ABSTRACT

Title of dissertation: DISTRIBUTED WIRELESS MULTICAST: THROUGHPUT AND DELAY

Brooke Shrader, Doctor of Philosophy, 2008

Dissertation directed by: Professor Anthony Ephremides
Department of Electrical and Computer Engineering

Multicast transmission, in which data is sent from a source to multiple destinations, is an important form of data communication in wireless networks. Numerous applications require multicast transmission, including content distribution, conferencing, and military and emergency messages, as well as certain network control mechanisms, such as timing synchronization and route establishment. Finding a means to ensure efficient, reliable multicast communication that can adapt to changing channel conditions and be implemented in a distributed way remains a challenging open problem. In this dissertation, we propose to meet that challenge through the use of random coding of data packets coupled with random access to a shared channel. We present an analysis of both the throughput and delay performance of this scheme.

We first analyze the multicast throughput in a random access network of finitely many nodes, each of which serves as either a source or a destination node. Our work quantifies throughput in terms of both the Shannon capacity region and

the stable throughput region and indicates the extent to which a random linear coding scheme can outperform a packet retransmission scheme. Next, we extend these notions to a random access network of general topology in which each node can act as a receiver or a sender for multiple multicast flows. We present schemes for nodes in the network to compute their random access transmission probabilities in such a way as to maximize a weighted proportional fairness objective function of the multicast throughput. In the schemes we propose, each node can compute its transmission probability using information from neighboring nodes up to two hops away.

We then turn our focus to queueing delay performance and propose that random coding of packets be modeled as a bulk-service queueing system, where packets are served and depart the queue in groups. We analyze within this framework two different coding schemes: one with fixed expected coding rate and another with a coding rate that adapts to the traffic load. Finally, we return to the question of multicast throughput and address the effects of packet length, overhead, and the time-varying nature of the wireless channel.

DISTRIBUTED WIRELESS MULTICAST: THROUGHPUT AND DELAY

by

Brooke Erin Shrader

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2008

Advisory Committee:
Professor Anthony Ephremides, Chair/Advisor
Professor Alexander Barg
Professor Richard La
Professor Aravind Srinivasan
Professor Sennur Ulukus

Acknowledgments

I wish to express my gratitude first and foremost to my advisor, Anthony Ephremides, who has helped me immeasurably by providing his professional and technical guidance as well as many bits of personal advice. I'd also like to thank the members of my committee - Alexander Barg, Richard La, Aravind Srinivasan, and Sennur Ulukus - for investing their time in reviewing and questioning the work in this thesis. Many other faculty and staff members at the University of Maryland have lent me their time and support, for which I am truly grateful.

I am privileged to work with wonderful collaborators, including Matthew Andrews, Randy Cogill, and Haim Permuter; I've learned immensely from each one of them.

My talented colleagues and treasured friends at the University of Maryland have made the past five years stimulating and enjoyable.

To the concert pianist next door, thank you for your focus and your failure to notice your audience on the other side of the wall.

I am most appreciative of the members of my family - my many parents, sisters, nephews, and in-laws - for their constant support and occasional torment. I must especially thank Mo, whose professional and personal accomplishments have always been inspiring; and Drew, who keeps her on the ground and who picked me up off it. And finally to Răzvan: ce bine că eşti.

Table of Contents

Chapter 1

Introduction

Multicast transmission, in which data is sent from a source to multiple destinations, is an important form of data communication in wireless networks. Numerous applications require multicast transmission, including content distribution, conferencing, and military and emergency messages, as well as certain network control mechanisms, such as timing synchronization and route establishment. In addition to practical interest, there is significant theoretical interest in multicast transmission due to its use in demonstrating the benefits of the emerging paradigm of *network coding* [1, 2], in which data from disparate sources is coded together at relay nodes in a network. Multicast transmission is also an arena for studying wireless cross-layer design and the ultimate limits on network capacity. The research carried out as part of this thesis is meant to unify both the practical and theoretical aspects of wireless multicast. In doing so, we consider the form of redundancy needed to provide reliable communication, the need for distributed operation in practical networks, and the nature of the wireless channel.

For purposes of reliability in multicasting, some form of *redundancy* must be introduced. As examples, redundancy might be added through retransmitting lost packets or through some more general form of channel coding. How much redundancy should be introduced and what form it should take are difficult questions.

Clearly, there must be enough redundancy to provide sufficiently reliable transmissions, yet too much redundancy would result in an inefficient flooding of data in the network. Limiting redundancy will also result in limiting the amount of energy expended through transmission, which is a major concern for wireless nodes operating on battery power [3].

Distributed implementation is important in wireless networks, particularly infrastructureless or ad hoc networks. In our work, we consider multiple sources transmitting over a shared channel, which requires some form of medium access control (MAC). We choose to focus our studies on the use of *random access* to the channel [4], which can be implemented in a purely distributed fashion. In the random access protocol we consider, each transmitting node decides independently and randomly whether it will transmit in each time slot. Clearly, this leaves open the possibility that multiple nodes will decide to transmit simultaneously; the resulting *collision* on the channel may cause a level of interference that makes reception impossible. In order to limit or resolve collisions, more complex MAC protocols might be constructed on top of the basic random access protocol we consider (eg, carrier sensing or random access with collision resolution). However, we choose to focus on a very simple form of random access which lends itself to a thorough analysis while still capturing the effects of contention in wireless multicast.

The nature of the wireless channel directly impacts multicast transmission. Assuming that the transmitter makes use of an omnidirectional antenna, a single transmission may automatically reach multiple surrounding receivers, a property referred to as the *wireless multicast advantage* [5]. However, the widespread signal propagation over the wireless channel also means that a transmission will interfere with other nearby transmissions. As such, we make a point to specifically account for interference in multicast transmissions. Furthermore, signal fading, node mobility, and the ability to create or destroy links by adjusting data rates and/or transmission power all indicate that specifying the network as a set of links may not be the most accurate description. Therefore, we avoid the use of tree-based or graph-based models typically used in studying wired multicast.

1.1 Random coding

Reliable communication requires some form of redundancy to overcome errors introduced by the channel. As an example, the form of redundancy typically used for wired and wireless unicast transmission is to resend data packets that are known to have been received in error. This type of retransmission protocol can be particularly inefficient for multicast communication since each destination may require a different packet to be resent. Alternatively, error control coding can be introduced by sending additional redundant packets with the aim that each destination receive one or more redundant packets in order to recover the original transmitted packets that have been received in error. The typical approach to error control coding is for it to be performed at a fixed rate, or at a fixed proportion of redundant information. The fixed rate is chosen based on the error rate of the channel, which must be known or estimated. However, fixed rate coding can be problematic on a wireless channel, where the error rate of the channel may change with signal fading or mobility, and particularly problematic for multicast transmission, where the channel to different destinations may introduce different amounts of error. If the coding rate is fixed according to the worst-case channel conditions, it may be inefficient and involve too much redundant information when the channel is good. If, however, the coding rate is fixed at a high value but the channel introduces more errors than expected, the transmitted data may never be received.

The shortcomings of retransmissions and fixed-rate coding for wireless multicast may be overcome by the use of a random or rateless code. This notion has recently arisen in the form of fountain codes and random linear network codes. Fountain codes were invented by Luby in [6] and further developed by Shokrollahi [7] for the purpose of multicast communication and involve sending random linear combinations of data packets rather than the original data packets themselves. By sending a potentially limitless stream of random linear combinations of data packets, the effective rate of the code is adapted to the needs of each receiving destination node. In terms of network coding, the new paradigm introduced in [1] for wired multihop networks requires that bottleneck links in the network be identified as coding points. The signal variations and mobility introduced in wireless communication mean that the location of bottleneck links may change and a lack of centralized control may make it difficult to identify those coding points. As a solution, the notion of random network coding was introduced in [8], in which random linear combinations of data are sent through the network. The performance of random coding for wireless multicast is the primary subject of this dissertation.

To introduce the notion of random coding, we describe the technique of random linear coding discussed in [9]. This is the random coding approach we consider in much of this dissertation. Let $s_1, s_2, \ldots, s_K$ denote K data packets awaiting transmission from a source node in a wireless network. These data packets are symbols from a finite field $\mathbb{F}_u$, where we refer to u as the alphabet size and assume throughout that u is a power of two. Equivalently, the data packets $s_1, s_2, \ldots, s_K$ can be viewed as strings of $\log_2 u$ bits. Random linear coding is performed in the following manner. A coded packet is randomly generated as

$$\sum_{i=1}^{K} a_i s_i \tag{1.1}$$

where the coefficients a_i, $i = 1, 2, \ldots, K$, are generated randomly and uniformly from the set $\{0, 1, 2, \ldots, u-1\}$ and $\sum$ denotes addition in the finite field $\mathbb{F}_u$. We note at this point that it is possible to generate $a_i = 0$ for all $i = 1, 2, \ldots, K$, meaning that the coded packet is an all-zero packet and contains no information. The outcome could lead to inefficiencies, particularly for small values of K where it is more likely, and would clearly be avoided in a practical system. A coded packet formed according to (1.1) is the same length as one of the original packets s_i and can be transmitted over the channel within the same time period and utilizing the same bandwidth as one of the data packets s_i. For each coded packet sent, the coefficients a_i used in generating the packet will be appended to the header of the packet. For each transmission, the source can generate a new coded packet according to (1.1) and send it over the channel.

