) and \( S_3^{(m)} \) accordingly.

According to the way \( S_2^{(m)} \) and \( S_3^{(m)} \) are constructed, it is easy to see that \( 2R_1(S_1^{(m)}) = R_2(S_2^{(m)}) + R_3(S_3^{(m)}) \). Now it remains to be shown that \( \cup_{m=1}^M S_2^{(m)}, \cup_{m=1}^M S_3^{(m)} \in S_{\text{MMa}} \).

Assume to the contrary that \( S_2 \triangleq \cup_{m=1}^M S_2^{(m)} \notin S_{\text{MMa}} \), which means we can find \((A, B)^2, (C, D)^2 \in S_2\) such that \( \{C, B\} \in E \), or \( B = D \), or \( B = C \). If \( B = D \), then \( B \) operates in mode 8 under schedule \( S_1 \). From part a.3 above, it is clear that \((A, B)^2, (C, B)^2 \) cannot be both assigned to \( S_2 \); if \( B = C \) or \( \{C, B\} \in E \), then \( B \) is affected by \( D \) and \((A, B)^1, (C, D)^1 \in S_1\). As a result, \((D, B) \in E_R(S_1)\) and \((A, B)^1, (C, D)^1 \in S_1^{(m)} \) for some \( m \). Then, it is also clear that \( B \) and \( D \) have different colors, and therefore \((A, B)^2\) and \((C, D)^2\) cannot be both in \( S_2 \). Thus, we have a contradiction, so \( S_2 \in S_{\text{MMa}} \). Similarly we can show that \( S_3 \triangleq \cup_{m=1}^M S_3^{(m)} \in S_{\text{MMa}} \).

**C.6 Proof of Theorem 4.3.2**

**Step 1**

“⇒”: This part is implied by Lemma C.1.4, because for any \( S_1 \in S_{\text{MMb}}, S_1 \in S_{\text{FMB}} \).
“⇐”: It suffices to show that if $\mathbb{D}_{\text{MMb}} \subseteq \mathbb{D}_{\text{MMA}}$, then there does not exist $S_1 \in \mathbb{S}_{\text{MMb}}$ such that $G_R(S_1)$ contains an odd length cycle. Assume to the contrary that we can find $S_1 \in \mathbb{S}_{\text{MMb}}$ such that $G_R(S_1)$ contains an odd length cycle. Denote the odd length cycle as $G_R(S_1)$. By Lemma C.1.3, we know that there is only one data flow associated with every node in $N_R(S_1)$, and each node on that cycle is in mode 9, which is prohibited in MMA constraint set. Since $\mathbb{D}_{\text{MMb}} \subseteq \mathbb{D}_{\text{MMA}}$, then for any $(A_i, A_j) \in S_1$, we can find a group of MMA-feasible schedules $\{S_2, S_3| i, j \in \mathbb{N}\}$ such that

$$\vec{R}(S_1) = \sum_i \alpha_i \vec{R}(S_2) + \sum_j \beta_j \vec{R}(S_3),$$  \hspace{1cm} (C.6.1)

where $\sum_i \alpha_i = \sum_j \beta_j = 1/2$ and $(A_i, A_j)^2 \in S_2$, $\forall i$. Since $G_R(S_1)$ is a cycle, we can find $B_i, B_j, C_i, C_j \in \mathcal{N}$ such that $\langle B_i, A_i \rangle, \langle A_j, C_j \rangle \in N_R(S_1)$ with $(B_i, B_j)^1, (C_i, C_j)^1 \in S_1$. Since $(A_i, A_j)^2$ cannot coexist with $(B_i, B_j)^1$ or $(C_i, C_j)^1$, both $(B_i, B_j)^2$ and $(C_i, C_j)^2$ are in $S_3$, $\forall j$. Now we can see that for any $D \in N_R(S_1)$ with $(D_i, D_j)^1 \in S_1$, $(D_i, D_j)^2$ is in either $S_2$, $\forall i$ or $S_3$, $\forall j$. However, since $G_R(S_1)$ is an odd length cycle, we cannot divide the nodes in $N_R(S_1)$ into two groups such that for every $\langle A_i, B_j \rangle \in E_R(S_1)$, $A_i$ and $B_j$ are in different groups. Therefore, we cannot find a set of MMA-feasible schedules $\{S_2, S_3| i, j \in \mathbb{N}\}$ such that Equation (C.6.1) holds. A contradiction is reached, and thus the proof of step 1 is complete.

**Step 2**

“A⇒”: Denote the cycle with odd length $N_l$ as $G_R(S_1)$. Let us first randomly pick a node in the cycle, say node $A_i$, and label it as 0, and denote the the label of $A_i$ as $n(A)$. Next, we label the rest of the nodes in $N_R(S_1)$ such that if $\langle B, C \rangle \in E_R(S_1)$, then $n(C) = (n(B)+2) \mod 2N_l$. Then the nodes in $N_R(S_1)$ have labels 0, 2, $\ldots$, $2N_l - 2$.  

150
By Lemma C.1.3, we know that all the nodes in $N_R(S_1^{\text{cycle}})$ is in mode 9, then
$\forall A \in N_R(S_1^{\text{cycle}}), \exists A'$ such that $(A', A)^1 \in S_1^{\text{cycle}}$. For any $A \in N_R(S_1^{\text{cycle}})$, let us label node $A'$ as $n(A') = (n(A) + 1) \mod 2N_l$. Now we have $2N_l$ nodes in the network such that a node with label $n$ is connected with two nodes with label $(n - 1) \mod 2N_l$ and $(n + 1) \mod 2N_l$ in the network topology graph.

For any $(A, B) \in E_R(S_1^{\text{cycle}})$ with $(A', A)^1 \in S_1^{\text{cycle}}$, we know that $\{A, A', A', B\} \in \mathcal{N}$ and $\{A', C\} \notin \mathcal{N}$ for any $C \in N_R(S_1^{\text{cycle}}) \setminus \{B\}$. Therefore, there is no chord between a node with even label and a node with odd label in the topology graph.

“$\Leftarrow$”: Let us construct a schedule $S_1^{\text{cycle}}$ in the following way. Initially set $S_1^{\text{cycle}}$ as empty set. For any node $A$ in the topology graph cycle that has an even label, add $(A, B)^1$ in $S_1^{\text{cycle}}$, where $n(B) = n(A) + 1$, then it is easy to check that $S_1^{\text{cycle}} \in S_{\text{MMb}}$, and $G_R(S_1^{\text{cycle}})$ is a cycle with $N_l$ nodes.

C.7 Proof of Theorem 4.3.3

Step 1

“$\Rightarrow$”: This part is proved by Lemma C.1.4.

“$\Leftarrow$”: It suffices to show that if $D_{\text{FMB}} \subseteq D_{\text{MMb}}$, then there does not exist $S_1 \in S_{\text{FMB}}$ such that $G_R(S_1)$ contains an odd length cycle and at least one node in that cycle is in mode 3. The rest of the proof is very similar to step 1 of the proof of Theorem 4.3.2, and is omitted here.

Step 2

“$\Rightarrow$”: Denote the cycle in $G_R(S_1)$ that has length $N_l + N_f$ as $G_R(S_1^{\text{cycle}})$, where $N_l$ nodes in $N_R(S_1^{\text{cycle}})$ are in mode 9 and $N_f$ nodes in $N_R(S_1^{\text{cycle}})$ are in mode 3. We first claim that $\forall (A, B) \in E_R(S_1^{\text{cycle}})$, it is not possible for both $A$&$B$ to operate in
Assume to the contrary that both $A$ & $B$ are in mode 3, then it is clear that $(C, B)^1, (B, A)^1, (A, D)^1$ are in $S_1^{\text{cycle}}$ for some $C, D \in \mathcal{N}$. However, it is easy to check that \{(C, B)^1, (B, A)^1, (A, D)^1\} $\not\in \mathcal{S}_{FMB}$. Therefore, the claim is true. Since the nodes in mode 3 can never be neighbors of each other, then $N_f$ cannot be larger than $N_l$, which, by combing the fact that $N_l + N_f$ is odd, indicates that $N_f < N_l$.

Denote the set of nodes in $N_R(S_1^{\text{cycle}})$ that operate in mode 9 as group X, and the set of node that operate in mode 3 as group Y. Let us first randomly pick a node in group X, say node $A$, and label it as 0, and denote the label of node $A$ as $n(A) = 0$. Next, we label the rest of the nodes in group X such that if $\langle B, C \rangle \in E_R(S_1^{\text{cycle}})$ with both $B$ & $C$ in group X, or if $\langle B, D \rangle, \langle D, C \rangle \in E_R(S_1^{\text{cycle}})$ with $B$ & $C$ in group X and $D$ in group Y, then let $n(C) = (n(B) + 1) \mod 2N_l$.