With each coded packet it receives, a destination node collects a u-ary column of length K in a matrix, where the column consists of the coefficients a_i,

$i = 1, 2, \ldots, K$. If enough coded packets are received, the destination can decode to recover the original K data packets by solving a system of linear equations, or equivalently, by Gaussian elimination. We let N denote the number of random linear combinations needed in order to decode. As the source may generate the same coded packet twice, or may generate a coded packet that is linearly dependent on the coded packets already received at the destination, N is a random variable that must be at least as large as K. Let $F_K(j)$ denote the cumulative distribution function (cdf) of N, or the probability that the number of coded packets needed for decoding is less than or equal to j. Then $F_K(j)$ is the probability that decoding can be performed if the matrix has at most j columns. Thus

$$F_K(j) = \Pr\{\text{a random } K \times j \text{ } u\text{-ary matrix has rank } K\}. \tag{1.2}$$

Note that for $j < K$ the matrix cannot possibly have rank K and $F_K(j) = 0$. For $j \geq K$, we can write an expression for $F_K(j)$ following the procedure in [9] by first counting up the number of $K \times j$ non-singular u-ary matrices, which is given by

$$(u^j - 1)(u^j - u)(u^j - u^2) \ldots (u^j - u^{K-1}). \tag{1.3}$$

In the product above, each term accounts for a row of the $K \times j$ u-ary matrix and reflects that the row is neither zero, nor equal to any of the previous rows, nor equal to any linear combination of previous rows. Since the total number of $K \times j$ u-ary matrices is u^{jK}, we obtain

$$F_K(j) = \begin{cases} \prod_{i=0}^{K-1} (1 - u^{-j+i}) & j \geq K \\ 0 & j < K \end{cases}. \tag{1.4}$$

The probability mass function (pmf) of N, or the probability that decoding can be performed when the destination has received *exactly* j columns and no fewer, is given by

$$f_K(j) = F_K(j) - F_K(j-1). \tag{1.5}$$

From (1.4) and (1.5) the expected value of N can be computed, where $E[N] \geq K$. The ratio $E[N]/K$ can be expressed as

$$\frac{E[N]}{K} = \prod_{i=0}^{K-1}(1-u^{-K+i}) + \frac{1}{K}\sum_{j=K+1}^{\infty} j\left(\prod_{i=0}^{K-1}(1-u^{-j+i}) - \prod_{i=0}^{K-1}(1-u^{-j+1+i})\right).$$

We note that $E[N]/K \to 1$ as $K \to \infty$ (with u fixed) and that $E[N]/K \to 1$ as $u \to \infty$ (with K fixed). This means that as either K or u grow large, the random linear code becomes more efficient in the sense that the number of coded packets needed for decoding approaches K, the number of original data packets s_i that are being transmitted.

In this dissertation, we investigate the performance of random coding for multicast transmission in a wireless network. Our assumptions regarding the use of random coding differ from its envisioned use as described in [6, 7] in that we consider a finite time-horizon in which the sender cannot send coded packets indefinitely, but must eventually move on to transmitting information about other data it wants to communicate, which is more in line with the use described in [8]. The use of random coding in this manner allows us to examine the notion of queueing delay and means that our results are applicable to data as it traverses relay nodes in a network. Also in contrast to [6, 7], we assume that there is feedback information available on the channel, so that the destination nodes can signal to the sender when they are able to decode.

1.2 Random access

We are interested in the scenario in which multiple transmitting nodes access a shared channel through the use of *random access*. Random access is of particular interest because it is robust to variations in the channel and network topology and can be implemented without coordination among the transmitting nodes. The form of random access we consider is a slotted version of Abramson's ALOHA protocol [4]. We assume that within the network, there is a common time frame established among the nodes so that all nodes maintain a common notion of time evolving in synchronized time slots. In each time slot, if node n has some data to send, it accesses the channel with probability p_n. We can view this protocol as one in which a transmitting node flips a coin and the outcome of its coin flip determines whether or not the node will access the channel. The value of p_n can be chosen by the node using information it has collected regarding previous transmissions and the level of interference those transmissions suffered. Ideally, the value of p_n can be adapted in time to account for varying channel conditions. In this dissertation, we analyze performance assuming a fixed value of p_n, but present results on performance for any value of p_n between zero and one.

1.3 Performance measures

1.3.1 Throughput

A primary measure for evaluating the performance of a network is the rate at which data can be communicated. For a point-to-point communication channel, the data rate is typically described in terms of the bits per second that can be sent. In this dissertation, we assume that data travels in the form of packets and that a common time reference is established among the users. As such we describe the data rate in terms of packets or bits per transmission or timeslot.

We consider three different theoretical frameworks leading to three different notions of data rate; each of these notions has a different meaning and applicability. The measures of data rate relate to different perspectives taken by the research community: from the *Networking perspective* we consider stable throughput and saturated throughput and from the *Information-theoretic perspective* we examine the Shannon capacity. For the stable throughput, we consider a system in which data packets randomly arrive at source nodes, packets are queued in infinite-capacity buffers when the sources are busy, and the goal is to successfully transmit the packets while ensuring that the queues at the sources remain stable. On the other hand, the saturated throughput assumes that the sources always have packets to transmit and simply aim to successfully transmit the packets at the highest rate possible. Finally, the Shannon capacity looks at an asymptotic regime in which data is coded before transmission over the channel and the probability of error at the destination must approach zero as the number of channel uses grows. Through examining these three measures, our work sheds light on the relation between the stable throughput, the saturated throughput, and the Shannon capacity.

Of particular interest, there is evidence that these three very different notions of data rate may be equivalent, and this possibility is explored in our work. As a simple illustration, consider an erasure channel where the parameter ϵ denotes the probability with which a transmission on the channel is lost; with probability $1 - \epsilon$ the transmission is received without error. The Shannon capacity of the erasure channel is $1 - \epsilon$ bits per channel use for a channel with binary inputs and outputs. If feedback is available to notify the sender when a channel input is erased, then the capacity can be achieved by retransmitting lost inputs [10]. Alternatively, the erasure channel can be viewed in a networking framework. Suppose we view the channel input as a packet of fixed length that arrives at a random time to a sender and is stored in an infinite-capacity buffer while awaiting its turn to be sent over the channel. The packets in the buffer form a queue that is emptied in first-in-first-out (FIFO) order. A retransmission protocol provides a form of redundancy to ensure that the packet is eventually received at the output of the erasure channel. If we assume that packets arrive to the sender according to a Bernoulli process and that feedback is available to notify the sender of lost packets, then the buffer at the sender forms a discrete-time $M/M/1$ queue [11] with maximum departure rate $1-\epsilon$. The stable throughput is $1 - \epsilon$, which is identical to the Shannon capacity. In this dissertation we often model the wireless channel as an erasure channel (generalized to account for interference and multiple users) and explore the relation between these different notions of throughput.

1.3.2 Multicast throughput

For unicast transmission, the throughput in packets per transmission clearly indicates the data rate between one sender and one receiver. For multicast transmission, the notion of throughput may have different meanings depending on whether it expresses the data rate from the perspective of the transmitter or from the perspective of a receiver. From the perspective of the transmitter, a multicast throughput of one packet per slot would indicate that the transmitter can reliability communicate one packet per slot to *all* intended receivers, and implies that the application requires that all intended receivers must receive the data. From the perspective of a receiver, a multicast throughput of one packet per slot would indicate the data rate to one receiver and another receiver may obtain an additional throughput for receiving the same data. The receiver-oriented notion of multicast throughput implies that it is not necessarily required for all destination nodes to receive the data, but more throughput "credit" is awarded for each destination that does. We explore both of these notions of multicast throughput in our work.

1.3.3 Delay

In addition to the throughput, we can also measure the performance of the wireless network in terms of the amount of time (in seconds or time slots) needed for data to be communicated. This notion of delay is closely linked with the data rate of the network. It includes the time that elapses between the first attempt at transmitting the packet over the channel and the instant at which successful reception is acknowledged at the receiver. In addition, in our work we place emphasis in understanding the queueing delay, or the time that elapses between the generation or arrival of data at the transmitter and the instant at which the transmitter becomes available to transmit or serve that data.

1.4 Outline of the thesis

The work we present in this dissertation is organized as follows. In Chapter 2 we analyze the stable throughput, the saturated throughput, and the Shannon capacity of a random access multicast system consisting of finitely many nodes, each of which can act *either* as a source node or as a destination node. We consider a scenario in which the destination nodes are common to *all* source nodes, and we carry out a performance analysis of both a retransmission protocol and random linear coding. Chapter 3 deals with the saturated throughput of a wireless random access network of arbitrary size and connection topology in which each node can act as a transmitter or a receiver and can serve multiple multicast flows. We consider both transmitter-oriented (or guaranteed) and receiver-oriented (or non-guaranteed) multicast and present distributed schemes for transmitters to choose their random access transmission probabilities p_n. In Chapter 4 we explore the queueing delay associated with random linear coding of packets that randomly arrive to a source node. We present a coding scheme that adapts to the amount of traffic at the source and analyze its performance. Finally, in Chapter 5, we revisit the transmitter-oriented multicast throughput and characterize performance with regard to the packet length, random coding overhead, and time-varying nature of the wireless channel. A summary of our contributions and a description of future work are included in Chapter 6.

Chapter 2

Throughput and capacity regions in random access multicast

2.1 Background

In previous work, the throughput and capacity regions for *unicast* transmission in random access networks have been analyzed and compared. The finite-user, buffered random access problem was first formulated by Tsybakov and Mikhailov [12], who provided a sufficient condition for stability and thus a lower bound on the maximum stable throughput. The problem they considered was a system in which finitely many source nodes with infinite-capacity buffers (forming queues) randomly access a shared channel to send messages to a central station. Feedback was used to notify the source nodes of failed transmissions and a retransmission scheme was used to ensure eventual successful reception at the central station. The users were assumed to access a *collision channel*, in which transmission by more than one source results in the loss of all transmitted packets with probability 1. This collision channel model is equivalent to an erasure channel, where for user n, $1 - \epsilon = p_n \prod_{j \neq n}(1 - p_j)$. Further progress on this problem was made in [13], in which stochastic dominance arguments were explicitly introduced to find the stable throughput region for a system with 2 source nodes, and in [14], wherein a stable throughput region based on the joint queue statistics was found for 3 source nodes. An exact stability result for arbitrarily (but finitely) many users has not been found, but bounds have been obtained in [14] and [15]. Recently, the authors of [16] improved upon the collision channel model used in previous works and studied a channel with multipacket reception (MPR) capability. They showed that the stable throughput region transfers from a non-convex region under the collision channel model to a convex region bounded by straight lines for a channel with MPR.

The Shannon capacity region of a random access system was considered in [17] and [18], which both obtained the capacity region for finitely many source nodes transmitting to a central station under the collision channel model. That capacity result can be viewed as the capacity of an asynchronous multiple access channel, which was obtained in [19] and [20]. The more recent contribution of [21] shows explicitly how the random access capacity region in [17, 18] is obtained from the results in [19, 20], in addition to analyzing the capacity for a channel in which packets involved in a collision can be recovered.