If $\langle A, B \rangle \in E_R(S_1^{\text{cycle}})$ with both $A$ & $B$ in mode 9, then $\exists A' \in \mathcal{N}$ such that $(A', A)^1 \in S_1^{\text{cycle}}$, and $\{A', B\} \in \mathcal{E}$. Let us label $A'$ as $n(A') = n(A) + 1$ and add $A'$ into group X; if $\langle A, B \rangle, \langle B, C \rangle \in E_R(S_1)$ with $B$ operates in mode 3, then both $A$ & $C$ are in mode 9, and $\exists A' \in \mathcal{N}$ such that $(A', B)^1, (B, A)^1 \in S_1^{\text{cycle}}$ and $\{A', C\} \in \mathcal{E}$. Let us label $A'$ as $n(A') = n(A) + 1$ and add $A'$ into group X.

Now we have a topology cycle in the network with $2N_l + N_f$ nodes, where the nodes are divided into two groups, group X with $2N_l$ nodes and group Y with $N_f$ nodes, and nodes in group X are labeled from 0 to $2N_l - 1$ sequentially.

Similarly as that in the proof of Theorem 4.3.2, we can show that there should be no chord which connect a node in group X with even label and a node in group X with odd label, and there also should be no chord connecting any node in group X with any node in group Y.

For any $\langle A, B \rangle \in E_R(S_1^{\text{cycle}})$ with $B$ in mode 3, we know that $\{A, B\}, \{B, A'\} \in \mathcal{E}$, where $A$ has an even label and $A'$ has an odd label. Therefore, the path on the topology cycle which connects any two nodes in group Y either start with an even
node in group X and end with an odd node in group X, or start with an odd node in group X and end with an even node in group X, i.e., the path on the topology cycle which connects any two nodes in group Y contains an even number of nodes in group X.

“⇐”: Let us construct a schedule $S_{1}^{\text{cycle}}$ in the following way. Initially set $S_{1}^{\text{cycle}}$ as empty set. Since any path on the topology cycle that connects any two nodes in group Y has an even number of nodes in group X, then by denoting the labels of the two neighboring nodes of a node in group Y, say node A, as $O(A)$ and $E(A)$, where $E(A)$ is an even number and $O(A)$ is an odd number, we know that either $E(A) = (O(A) + 1) \mod 2N_l$ for any $A$ in group Y, or $O(A) = (E(A) + 1) \mod 2N_l$ for any $A$ in group Y. If $E(A) = (O(A) + 1) \mod 2N_l$ for any $A$ in group Y, then denote group Z as the set of nodes in group X that have odd labels; if $O(A) = (E(A) + 1) \mod 2N_l$, then denote group Z as the set of nodes in group X that have even labels.

For any node $A$ in group Z, if the node $B$ with label $(n(A) + 1) \mod 2N_l$, is directly connected to $A$ in the topology cycle, then add $(A, B)^1$ to $S_{1}^{\text{cycle}}$. Otherwise, if $A$ is not directly connect to $B$ on the cycle, we know that there exist a node $C$ in group Y such that $\{A, C\}, \{C, B\} \in E$. Then add $(A, C)^1, (C, B)^1$ in $S_{1}^{\text{cycle}}$. It is easy to check that $S_{1}^{\text{cycle}} \in S_{\text{BF}}$ and $G_c(S_{1}^{\text{cycle}})$ is a cycle with $N_f + N_l$ nodes. See Figure C.2 for two examples on the construction of $S_{1}^{\text{cycle}}$. 
Figure C.2: Two examples of the construction of $S^\text{cycle}_1$ in step 2 of the proof of Theorem 4.3.3, given a network cycle with $2N_l + N_f$ nodes, where $N_l = 3$ and $N_f = 5$. The number beside each node in group X indicates its label.
APPENDIX D: PROOF OF CHAPTER 5

D.1 Proof of Lemma 5.4.1

To prove that $S(x) \cup M(x)$ is a cut-through feasible schedule, we need to show that for any link $(A_C, B_A) \in S(x) \cup M(x)$, $I(A) \cap R(S(x) \cup M(x))$ and $I(B) \cap T(S(x) \cup M(x))$ satisfy the conditions (C1), (C2) and (C3) in Definition 5.3.1.

First, we prove that condition (C3) always holds for any $(A_C, B_A)$ by contradiction. Assume to the contrary that we can find another link $(A_D, B_A) \in S(x) \cup M(x)$, then based on Definition 5.3.1 and Definition 5.4.1, the two links cannot be both in $S(x)$ or $M(x)$. Assume w.l.o.g. that $(A_C, B_A) \in S(x)$ and $(A_D, B_A) \in M(x)$, then it is clear that $(A_D, B_A) \in (M^3_{(x)} \cup M^4_{(x)})$ since $M^1_{(x)}, M^2_{(x)} \in S(x)$. However, according to Definition 5.4.2, $(A_D, B_A)$ cannot be included in $M^3_{(x)}$ or $M^4_{(x)}$ if $(A_C, B_A) \in S(x)$. Thus, a contradiction is reached and condition (C3) is satisfied.

For any $(A_C, B_A) \in S(x) \cup M(x)$, we prove that conditions (C1) and (C2) also hold by discussing the following three cases: (i) $(A_C, B_A) \in M^1_{(x)}$, (ii) $(A_C, B_A) \in M^3_{(x)}$, (iii) $(A_C, B_A) \in S(x) \setminus M(x)$. The discussion for the cases when $(A_C, B_A) \in M^2_{(x)}$ and $(A_C, B_A) \in M^4_{(x)}$ are similar to (i) and (ii) and, therefore, are omitted for brevity.
(i) If \((A_C, B_A) \in M^1_x\), then

\[
\mathcal{I}(A) \cap \mathcal{R}(S(x) \cup M(x)) = \mathcal{I}(A) \cap (\mathcal{R}(S(x)) \cup \mathcal{R}(M(x))) \\
= (\mathcal{I}(A) \cap \mathcal{R}(S(x))) \cup (\mathcal{I}(A) \cap \mathcal{R}(M(x))) \\
\overset{(a)}{=} (\mathcal{I}(A) \cap \mathcal{R}(S(x))) \cup \{B\},
\]

and \(\mathcal{I}(B) \cap \mathcal{T}(S(x) \cup M(x)) = (\mathcal{I}(B) \cap \mathcal{T}(S(x))) \cup (\mathcal{I}(B) \cap \mathcal{T}(M(x))) \overset{(b)}{=} (\mathcal{I}(B) \cap \mathcal{T}(S(x))) \cup \{A\}\), which, by combing the fact that \(M^1_x \subset S(x)\), imply that the condition (C1) and (C2) in Definition 5.3.1 are trivially satisfied. Equality (a) and (b) in the above equations follow from the definition of valid decision schedule in Definition 5.4.1.

(ii) If \((A_C, B_A) \in M^3_x\), then

\[
\mathcal{I}(A) \cap \mathcal{R}(S(x) \cup M(x)) \\
= (\mathcal{I}(A) \cap \mathcal{R}(S(x))) \cup (\mathcal{I}(A) \cap \mathcal{R}(M(x))) = \emptyset \cup \{B\},
\]

and \(\mathcal{I}(B) \cap \mathcal{T}(S(x) \cup M(x)) = (\mathcal{I}(B) \cap \mathcal{T}(S(x))) \cup (\mathcal{I}(B) \cap \mathcal{T}(M(x))) = (\mathcal{I}(B) \cap \mathcal{T}(S(x))) \cup \{A\}\), which, according to the definition of \(M^3_x\), either equals to \(\{A\}\), or \(\{A, D\}\) for some \(D\) with \(D_B \in \hat{T}(S(x))\). Therefore, condition (C1) and (C2) are satisfied.

(iii) Now we focus on the case when \((A_C, B_A) \in S(x) \setminus M(x)\). Since \(M^1_x, M^2_x \in S(x)\), we have

\[
\mathcal{I}(A) \cap \mathcal{R}(S(x) \cup M(x)) = \\
\mathcal{I}(A) \cap \mathcal{R}(S(x) \cup M^3_x \cup M^4_x). \\
\text{(D.1.1)}
\]

We use a two-step argument to show that condition (C1) holds. In the first step, we claim that \(\mathcal{I}(A) \cap \mathcal{R}(M^3_x \cup M^4_x) = \emptyset\) or \(\{C\}\). In the second step, we claim that
if \(|I(C) \cap T(S_{(x)})| > 1\), then \(I(A) \cap R(M_{3(x)} I \cup M_{4(x)}) = \emptyset\). The proof of condition (C2) can be obtained in a similar manner and is omitted here.