The relation between the stable throughput, the saturated throughput, and the Shannon capacity regions have been explored in previous works on unicast transmission in random access networks. It was noted by Massey and Mathys [17] and Rao and Ephremides [13] that the stable throughput and Shannon capacity regions coincide for the special cases of two source nodes and infinitely-many source nodes. This observation is surprising in that it suggests that the bursty nature of arriving packets, which is captured in the stability problem but not in the capacity problem, is immaterial in determining the limits on the rate of reliable communication. It has been conjectured that for finitely-many source nodes transmitting to a central station, the stable throughput region (which is an unsolved problem) coincides with both the saturated throughput and capacity regions. This conjecture was explored in [22], in which it was shown that the stable throughput coincides with the capacity in a special case involving correlated arrivals of packets at the source nodes. Recently, further progress was made in [23], which explored transmission over a channel with MPR. That work showed that for finitely-many source nodes transmitting to a central station, the saturated throughput and capacity regions coincide.

In this chapter we explore the stable throughput, saturated throughput, and capacity regions of random access for *multicast* transmission. We first consider a small network consisting of two source nodes and two destination nodes and a channel model that allows for MPR. After characterizing the capacity region of this system, we contrast the throughput (stable throughput and saturated throughput) for two different transmission schemes: a retransmission scheme and random linear coding. Next, we consider a network consisting of N source nodes and M destination nodes. We provide inner and outer bounds on the stable throughput region and an exact result on the saturated throughput, all in the context of a retransmission scheme.

2.2 A network of two sources and two destinations

The system of two source nodes, indexed by n, and two destination nodes, indexed by m, that we consider is shown in Fig. 2.1. The data generated at source $n = 1$ is assumed independent of the data generated at source $n = 2$. Time is slotted; one time slot corresponds to the amount of time needed to transmit a single

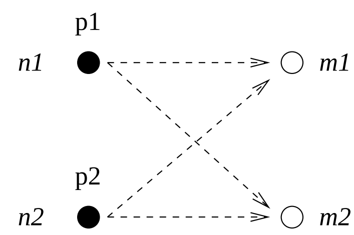

Figure 2.1: Two source nodes randomly access the channel to multicast to two destination nodes. When both sources transmit, which happens in a slot with probability $p_1 p_2$, the reception probabilities are as shown above.

fixed-length packet over the shared channel. In each time slot, if source n has a packet to transmit, then we refer to the source as being *backlogged*; otherwise the source is *empty*. A backlogged source transmits in a slot with probability p_n. We refer to p_n as the transmission probability; it encapsulates random access to the channel. We assume the value of p_n to be fixed in time. (I.e., we do not assume retransmission control, in which p_n is varied over time according to the history of successful transmissions.)

The channel model we consider is similar to the model used in [16]. A transmitted packet is received without error with a certain probability. Otherwise, the packet is lost and cannot be recovered. We assume that the channels between different source-destination pairs are independent. We introduce the following *reception probabilities* for sources $n = 1, 2$ and destinations $m = 1, 2$.

$$q_{n|n}^{(m)} = \Pr\{\text{packet from } n \text{ is received at } m | \text{ only } n \text{ transmits}\} \tag{2.1}$$

$$q_{n|1,2}^{(m)} = \Pr\{\text{packet from } n \text{ is received at } m | \text{ both sources transmit}\} \tag{2.2}$$

We assume throughout that interference cannot increase the reception probability on the channel, i.e., $q_{n|n}^{(m)} > q_{n|1,2}^{(m)}$. The reception probabilities inherently account for interference and also allow for multipacket reception (MPR). Note that these probabilities can capture the effects of fading on the wireless channel by setting them equal to the probability that a fading signal, represented by a random variable, exceeds a certain signal to interference plus noise (SINR) threshold. The collision channel model used in a number of previous works is given by $q_{n|n}^{(m)} = 1$, $q_{n|1,2}^{(m)} = 0$.

2.2.1 Shannon capacity region

We now describe the system shown in Fig. 2.1 in an information-theoretic framework in order to characterize the Shannon capacity region. We carry this out in two steps: we first characterize the capacity of a general discrete memoryless channel, and then evaluate the result to determine the capacity of the random access system with the channel model described in Eqns. 2.1 and 2.2.

The discrete memoryless channel we consider consists of discrete alphabets $\mathcal{X}_1, \mathcal{X}_2, \mathcal{Y}_1$, and $\mathcal{Y}_2$ and transition probability matrix $W(y_1, y_2|x_1, x_2)$. The channel can be decomposed into two multiple access channels, each corresponding to a destination node and defined as follows.

$$W_1(y_1|x_1, x_2) = \sum_{y_2 \in \mathcal{Y}_2} W(y_1, y_2|x_1, x_2) \tag{2.3}$$

$$W_2(y_2|x_1, x_2) = \sum_{y_1 \in \mathcal{Y}_1} W(y_1, y_2|x_1, x_2) \tag{2.4}$$

We assume that there is no feedback available on the channel. Source node n generates a sequence of messages $J_n^1, J_n^2, \ldots$ where the t^{th} message J_n^t takes values from the message set $\{1, 2, \ldots, 2^{NR_n}\}$. Here N denotes the blocklength and R_n the rate of source n. The messages are chosen uniformly from $\{1, 2, \ldots, 2^{NR_n}\}$ and independently over n. The encoding function f_n at source n is given by the mapping

$$f_n : \{1, 2, \ldots, 2^{NR_n}\} \rightarrow \mathcal{X}_n^N, \quad n = 1, 2. \tag{2.5}$$

The encoder output consists of a sequence of codewords $X_n(J_n^t)$, $t \geq 1$. The system is asynchronous in the following sense. Each source and each destination maintain a clock. Let S_{nm} denote the amount of time that the clock at source n is running ahead of the clock at destination m. The S_{nm} are assumed to be integers, meaning that time is discrete and transmissions are symbol-synchronous. The time at each clock can be divided into *periods* of length N corresponding to the length of a codeword. Let D_{nm} denote the offset between the start of periods at source n and destination m, where $0 \leq D_{nm} \leq N-1$. We assume that D_{nm} are uniform over $[0, 1, \ldots, N-1]$ for all N. The codeword $X_n(J_n^1)$ is sent at time 0 on the clock at source n.

A sequence of channel outputs are observed at each destination, where the outputs at destination m each take values from the alphabet $\mathcal{Y}_m$. The decoder operates on a sequence of $N(T+1)$ channel outputs to form an estimate of a sequence of $T+1$ messages. A decoder is defined as follows.

$$\phi_{nm} : \mathcal{Y}_m^{N(T+1)} \rightarrow \{1, 2, \ldots, 2^{NR_n}\}^{T+1}, \quad n, m = 1, 2 \tag{2.6}$$

where $\{1, 2, \ldots, 2^{NR_n}\}^{T+1}$ denotes the $(T+1)$-fold Cartesian product of $\{1, 2, \ldots, 2^{NR_n}\}$. Since the decoder must synchronize on a particular source n, the decoding function is defined separately for each source. The output of the decoder is a sequence of message estimates $\hat{J}_{nm}^1, \hat{J}_{nm}^2, \ldots, \hat{J}_{nm}^{T+1}$, where $\hat{J}_{nm}^t$ denotes the estimate at destination m of the t^{th} message sent by source n. The error criterion we consider is the average probability of error P_e^t defined as

$$P_e^t = \Pr\left\{\bigcup_m \bigcup_n \{J_n^t \neq \hat{J}_{nm}^t\}\right\}. \tag{2.7}$$

The rate pair (R_1, R_2) is *achievable* if there exists encoding functions (f_1, f_2) and decoding functions $(\phi_{11}, \phi_{12}, \phi_{21}, \phi_{22})$ such that $P_e^t \to 0$ for all t as $N \to \infty$. The capacity region is the set of all achievable rate pairs.

In order to establish the capacity region, we first present a lemma that deals with the error probability for a compound channel. Let P_m^t denote the error probability for the t^{th} message from the sources at receiver m.

$$P_m^t \triangleq \Pr\left\{\bigcup_n \{J_n^t \neq \hat{J}_{nm}^t\}\right\}. \tag{2.8}$$

The following lemma provides a condition equivalent to $P_e^t \to 0$ and is used to establish the capacity region.

Lemma 1. *The average error* $P_e^t \to 0$ *if and only if* $\max_m P_m^t \to 0$.

Proof. The average error P_e^t can be upper bounded by the union bound as follows: $P_e^t \leq P_1^t + P_2^t \leq 2\max_m P_m^t$. A similar lower bound also holds, namely $P_e^t \geq \max_m P_m^t$. Thus $\max_m P_m^t \leq P_e^t \leq 2\max_m P_m^t$ and the result follows. $\square$

The model we consider here is a compound version [24] of the *totally asynchronous multiple access channel* treated in [19] and [20]. As shown in those works, the asynchrony in the system results in the lack of a convex hull operation over the achievable rates, and this holds as well in our compound version of the problem. The capacity region is presented below.

Theorem 2. *The capacity region of the asynchronous compound multiple access channel is the closure of all rate points* (R_1, R_2) *that lie in the region*

$$\bigcap_m \left\{ (R_1, R_2) : \begin{array}{rcl} R_1 & < & I(X_1; Y_m | X_2) \\ R_2 & < & I(X_2; Y_m | X_1) \\ R_1 + R_2 & < & I(X_1, X_2; Y_m) \end{array} \right\}$$

for some product distribution $P(x_1)P(x_2)W$.

Proof. Achievability for our system is shown by first establishing achievability for the MAC W_m. This is shown in [20] and [19]; the approach presented in [20] is summarized here. Each codeword symbol in the codebook for source n is generated according to the distribution $P(x_n)$, independently over codeword symbols and independently over messages. The following two properties are assumed.

(I) The codewords $x_n(1)$ and $x_n(2)$ are reserved for use as preambles. A preamble is sent after every sequence of T messages and $x_n(1)$ and $x_n(2)$ are used as preambles in an alternating fashion. In [20] it is shown that by using the preamble, the receiver can synchronize on codeword boundaries with arbitrarily small probability of synchronization error.

(II) In a sequence of $T + 1$ messages (including a preamble), no messages are repeated. As a result, any two disjoint subsets of $N(T+1)$ codeword symbols (corresponding to $T + 1$ messages) are independent. For $T \ll 2^{NR_n}$ the resulting loss in rate is negligible.

By observing the channel outputs, the decoder ϕ_{1m} can detect the preambles $x_1(1)$ and $x_1(2)$ to determine that the output symbols in between correspond to inputs $x_1(j_1^1), x_1(j_1^2), \ldots, x_1(j_1^T)$. Let x_1^+ denote the sequence of $N(T + 1)$ symbols corresponding to $x_1(j_1^1), x_1(j_1^2), \ldots, x_1(j_1^T)$ augmented by portions of the preambles $x_1(1)$ and $x_2(2)$. At the channel output, x_1^+ will overlap with a sequence x_2^+ consisting of $N(T+1)$ symbols input by source 2, including N preamble symbols. Let $y_m^{N(T+1)}$ denote the output sequences corresponding to x_1^+ and x_2^+ at destination m. The decoder ϕ_{1m} outputs the unique sequence of messages $\hat{j}_{1m}^1, \hat{j}_{1m}^2, \ldots, \hat{j}_{1m}^T$ that lies in the set of typical $(x_1^+, x_2^+, y_m^{N(T+1)})$ sequences. With this approach it is shown in [20] that $\Pr\{J_1^t \neq J_{1m}^t\} \to 0$ for all t. A similar technique can be used by decoder ϕ_{2m} to show that $\Pr\{J_2^t \neq J_{2m}^t\} \to 0$. Then by the union bound, $P_m^t \to 0$ for all t. Finally, if the rate pair (R_1, R_2) lies in the intersection of the achievable rates for MACs W_1 and W_2, then $\max_m P_m^t \to 0$ and thus $P_e^t \to 0$ for all t by Lemma 1.

The converse for the MAC W_m is shown in [20] and [19] by using Fano's inequality, the data processing inequality, and the concavity of mutual information. Then $\max_m P_m^t \to 0$ implies that the rate pair (R_1, R_2) must lie within the intersection of the capacity regions of W_1 and W_2. $\square$

We now turn our attention to a random access system and apply Theorem 2 to determine the capacity region of the system. Each codeword symbol corresponds to a packet transmitted over the channel shown in Figure 2.1. We define the common input alphabet as $\mathcal{X} = \{0, 1, 2, \ldots, 2^b\}$, where $X_n \in \mathcal{X}$, for $n = 1, 2$. A channel input X_n can be either a packet of length b bits (an information-bearing symbol) or an idle symbol. The 0 symbol is the idle symbol and we let $\Pr\{X_n = 0\} = 1 - p_n$ according to the random access transmission probability. We assume a uniform distribution on the information-bearing codeword symbols, $\Pr\{X_n = x\} = p_n/2^b$, $x = 1, 2, \ldots 2^b$, meaning that a packet is equally likely to be any sequence of b bits. The channel output at receiver m is given by $Y_m = (Y_{1m}, Y_{2m}) \in \mathcal{X}' \times \mathcal{X}'$ where Y_{nm} denotes the packet from source n and $\mathcal{X}' = \mathcal{X} \cup \Delta$. The Δ symbol denotes a packet in error.

The introduction of the idle symbol 0 results in additional protocol or timing information being transmitted over the channel. The information content of this idle symbol is $h_b(p_n), n = 1, 2$, where h_b denotes the binary entropy function. The term $h_b(p_n)$ appears in the proof provided below and represents the protocol information that is studied by Gallager in [25]. Because we would like our capacity result to represent the rate of reliable communication of data packets, we will aim to exclude this timing information. We do so by considering capacity in packets/slot in the limit as $b \to \infty$, meaning that the data packets grow large and the fraction of timing information transmitted approaches 0. The timing information is excluded in previous work on random access capacity. In [17], prior to the start of transmission, a "protocol sequence" indicating the occurrence of idle slots is generated at the source and communicated to the receiver, effectively eliminating timing information. In [21], the capacity for $b \to \infty$ is presented. The capacity of the random access multicast system is given in the following Corollary to Theorem 2.

Corollary 3. *The capacity region of the random access system with two sources and two destinations is the closure of* (R_1, R_2) *for which*

$$\begin{aligned} R_1 &< \min_{m=1,2} \left\{ p_1(1-p_2)q_{1|1}^{(m)} + p_1 p_2 q_{1|1,2}^{(m)} \right\} \\ R_2 &< \min_{m=1,2} \left\{ (1-p_1)p_2 q_{2|2}^{(m)} + p_1 p_2 q_{2|1,2}^{(m)} \right\} \end{aligned}$$

for some $(p_1, p_2) \in [0, 1]^2$*. The rate* R_n *is expressed in packets/slot.*

Proof. The result follows by applying the assumptions about the input distribution and channel reception probabilities to the expressions given in Theorem 2. We first solve for $I(X_1; Y_m|X_2)$ by conditioning on X_2 to obtain

$$I(X_1; Y_m|X_2) = (1-p_2)I(X_1; Y_m|X_2 = 0) + p_2 I(X_1; Y_m|X_2 \neq 0). \tag{2.9}$$

An expression for $I(X_1; Y_m|X_2 = 0)$ can be found from the following sequence of equalities.

$$\begin{aligned}
I(X_1; Y_m|X_2 = 0) &= H(X_1|X_2 = 0) - H(X_1|Y_m, X_2 = 0) \\
&\overset{(a)}{=} H(X_1) - \Pr(Y_{1m} = \Delta|X_2 = 0)\log_2 2^b \\
&= -(1-p_1)\log_2(1-p_1) - p_1\log_2(p_1/2^b) - p_1(1-q_{1|1}^{(m)})b \\
&= h_b(p_1) + p_1\log_2 2^b - p_1 b + p_1 q_{1|1}^{(m)} b \\
&= h_b(p_1) + p_1 q_{1|1}^{(m)} b
\end{aligned} \tag{2.10}$$

where (a) holds since X_2 is independent of X_1 and since $H(X_1|Y_{1m} \neq \Delta, X_2 = 0) = 0$. For $(X_1; Y_m|X_2 \neq 0)$ we have

$$\begin{aligned}
I(X_1; Y_m|X_2 \neq 0) &= H(X_1|X_2 \neq 0) - H(X_1|Y_m, X_2 \neq 0) \\
&= H(X_1) - \Pr(Y_{1m} = \Delta|X_2 \neq 0)\log_2 2^b \\
&= -(1-p_1)\log_2(1-p_1) - p_1\log_2(p_1/2^b) - p_1(1-q_{1|1,2}^{(m)})b \\
&= h_b(p_1) + p_1\log_2 2^b - p_1 b + p_1 q_{1|1,2}^{(m)} b \\
&= h_b(p_1) + p_1 q_{1|1,2}^{(m)} b.
\end{aligned} \tag{2.11}$$

Combining expressions (2.10) and (2.11) results in

$$I(X_1; Y_m|X_2) = h_b(p_1) + bp_1(1-p_2)q_{1|1}^{(m)} + bp_1p_2q_{1|1,2}^{(m)} \text{ bits/transmission.} \tag{2.12}$$

Since one packet corresponds to b bits, we divide by b to obtain a result in units of packets per slot. We then let $b \to \infty$ to obtain

$$I(X_1; Y_m|X_2) = p_1(1-p_2)q_{1|1}^{(m)} + p_1 p_2 q_{1|1,2}^{(m)} \text{ packets/slot.} \tag{2.13}$$

By following a similar approach, we can show that $I(X_2; Y_m|X_1)$ is given as follows.

$$I(X_2; Y_m|X_1) = (1-p_1)p_2 q_{2|2}^{(m)} + p_1 p_2 q_{2|1,2}^{(m)} \text{ packets/slot} \tag{2.14}$$

The bound on the sum rate can be found by breaking up $I(X_1, X_2; Y_m)$ into four terms.

$$\begin{aligned} I(X_1, X_2; Y_m) &= H(X_1, X_2) - H(X_1, X_2|Y_m) \\ &= H(X_1|X_2) + H(X_2) - H(X_1|Y_m, X_2) - H(X_2|Y_m) \\ &= H(X_1) + H(X_2) - H(X_1|Y_{1m}, X_2) - H(X_2|Y_{1m}, Y_{2m}) \end{aligned} \tag{2.15}$$

For $n = 1, 2$, the terms $H(X_n)$ can be expressed as

$$\begin{aligned} H(X_n) &= -(1-p_n)\log_2(1-p_n) - p_n \log_2(p_n/2^b) \\ &= h_b(p_n) + p_n \log_2 2^b \\ &= h_b(p_n) + p_n b. \end{aligned} \tag{2.16}$$

The last two terms in (2.15) can be found in the following manner.

$$\begin{aligned}
H(X_1|Y_{1m},X_2) &= H(X_1|Y_{1m}=\Delta,X_2)\Pr(Y_{1m}=\Delta|X_2)\\
&= (1-p_2)H(X_1|Y_{1m}=\Delta,X_2=0)\Pr(Y_{1m}=\Delta|X_2=0)\\
&\quad +p_2H(X_1|Y_{1m}=\Delta,X_2\neq 0)\Pr(Y_{1m}=\Delta|X_2\neq 0)\\
&= (1-p_2)\log_2 2^b\Pr(Y_{1m}=\Delta|X_2=0)+p_2\log_2 2^b\Pr(Y_{1m}=\Delta|X_2\neq 0)\\
&= b(1-p_2)p_1(1-q_{1|1}^{(m)})+bp_2p_1(1-q_{1|1,2}^{(m)})\\
&= bp_1-bp_1(1-p_2)q_{1|1}^{(m)}-bp_1p_2q_{1|1,2}^{(m)} \qquad (2.17)
\end{aligned}$$

$$\begin{aligned}
H(X_2|Y_{1m},Y_{2m}) &= (1-p_1)H(X_2|Y_{1m}=0,Y_{2m})+p_1H(X_2|Y_{1m}\neq 0,Y_{2m})\\
&= (1-p_1)H(X_2|Y_{1m}=0,Y_{2m}=\Delta)\Pr(Y_{2m}=\Delta|Y_{1m}=0)\\
&\quad +p_1H(X_2|Y_{1m}\neq 0,Y_{2m}=\Delta)\Pr(Y_{2m}=\Delta|Y_{1m}\neq 0)\\
&= (1-p_1)\log_2 2^bp_2(1-q_{2|2}^{(m)})+p_1\log_2 2^bp_2(1-q_{2|1,2}^{(m)})\\
&= bp_2-b(1-p_1)p_2q_{2|2}^{(m)}-bp_1p_2q_{2|1,2}^{(m)} \qquad (2.18)
\end{aligned}$$