For the first claim, let us assume to the contrary that we can find a node \(D \in \mathcal{E}\backslash\{C\}\) such that \(D \in I(A) \cap R(M_{3(x)} I \cup M_{4(x)})\). Since \(D \in I(A)\) and \(A \in T(S_{(x)})\), we know that \(\{A\} \subseteq I(D) \cap T(S_{(x)})\). On the other hand, since \(D \in R(M_{3(x)} I \cup M_{4(x)})\), we have, according to the definition of \(M_{3(x)} I \cup M_{4(x)}\), that \(I(D) \cap T(S_{(x)}) = \{A\}\) implies \(A_D \in \hat{T}(S_{(x)})\). However, since \((A_C, B_A) \in S_{(x)}\) and \(A_D \in \hat{T}(S_{(x)})\), \(S_{(x)}\) cannot be a cut-through feasible schedule. Therefore, a contradiction is reached, and the first claim holds.

For the second claim, again, let us assume to the contrary that \(|I(C) \cap T(S_{(x)})| > 1\) and \(I(A) \cap R(M_{3(x)} I \cup M_{4(x)}) = \{C\}\). However, from the definition of \(M_{3(x)} I \cup M_{4(x)}\) we know that if \(C \in R(M_{3(x)} I \cup M_{4(x)})\) then \(|I(C) \cap T(S_{(x)})| \leq 1\). Therefore, a contradiction is reached and the second claim holds.

Now, let us combine the two claims. If \(I(A) \cap R(M_{3(x)} I \cup M_{4(x)}) = \emptyset\), then it is easy to see that condition (C1) holds trivially. Otherwise, according to the two claims, we must have \(I(A) \cap R(M_{3(x)} I \cup M_{4(x)}) = \{C\}\) with \(|I(C) \cap T(S_{(x)})| \leq 1\). In other words, \(I(C) \cap T(S_{(x)}) = \{A\}\), and \(I(A) \cap R(S_{(x)} \cup M_{3(x)} I \cup M_{4(x)}) = \{B, C\}\), which, by combing Equation (D.1.1), completes the proof.

### D.2 Proof of Lemma 5.4.2

Let us focus on a particular valid decision schedule \(M\), and pick two cut-through feasible schedules \(S_{(y)}, S_{(z)} \in S_{CT}\) that satisfy \(S_{(y)} \setminus M_{(y)} = S_{(z)} \setminus M_{(z)}\). We need to show that for any link \((A_C, B_A) \in M\), it is contained either in both \(M_{(y)}\) and \(M_{(z)}\), or in neither \(M_{(y)}\) nor \(M_{(z)}\). The proof can be broken down into the following four cases: (i) \((A_C, B_A) \in S_{(y)} \cap S_{(z)}\); (ii) \((A_C, B_A) \in S_{(y)}\) and \((A_C, B_A) \notin S_{(z)}\); (iii) \((A_C, B_A) \notin S_{(y)} \cup S_{(z)}\); (iv) \((A_C, B_A) \notin S_{(y)}\) and \((A_C, B_A) \in S_{(z)}\). Since the discussion
on case (i) and (ii) is similar to that on case (iii) and (iv), we omit the last two cases for brevity.

Before we start the discussion, it is worth noting, from the definition of the trimmed decision schedule in Definition 5.4.2, that whether a link \((A, C, B, A)\) is included in \(M(y)\) or not depends only on the outcome of the following three sets: 
\[
I(A) \cap R(S(y)), \quad I(B) \cap T(S(y)), \quad \text{and} \quad I(C) \cap T(M) .
\]

(i) Given that \((A, C, B, A) \in M\) and \((A, C, B, A) \in S(y) \cap S(z)\), in order to prove \((A, C, B, A)\) is either in both \(M(y)\) and \(M(z)\) or in neither \(M(y)\) nor \(M(z)\), it suffices to show that the following two claims hold: 1, \(I(A) \cap R(S(y)) = I(A) \cap R(S(z))\). 2, \(I(B) \cap T(S(y)) = I(B) \cap T(S(z))\).