By substituting (2.16), (2.17), and (2.18) in (2.15), dividing by b and taking $b\to\infty$, we obtain

$$I(X_1,X_2;Y_m)=p_1(1-p_2)q_{1|1}^{(m)}+p_1p_2q_{1|1,2}^{(m)}+(1-p_1)p_2q_{2|2}^{(m)}+p_1p_2q_{2|1,2}^{(m)}\text{ packets/slot.} \qquad (2.19)$$

Since $I(X_1,X_2;Y_m)=I(X_1;Y_m|X_2)+I(X_2;Y_m|X_1)$, the bound on the sum rate is superfluous in terms of describing the capacity region. The result in Corollary 3 follows. □

2.2.2 Throughput regions

In this section we treat the system shown in Fig. 2.1 as a network of queues and state the stable and saturated throughput regions of the system for two different transmission protocols: a retransmission protocol and random linear coding. We first describe the derivation of the stable and saturated throughput regions for random access, and the relation between them. This formulation is a generalization of the formulation provided in previous work on unicast transmission.

The model we consider is as follows. Packets arrive to source n according to a Bernoulli process with rate λ_n, $n = 1, 2$ packets per slot. The arrival process is independent from source to source and independent, identically distributed (i.i.d.) over slots. Packets that are not immediately transmitted are stored in an infinite buffer maintained at each source. Transmissions occur according to the random access protocol with source n transmitting with probability p_n when it is backlogged. If a source is empty in a given slot, it does not access the channel. Each source-destination pair is assumed to maintain an orthogonal feedback channel so that instantaneous and error-free acknowledgements can be sent from the destinations to the sources.

Let $Q_n(k)$ denote the length of the queue at the n^{th} source node at the beginning of the k^{th} slot. We refer to the system shown in Fig. 2.1 as the original system and denote it $\mathcal{S}$. The evolution of the queue in $\mathcal{S}$ is expressed as follows.

$$Q_n(k+1) = (Q_n(k) - B_n(k))^+ + A_n(k), \tag{2.20}$$

where $x^+ = x$ if $x \geq 0$, and 0 otherwise. In the above equation, $A_n(k)$ denotes the arrivals to source n, where $E[A_n(k)] = \lambda_n, \forall k$ and $B_n(k)$ denotes completed services (or departures) from n. Thus $B_n(k)$ takes value 1 if a packet from source n completes service in slot k. We introduce the service rate $\mu_n \triangleq \lim_{k\to\infty} \Pr\{B_n(k)\}$ as the probability that a packet completes service in the steady-state.

The vector of queue lengths forms a two-dimensional, irreducible, aperiodic Markov chain $\mathbf{Q}(k) = (Q_1(k), Q_2(k))$. The system is stable if, for $\mathbf{x} \in \mathbb{N}^2$

$$\lim_{\mathbf{x}\to\infty} \lim_{k\to\infty} \Pr\{\mathbf{Q}(k) < \mathbf{x}\} = 1. \tag{2.21}$$

For our Markov chain $\mathbf{Q}(k)$, stability is equivalent to positive recurrence of the Markov chain. For a given transmission protocol, we define the *stable throughput region* of the system as the set of all arrival rates (λ_1, λ_2) for which there exists a set of transmission probabilities (p_1, p_2) such that the system is stable. A primary tool used in our work is Loynes' result [26], which tells us that if $A_n(k)$ and $B_n(k)$ are nonnegative, finite, and strictly stationary, then source n is stable if and only if $\lambda_n < \mu_n$.

The difficulty in finding the stable throughput region for our system $\mathcal{S}$ (and for any buffered random access system) arises from the interaction of the queues: the service rate μ_n varies depending on whether the *other* source is empty or backlogged and can create interference on the channel. To overcome this difficulty, the technique provided in [13] of introducing a *dominant* system can be used to decouple the sources. Let $\mathcal{S}^{[1]}$ denote a system which behaves exactly like system $\mathcal{S}$ except that both sources continue to transmit "dummy" packets when empty. The dummy packets do not affect the information-carrying ability of the source, but their transmission results in a decoupling of the queues. In the dominant system $\mathcal{S}^{[1]}$, all sources behave as if they are backlogged, the probability of interference from other sources is known according to the p_n values, and we can easily write down the service rates $\mu_n^{[1]}$. Let $Q_n^{[1]}$ denote the length of the queue at source n in system $\mathcal{S}^{[1]}$. It can be shown [12] that if $\mu_n^{[1]} \leq \mu_n$ then $\forall x \in \mathbb{N}$,

$$\Pr\{Q_n^{[1]} > x\} \geq \Pr\{Q_n > x\}.$$

In other words, the length of the queue in $\mathcal{S}^{[1]}$ is never shorter than in $\mathcal{S}$, or $Q_n^{[1]}$ stochastically dominates Q_n. If we can find the conditions for stability in $\mathcal{S}^{[1]}$, then stability in $\mathcal{S}$ is implied. Thus, stability in the dominant system is a sufficient condition for stability in the original system.

Let μ_{nb} denote the service rate at source n when the other source is backlogged and μ_{ne} the service rate when the other source is empty. Using the dominant systems approach, the stable throughput region for a system with two sources can be found exactly. This region is stated in the following theorem, which is a generalization of a result in [13]. Note that in the stable throughput region presented below, the service rates μ_{nb} and μ_{ne} are functions of p_n, $n = 1, 2$ (although not explicitly shown in these expressions).

Theorem 4. *[13] For a network with two sources and two destinations, the stable throughput region is given by the closure of $\mathcal{L}(p_1, p_2)$ where*

$$\mathcal{L}(p_1, p_2) = \bigcup_{i=1,2} \mathcal{L}_i(p_1, p_2) \tag{2.22}$$

and

$$\mathcal{L}_1(p_1,p_2) = \left\{ (\lambda_1,\lambda_2) : \begin{array}{rcl} \lambda_1 & < & \dfrac{\lambda_2}{\mu_{2b}}\mu_{1b} + \left(1 - \dfrac{\lambda_2}{\mu_{2b}}\right)\mu_{1e} \\ \lambda_2 & < & \mu_{2b} \end{array} \right\}$$

$$\mathcal{L}_2(p_1,p_2) = \left\{ (\lambda_1,\lambda_2) : \begin{array}{rcl} \lambda_1 & < & \mu_{1b} \\ \lambda_2 & < & \dfrac{\lambda_1}{\mu_{1b}}\mu_{2b} + \left(1 - \dfrac{\lambda_1}{\mu_{1b}}\right)\mu_{2e} \end{array} \right\}$$

for some $(p_1, p_2) \in [0,1]^2$.

In addition to the stable throughput region, we will be interested in the saturated throughput region of the random access system shown in Fig. 2.1. We define the *saturated throughput region* of the system as the stable throughput region under the assumption that all sources are always backlogged. Equivalently, the saturated throughput region is the closure of the service rates μ_{nb} for all p_n, $n = 1, 2$, where both sources are assumed to be backlogged. The saturated throughput region of the system $\mathcal{S}$ is equivalent to the stable throughput region of the dominant system $\mathcal{S}^{[1]}$. As we noted above, stability in the dominant system $\mathcal{S}^{[1]}$ implies stability in the original system $\mathcal{S}$, so the saturated throughput region of the system provides an inner bound to the stable throughput region. Previous work on buffered random access systems has shown that in all cases in which the stable throughput region has been found, it is known to coincide with the saturated throughput region. (Note that for fixed (p_1, p_2), the stable throughput and saturated throughput regions are not equivalent; it is when we take the closure over (p_1, p_2) that the two become equivalent.) This result holds in general (and particularly for our multicast problem), as stated in the following theorem.

Theorem 5. *For a random access system with two source nodes the stable throughput region given in Theorem 4 is identical to the saturated throughput region, which is given by the closure of* (λ_1, λ_2) *for which*

$$\lambda_1 < \mu_{1b}, \quad \lambda_2 < \mu_{2b} \tag{2.23}$$

for some $(p_1, p_2) \in [0, 1]^2$.

Proof. We show that the boundaries of the stable throughput and saturated throughput regions, given by the solutions of two constrained optimization problems, are identical. We express the backlogged service rates μ_{nb} as follows.

$$\mu_{1b} = p_1 f_1(p_2), \quad \mu_{2b} = p_2 f_2(p_1). \tag{2.24}$$

The service rate μ_{nb} is given by p_n, the probability that source n transmits, times a function $f_n(\cdot)$ of the interference by the other user. The function $f_1(p_2)$ is functionally independent of p_1 and decreasing in p_2, meaning that interference by source 2 decreases the service rate of source 1. Accordingly, $f_1(p_2) \leq f_1(0)$. Similarly, $f_2(p_1)$ is independent of p_2 and decreasing in p_1. The empty service rates can then be expressed as $\mu_{1e} = p_1 f_1(0)$ and $\mu_{2e} = p_2 f_2(0)$.

We replace λ_1 by x and λ_2 by y. With this notation in place, the *boundary* of the region given in Theorem 4 for fixed (p_1, p_2) can be written as follows.

$$x = p_1 f_1(0) - \frac{y p_1 \left(f_1(0) - f_1(p_2)\right)}{p_2 f_2(p_1)}, \quad 0 \leq y \leq p_2 f_2(p_1) \tag{2.25}$$

$$y = p_2 f_2(0) - \frac{x p_2 \left(f_2(0) - f_2(p_1)\right)}{p_1 f_1(p_2)}, \quad 0 \leq x \leq p_1 f_1(p_2) \tag{2.26}$$

To find the stable throughput region, we should maximize the expressions in (2.25) and (2.26) over (p_1, p_2) and take the intersection of the regions bounded by the resulting curves. We note that this is not a standard optimization problem because the objective function is piece-wise and non-differentiable at a point in its domain. An example of an analytical solution to this optimization problem is given in [16] for the single destination case.

The boundary of the saturated throughput region for fixed (p_1, p_2) can be written as follows.

$$x = p_1 f_1(p_2), \quad 0 \leq y \leq p_2 f_2(p_1) \tag{2.27}$$

$$y = p_2 f_2(p_1), \quad 0 \leq x \leq p_1 f_1(p_2) \tag{2.28}$$

The saturated throughput region is found by maximizing the expressions in (2.27) and (2.28) and taking the intersection of the resulting regions. Consider Eqn. (2.28) in which we wish to maximize y over (p_1, p_2). Note that the constraint $x \leq p_1 f_1(p_2)$ is a lower bound on p_1 over which we perform the maximization. Since $f_2(p_1)$ is decreasing in p_1, the lower bound $p_1 \geq x/f_1(p_2)$ provides an upper bound on y, and this upper bound can be achieved when maximizing (2.28) over (p_1, p_2). Then maximization of y in (2.28) is equivalent to maximizing y as follows.

$$y = p_2 f_2(p_1)\Big|_{p_1 = \frac{x}{f_1(p_2)}}, \quad 0 \leq x \leq p_1 f_1(p_2) \tag{2.29}$$

To see that this is identical to (2.26), we write $f_2(p_1)$ as follows.

$$\begin{aligned} f_2(p_1) &= f_2(0) - (f_2(0) - f_2(p_1)) &(2.30)\\ &= f_2(0) - \frac{p_1 \left(f_2(0) - f_2(p_1)\right)}{p_1} &(2.31) \end{aligned}$$

Then

$$p_2 f_2(p_1)\Big|_{p_1 = \frac{x}{f_1(p_2)}} = p_2 f_2(0) - \frac{x p_2 \left(f_2(0) - f_2(p_1)\right)}{p_1 f_1(p_2)} \tag{2.32}$$

and the maximum y in (2.28) is identical to the maximum y in (2.26). Similarly, the maximum of x in (2.27) is identical to the maximum of x in (2.25). $\square$

We next derive the backlogged and empty service rates μ_{nb} and μ_{ne} for two different transmission schemes. Together with Theorems 4 and 5, this provides us a complete characterization of the stable and saturated throughput regions.

2.2.2.1 Retransmissions

In this section we describe the service rates assuming that a retransmission protocol is used to ensure reliable communication. In the retransmission scheme, as long as source n has not received feedback acknowledgements from both destinations $m = 1, 2$, it will continue to transmit the packet over the channel with probability p_n in each slot. As soon as the source has received acknowledgements from both destinations, it will remove the packet from its queue and begin transmitting the next packet waiting in its buffer, if any. Let random variable T_n denote the service time for source n, given by the total number of slots that transpire before the packet from source n is successfully received at both destinations. (Note that the service time includes slots during which the source does not transmit, which happens with probability $1 - p_n$). Since each completed service in the retransmission scheme results in 1 packet being removed from the queue, the average service rate, which we denote μ_n^R, is given by $\mu_n^R = 1/E[T_n]$.

We first find the backlogged service rates μ_{nb}^R. Let $T_n^{(m)}$ denote the number of slots needed for successful reception of a packet from source n at destination m,

$m = 1, 2$. The $T_n^{(m)}$ are geometrically distributed according to the transmission probabilities p_n and reception probabilities $q_{n|n}^{(m)}$, $q_{n|1,2}^{(m)}$. We introduce the following notation. Let ϕ_n denote the probability of successful reception of the packet from source n at destination 1 given that source n transmits and that both sources are backlogged. Similarly, σ_n denotes the probability of successful reception at destination 2 given that both sources are backlogged and that source n transmits. For instance, ϕ_1 and σ_1 are given by

$$\phi_1 = \overline{p}_2 q_{1|1}^{(1)} + p_2 q_{1|1,2}^{(1)} \tag{2.33}$$

$$\sigma_1 = \overline{p}_2 q_{1|1}^{(2)} + p_2 q_{1|1,2}^{(2)} \tag{2.34}$$

where $\overline{p}_n = 1 - p_n$. When source 2 is backlogged, $T_1^{(1)}$ is geometrically distributed with parameter $p_1\phi_1$ and $T_1^{(2)}$ is geometrically distributed with parameter $p_1\sigma_1$. The total service time for source 1 when source 2 is backlogged will be given by the maximum of the service times to each destination,

$$T_1 \sim \max_m T_1^{(m)}. \tag{2.35}$$

Similarly, when source 1 is backlogged, the service time for source 2 is given by

$$T_2 \sim \max_m T_2^{(m)}, \tag{2.36}$$

where

$$T_2^{(1)} \sim \text{geom}\left(p_2\phi_2\right), \quad T_2^{(2)} \sim \text{geom}\left(p_2\sigma_2\right). \tag{2.37}$$

The expected maximum value $E[T_n]$ can be readily found and (after some algebra) the backlogged service rates are given by

$$\mu_{nb}^R = \frac{p_n\phi_n\sigma_n(\phi_n + \sigma_n - \tau_n)}{(\phi_n + \sigma_n)(\phi_n + \sigma_n - \tau_n) - \phi_n\sigma_n} \tag{2.38}$$

where τ_n denotes the probability that a packet sent from n is received at both destinations given that source n transmits, e.g., $\tau_1 = \overline{p}_2 q^{(1)}_{1|1} q^{(2)}_{1|1} + p_2 q^{(1)}_{1|1,2} q^{(2)}_{1|1,2}$. The empty service rates μ^R_{ne} can be found directly from μ^R_{nb} as

$$\mu^R_{1e} = \mu^R_{1b}|_{p_2=0}, \quad \mu^R_{2e} = \mu^R_{2b}|_{p_1=0}. \tag{2.39}$$

We now compare the throughput region for the retransmissions scheme to the Shannon capacity region. We consider the backlogged service rate μ^R_{nb}, since, by Theorem 5, this is the term that bounds both the stable throughput and saturated throughput regions for source n. The expected service time for source 1 when source 2 is backlogged is bounded as

$$\begin{aligned} E[T_1] = E[\max_m T^{(m)}_1] &\overset{(a)}{\geq} \max_m E[T^{(m)}_1] &(2.40)\\ &\overset{(b)}{=} \max_m \left\{ \frac{1}{p_1(1-p_2)q^{(m)}_{1|1} + p_1 p_2 q^{(m)}_{1|1,2}} \right\} &(2.41)\\ &= \frac{1}{\min_m \left\{ p_1(1-p_2)q^{(m)}_{1|1} + p_1 p_2 q^{(m)}_{1|1,2} \right\}} &(2.42) \end{aligned}$$

where (a) follows from Jensen's inequality and (b) follows from the expected value of a geometrically distributed random variable. Then the backlogged service rate μ^R_{1b} is bounded as

$$\mu^R_{1b} = \frac{1}{E[T_1]} \leq \min_m \left\{ p_1(1-p_2)q^{(m)}_{1|1} + p_1 p_2 q^{(m)}_{1|1,2} \right\} \tag{2.43}$$

and μ^R_{2b} can be bounded similarly. Note that the right-hand side of (2.43) is equal to the bound on the Shannon achievable rate R_1 given in Corollary 3. Thus the Shannon capacity region outer bounds the throughput region for the retransmission scheme.

2.2.2.2 Random linear coding

In this section we present two different approaches to analyzing the throughput region for a transmission scheme in which groups of K packets at the front of the queue are randomly encoded and transmitted over the channel. By encoded, we mean that a random linear combination of the K packets is formed, and we refer to this random linear combination as a coded packet. As described in Chapter 1, a random linear combination is formed by randomly and uniformly generating a weight a_i from a u-ary alphabet for each of the K packets and computing the weighted sum in the finite field $\mathbb{F}_u$. A new coded packet is formed for each time slot in which the source transmits. As soon as a destination has received enough coded packets and is able to decode the original K packets, it does so and sends an acknowledgement to the source over its feedback channel. Once the source receives acknowledgements from both destinations, it removes the K packets it has been encoding and transmitting from its queue and begins encoding and transmission of the next K packets waiting in its buffer. The stable throughput region for a similar system with $K = 2$ is found in [27].

We first analyze the random linear coding scheme by examining the expected service time. Let $\widetilde{T}_n$ denote the service time for source n, which is the number of slots that elapse from the transmission of the first coded packet until the source has received acknowledgements from both destinations. Since each completed service in the random linear coding scheme results in K packets being removed from the queue, the average service rate is given by $\mu_n^C = K/E[\widetilde{T}_n]$. The service time will be a random variable dependent upon the random access transmission probabilities p_n, the reception probabilities $q_{n|n}^{(m)}$ and $q_{n|1,2}^{(m)}$, and the number of coded packets needed to decode. Let $N_n^{(m)}$ denote the number of coded packets received from source n at destination m such that m can decode the original K packets. The $N_n^{(m)}$ will be identically distributed over n and m since all coded packets are generated using the same distribution. Additionally, $N_n^{(m)}$ will be independent over n since the two sources generate their coded packets independently. However, $N_n^{(m)}$ will be correlated over m, since a given source n will generate and transmit the same coded packets to both destinations $m = 1, 2$.