Given that \(S(y) \setminus M(y) = S(z) \setminus M(y)\), it is easy to see that \((S(y) \setminus S(z)) \cup (S(z) \setminus S(y)) \subseteq M(y) \setminus (A, C, B, A) \subseteq M \setminus (A, C, B, A)\), implying
\[
I(A) \cap R ((S(y) \setminus S(z)) \cup (S(z) \setminus S(y))) \\
\subseteq I(A) \cap R (M(y) \setminus (A, C, B, A)) \\
\subseteq I(A) \cap R (M \setminus (A, C, B, A)) \overset{(a)}{=} \emptyset,
\]
which is an equivalent statement of \(I(A) \cap R(S(y)) = I(A) \cap R(S(z))\), and thus claim 1 holds. Note that equality (a) above follows from Definition 5.4.1.

Similarly, the second claim follows by noting that
\[
I(B) \cap T ((S(y) \setminus S(z)) \cup (S(z) \setminus S(y))) \\
\subseteq I(B) \cap T (M(y) \setminus (A, C, B, A)) \\
\subseteq I(B) \cap T (M \setminus (A, C, B, A)) = \emptyset.
\]

(iii) If \((A, C, B, A) \in S(y)\) and \((A, C, B, A) \not\in S(z)\), then since \(S(y) \setminus M(y) = S(z) \setminus M(z)\),
we must have \((A_C, B_A) \in \mathcal{M}_{(y)}\). Moreover, we know that \((A_C, B_A) \in \mathcal{M}_{(y)}^1 \cup \mathcal{M}_{(y)}^2\), since, by definition, \(\mathcal{M}_{(y)}^3\) and \(\mathcal{M}_{(y)}^4\) have no intersection with \(\mathcal{S}_{(y)}\). As a result, in this case, it suffices to show that \((A_C, B_A) \in \mathcal{M}_{(z)}^3\), which can be proved by showing that the following two claims hold: 1, if \((A_C, B_A) \in \mathcal{M}_{(y)}^1\), then \((A_C, B_A) \in \mathcal{M}_{(z)}^3\). 2, if \((A_C, B_A) \in \mathcal{M}_{(y)}^2\), then \((A_C, B_A) \in \mathcal{M}_{(z)}^4\). The proofs of the two claims are similar and we only show the first one.

If \((A_C, B_A) \in \mathcal{M}_{(y)}^1\) and \((A_C, B_A) \notin \mathcal{S}_{(z)}\), then using the argument we have for the previous case, we can obtain

\[
\begin{align*}
\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(y)} \setminus (A_C, B_A)) &= \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(z)}), \\
\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(y)} \setminus (A_C, B_A)) &= \mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(z)}),
\end{align*}
\]

which, by combing the fact that \((A_C, B_A)\) is a cut-through feasible schedule, yields

\[
\begin{align*}
\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(z)}) &= \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(y)}) \setminus \{B\} = \emptyset, & (D.2.1) \\
\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(z)}) &= \mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(y)}) \setminus \{A\} = \emptyset \text{ or} & (D.2.2) \\
& \{D\} \text{ for some node } D \text{ with } D_B \in \hat{T}(\mathcal{S}_{(y)}).
\end{align*}
\]

Finally, if \(\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(z)}) = \{D\} \) with \(D_B \in \hat{T}(\mathcal{S}_{(y)})\), then \(D_B\) must also be in \(\hat{T}(\mathcal{S}_{(z)})\). Assume to the contrary that \(D_B \notin \hat{T}(\mathcal{S}_{(z)})\), then from the fact that \(\mathcal{S}_{(y)} \setminus \mathcal{M}_{(y)} = \mathcal{S}_{(z)} \setminus \mathcal{M}_{(y)}\), we know that \(D \in \mathcal{R}(\mathcal{M}_{(y)})\), which makes \(\mathcal{I}(B) \cap \mathcal{R}(\mathcal{M}_{(y)}) = \{A, D\}\), violating the fact that \(\mathcal{M}_{(y)}\) is a valid decision schedule. This argument, by combining with Equation (D.2.1) and (D.2.2), implies that \((A_C, B_A) \in \mathcal{M}_{(z)}^3\), which completes our proof.