Let $F_K(j)$ denote the common cumulative distribution function (cdf) of $N_n^{(m)}$, or the probability that the number of coded packets needed for decoding is less than or equal to j. Similarly, let $f_K(j)$ denote the probability mass function (pmf) of $N_n^{(m)}$, or the probability that decoding can be performed when the destination has received exactly j columns and no fewer. As described in Chapter 1, the distribution of $N_n^{(m)}$ is given by the probability that a random u-ary matrix is full-rank. We have

$$F_K(j) = \begin{cases} \prod_{i=0}^{K-1}(1 - u^{-j+i}) & j \geq K \\ 0 & j < K \end{cases} \tag{2.44}$$

and

$$f_K(j) = F_K(j) - F_K(j-1). \tag{2.45}$$

From Eqns. 4.6 and 4.7 the expected value of $N_n^{(m)}$ can be computed, where

$E[N_n^{(m)}] \geq K$. The ratio

$$\begin{aligned}\frac{E[N_n^{(m)}]}{K} &= \prod_{i=0}^{K-1}(1-u^{-K+i}) + \frac{1}{K}\sum_{j=K+1}^{\infty} j\left(\prod_{i=0}^{K-1}(1-u^{-j+i}) - \prod_{i=0}^{K-1}(1-u^{-j+1+i})\right) \\ &\searrow 1 \quad \text{as } K \to \infty \end{aligned} \tag{2.46}$$

for all $n, m = 1, 2$.

With the distribution of $N_n^{(m)}$ characterized, we can now describe the service time for random linear coding from source n to destination m, which we denote $\widetilde{T}_n^{(m)}$. The number of slots needed for the successful reception of each coded packet will be geometrically distributed and in total $N_n^{(m)}$ coded packets must be received. Then $\widetilde{T}_n^{(m)}$ can be modeled as the sum of $N_n^{(m)}$ independent geometrically distributed random variables.

$$\widetilde{T}_n^{(m)} = g_{n,1}^{(m)} + g_{n,2}^{(m)} + \ldots + g_{n,N_n^{(m)}}^{(m)} \tag{2.47}$$

In the above expression, $g_{n,i}^{(m)}$, $i = 1, 2, \ldots, N_n^{(m)}$, will be geometrically distributed with a parameter that depends on the reception probabilities and on the assumption of whether the other source is backlogged. For instance, when source 2 is backlogged

$$g_{1,i}^{(m)} \sim \text{geom}\left(p_1(1-p_2)q_{1|1}^{(m)} + p_1 p_2 q_{1|1,2}^{(m)}\right), \quad i = 1, 2, \ldots, N_1^{(m)} \tag{2.48}$$

and when source 2 is empty,

$$g_{1,i}^{(m)} \sim \text{geom}\left(p_1 q_{1|1}^{(m)}\right), \quad i = 1, 2, \ldots, N_1^{(m)}. \tag{2.49}$$

The total service time is given as

$$\widetilde{T}_n = \max_m \widetilde{T}_n^{(m)}. \tag{2.50}$$

As in the case of random access with retransmissions, we argue that the throughput region for random access with random linear coding will be outer bounded by the Shannon capacity region. In the case that source 2 is backlogged, the expected service time for source 1 is now bounded as

$$
\begin{aligned}
E[\widetilde{T}_1] = E[\max_m \widetilde{T}_1^{(m)}] \quad &\overset{(a)}{\geq} \quad \max_m E[\widetilde{T}_1^{(m)}] &(2.51)\\
&\overset{(b)}{=} \quad \max_m \frac{E[N_1^{(m)}]}{p_1(1-p_2)q_{1|1}^{(m)} + p_1 p_2 q_{1|1,2}^{(m)}} &(2.52)\\
&\overset{(c)}{=} \quad \frac{E[N_1^{(m)}]}{\min_m \left\{p_1(1-p_2)q_{1|1}^{(m)} + p_1 p_2 q_{1|1,2}^{(m)}\right\}} &(2.53)
\end{aligned}
$$

where (a) again follows from Jensen's inequality, (b) holds since $g_{1,i}^{(m)}$ are independent, identically distributed, and (c) holds since $N_1^{(m)}$ is identically distributed over m, meaning that $E[N_1^{(1)}] = E[N_1^{(2)}]$. The backlogged service rate μ_{1b}^C is bounded as

$$
\begin{aligned}
\mu_{1b}^C = \frac{K}{E[\widetilde{T}_1]} \quad &\leq \quad \frac{K}{E[N_1^{(m)}]} \min_m \left\{p_1(1-p_2)q_{1|1}^{(m)} + p_1 p_2 q_{1|1,2}^{(m)}\right\} &(2.54)\\
&\leq \quad \min_m \left\{p_1(1-p_2)q_{1|1}^{(m)} + p_1 p_2 q_{1|1,2}^{(m)}\right\}. &(2.55)
\end{aligned}
$$

Then the Shannon capacity region outer bounds the stable throughput region for random access with random linear coding.

Unfortunately a difficulty arises in finding the service rates μ_{nb}^C and μ_{ne}^C in closed form from $E[\max_m \widetilde{T}_n^{(m)}]$. This difficulty arises for a number of reasons: $\widetilde{T}_n^{(1)}$ and $\widetilde{T}_n^{(2)}$ are not independent, and $\widetilde{T}_n^{(m)}$ is distributed according to a composite distribution function, for which the pdf is not easily expressed in closed form. In fact, even if these two difficulties were to be removed, $E[\max_m \widetilde{T}_n^{(m)}]$ cannot be easily handled. For instance, let us assume that $\widetilde{T}_n^{(1)}$ and $\widetilde{T}_n^{(2)}$ are independent and that $N_n^{(m)} = n^{(m)}$ are deterministic (which means that the pdf is no longer composite).

In that case, $\widetilde{T}_n^{(m)}$ is the sum of $n^{(m)}$ iid geometric random variables, meaning that $\widetilde{T}_n^{(m)}$ follows a negative binomial distribution. Let us further make the assumption that $q_{n|n}^{(1)} = q_{n|n}^{(2)}$ and $q_{n|1,2}^{(1)} = q_{n|1,2}^{(2)}$, which means that $\widetilde{T}_n^{(m)}$ are identically distributed over m. In this very simplified case, $E[\max_m \widetilde{T}_n^{(m)}]$ is the expected maximum of two iid negative binomial random variables. The computation of this expected value is treated in [28], and the result involves a periodic function which is approximated by a Fourier series. Thus, even in this very simplified case, we can at best approximate $E[\max_m \widetilde{T}_n^{(m)}]$, and this approximation must be computed numerically.

As an alternative to the analysis presented above, we now develop a Markov chain model which allows us to find the queueing service rates. For a given source node, we set up a vector Markov chain with state (i, j, k) corresponding to the number of linearly independent coded packets that have been received from the source node. In this model, i represents the number of linearly independent coded packets that have been received at destination 1, and j represents the number of linearly independent packets that have been received at destination 2. Since the coded packets are generated by the same source, the two destinations may have received some packets in common; the number of those packets which are received in common and are linearly independent to those received at both destinations is represented by k, where $k \leq \min(i, j)$. The variable k allows us to track the correlation between $N_n^{(1)}$ and $N_n^{(2)}$, which was a difficulty in our previous approach described above. The Markov chain evolves in discrete time over the time slots in our system model.

The state space of the Markov chain is the three-dimensional discrete set of points $[0, K]^3$. The states (K, K, k), $0 \leq k \leq K$ correspond to completion of the encoding and transmission process of the current group of K packets. From any state (K, K, k), $0 \leq k \leq K$, a transition back to the $(0, 0, 0)$ state occurs with probability one and the source can begin encoding and transmission for the next group of packets. With the exception of self-transitions and transitions into the $(0, 0, 0)$ state, transitions in the Markov chain can only occur "upward", corresponding to the reception of a new linearly independent packet, and a transition results in an increase of the indices i, j, k by at most 1, meaning that at most 1 new linearly independent packet can be received in a slot. We use the notation $(i_1, j_1, k_1) \rightarrow (i_2, j_2, k_2)$ to denote the transition from state (i_1, j_1, k_1) to state (i_2, j_2, k_2).

The Markov chain is irreducible and aperiodic, and because it has a finite state space, a stationary distribution exists. Let $\pi_{i,j,k}$ denote the steady-state probability of (i, j, k). The steady-state probabilities are found by solving the set of equations

$$\pi_{i_1,j_1,k_1} = \sum_{(i_2,j_2,k_2)} \pi_{i_2,j_2,k_2} \mathrm{Pr}((i_2, j_2, k_2) \rightarrow (i_1, j_1, k_1)) \tag{2.56}$$

and

$$\sum_{(i,j,k)} \pi_{i,j,k} = 1. \tag{2.57}$$

The service rate $\widetilde{\mu}_n$ is equal to K times the probability of transitioning into state (K, K, k), $0 \leq k \leq K$, from a non-zero state, thus completing transmission to both destinations. There are only a few ways to transition into (K, K, k) from a non-zero state. Let $\mathcal{A}_k$, $0 \leq k \leq K$ denote the set of states that are one step away from

(K, K, k). For $k \in [0, K-1]$ we have

$$\mathcal{A}_k = \{(K-1, K, k), (K-1, K, k-1), (K, K-1, k), (K, K-1, k-1), (K-1, K-1, k-1)\}, \tag{2.58}$$

and for $k = K$,

$$\mathcal{A}_K = \{(K-1, K, K-1), (K, K-1, K-1), (K-1, K-1, K-1)\}. \tag{2.59}$$

Note that we define $\mathcal{A}_K$ in this way since the states $(K-1, K, K)$ and $(K, K-1, K)$ violate $k \leq \min(i, j)$. The service rate for source n is given by

$$\mu_n^C = K \sum_{k=0}^{K} \sum_{(i,j,k) \in \mathcal{A}_k} \pi_{i,j,k} \Pr((i, j, k) \to (K, K, k)). \tag{2.60}$$

The transition probabilities $(i_1, j_1, k_1) \to (i_2, j_2, k_2)$ for source n can be written assuming that the other source is either backlogged or empty, leading to the service rates μ_{nb}^C and μ_{ne}^C.

As an example, consider the transition $(i, j, k) \to (i+1, j, k)$ in the Markov chain for source 1 when source 2 is backlogged. Assume first that source two does not transmit, which happens with probability $1 - p_2$. Then there are two ways for the transition $(i, j, k) \to (i+1, j, k)$ to occur. First, destination 2 could receive no packet, which happens with probability $1 - q_{1|1}^{(2)}$, while destination 1 receives a coded packet which is neither an all-zero packet nor equal to any linear combination of the i packets it has already received, which happens with probability $q_{1|1}^{(1)}(1 - u^i u^{-K})$. Alternatively, both destinations could receive a coded packet, but that packet is either the all zero packet or some linear combination of the packets that have been received by destination 2 and *not* by destination 1. This happens with probability $q_{1|1}^{(1)}q_{1|1}^{(2)}(u^j - u^k)u^{-K}$. The same two alternatives are possible in the case that source 2 does transmit, which happens with probability p_2, except that the reception probabilities are now given by $q_{1|1,2}^{(m)}$. Then the transition $(i, j, k) \to (i+1, j, k)$ for source 1 when source 2 is backlogged occurs with probability

$$p_1 \left[\overline{p}_2 \left\{ q_{1|1}^{(1)}(1 - q_{1|1}^{(2)})(1 - u^i u^{-K}) + q_{1|1}^{(1)}q_{1|1}^{(2)}(u^j - u^k)u^{-K} \right\} \right.$$
$$\left. + p_2 \left\{ q_{1|1,2}^{(1)}(1 - q_{1|1,2}^{(2)})(1 - u^i u^{-K}) + q_{1|1,2}^{(1)}q_{1|1,2}^{(2)}(u^j - u^k)u^{-K} \right\} \right]. \quad (2.61)$$

The same transition probability can be used when source 2 is empty by setting $p_2 = 0$. Similar arguments can be used to find all transition probabilities for our Markov chain model; we have stated those probabilities in Appendix A. Ultimately, we would like to find closed-form expressions for the service rates μ_{nb}^C and μ_{ne}^C, but due to the size of the state-space, this is a difficult task. Instead we have computed some numerical examples based on the Markov chain model presented above, and those are presented next.

2.2.3 Numerical examples

We have computed numerical examples of the Shannon capacity region and the throughput regions for retransmissions and random linear coding. The results for random linear coding have been computed with the service rates given by Eqn. 2.60. Fig. 2.2 shows results for a "good" channel with relatively large reception probabilities while Fig. 2.3 shows results for a "poor" channel with smaller reception probabilities. In both figures, we have plotted the throughput for random linear coding (labeled "RLC") with various values of K. The results show that the

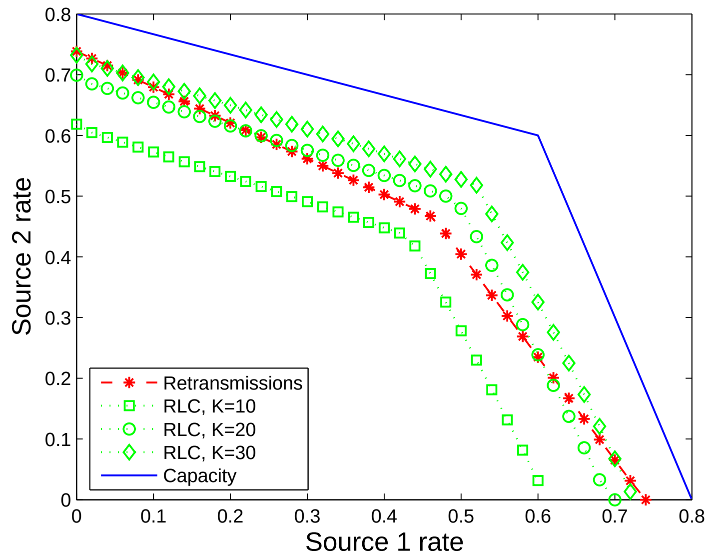

Figure 2.2: Closure of the throughput and capacity regions for a channel with reception probabilities $q^{(1)}_{1|1} = q^{(2)}_{2|2} = 0.8$, $q^{(2)}_{1|1} = q^{(1)}_{2|2} = 0.7$, and $q^{(m)}_{1|1,2} = q^{(m)}_{2|1,2} = 0.6, m = 1, 2$. Random linear coding is carried out over binary symbols ($u = 2$) and is labeled RLC.

Shannon capacity region is strictly larger than the throughput regions for both the retransmissions and random linear coding schemes. Additionally, the throughput region for random linear coding grows with increasing values of K.

The random linear coding scheme does not necessarily outperform the retransmission scheme. For small values of K, the coding scheme is inefficient in the sense that the ratio $K/E[N^{(m)}_n]$ is small. This inefficiency is largely due to the fact that an all-zero coded packet can be generated and transmitted; this occurs more often for small values of K. As the channel improves, the retransmission scheme performs better relative to random linear coding, since for a "good" channel, packets are more often received correctly and do not need to be retransmitted.

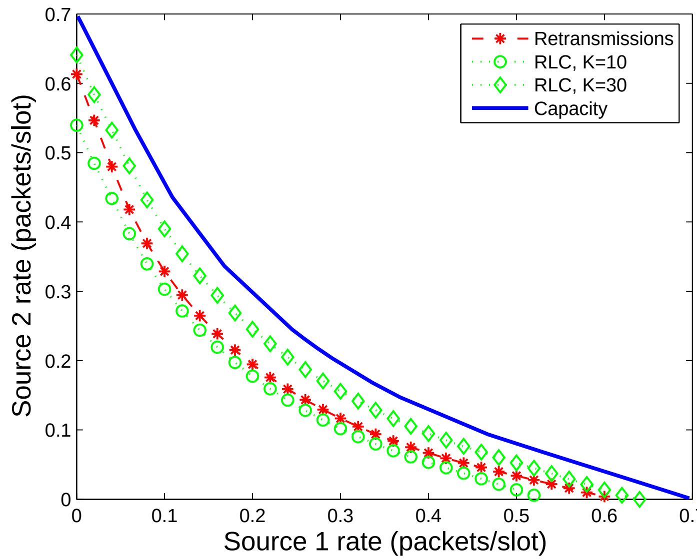

Figure 2.3: Closure of the throughput and capacity regions for a channel with reception probabilities $q^{(1)}_{1|1} = q^{(2)}_{2|2} = 0.8$, $q^{(2)}_{1|1} = q^{(1)}_{2|2} = 0.7$, and $q^{(m)}_{1|1,2} = q^{(m)}_{2|1,2} = 0.2, m = 1, 2$. Random linear coding is carried out over binary symbols ($u = 2$) and is labeled RLC.

2.3 A network of N sources and M destinations

We now consider the multicast system shown in Fig. 2.4, where N source nodes, $s_1, s_2, \ldots, s_N$ multicast to M destination nodes. Once again, we assume that packets arrive to s_n according to a Bernoulli process with rate λ_n, $n = 1, 2, \ldots, N$ packets per slot. The arrival process is independent from source to source and independent, identically distributed over slots. Packets that are not immediately transmitted are stored in an infinite buffer maintained at each source. All source nodes compete in a random fashion for access to the channel in order to transmit a packet of information to *all* M destination nodes. When source n has a packet to transmit, it does so with probability p_n in the first available slot. Each packet is intended for all M destinations. We assume that instantaneous and error-free acknowledge-

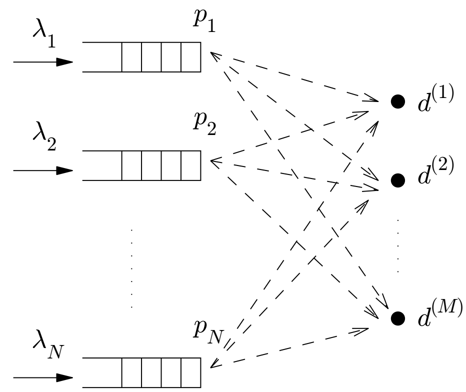

Figure 2.4: N source nodes multicast to M destination nodes.

ments (ACKs) are sent from the destinations and that each source-destination pair has a dedicated channel for ACKs. We focus now on the retransmission scheme. If the source has not yet received an ACK from all M destinations, the packet is retransmitted. We know from the analysis of two sources and two destinations that retransmissions can be inferior to random linear coding. However, our aim here is to understand the relationship between the stable throughput and the saturated throughput for *multicast* transmission, and in doing so, we focus on one particular transmission scheme.

The channel model is now a simplified version of the model presented for the $N = 2$, $M = 2$ scenario. We assume that whenever two or more sources transmit simultaneously, none of the transmissions are successful. Additionally, we assume that the channel reception probabilities from a source are the same for all destinations, $q^{(1)}_{n|n} = q^{(2)}_{n|n} = \ldots = q^{(M)}_{n|n} = q_{n|n}$, $n = 1 \ldots N$. We refer to the destinations as being *indistinguishable* in this channel model.

The definitions given for two sources and two destinations can be generalized to this scenario. Once again, $Q_n(k)$ denotes the length of the n^{th} queue at the beginning of time slot k, $A_n(k)$ denotes the number of arrivals at s_n (which can be at most one), and $B_n(k)$ denotes the number of departures. The service rate of s_n is denoted μ_n. The vector of queue lengths forms an N-dimensional Markov chain $\mathbf{Q}(k) = (Q_1(k), Q_2(k), \ldots, Q_N(k))$ and the system is stable if the condition in Eqn. 2.21 holds. The stable throughput region is the set of all arrival rate vectors $\boldsymbol{\lambda_N} = (\lambda_1, \lambda_2, \ldots, \lambda_N)$ for which there exists a set of transmission probabilities $\mathbf{p_N} = (p_1, p_2, \ldots, p_N)$ such that the system is stable. We again make use of the dominant systems approach and refer to the original system, in which none of the sources transmit when they become empty, as $\mathcal{S}$. The backlogged service rate (i.e., the service rate when *all* other sources are backlogged) is denoted μ_{nb} and the empty service rate (i.e., the service rate when *all* other sources are empty) is denoted μ_{ne}.

The saturated throughput region of the network of arbitrary size can be determined exactly by taking the closure over $\mathbf{p_N} \in [0, 1]^N$ of the the backlogged service rates μ_{nb}. As in the case of unicast transmission, the stable throughput region has not been determined exactly, but instead, we develop inner and outer bounds. Before doing so, we characterize the service rates for the N-source M-destination network.

2.3.1 Service rates for retransmissions

We define the receiver-state variable r_n for each source $n = 1, 2, \ldots, N$ as the number of destinations that have received the packet that source n is attempting to transmit, $r_n \in [0, 1, \ldots, M-1]$. We do not allow r_n to take value M since as soon

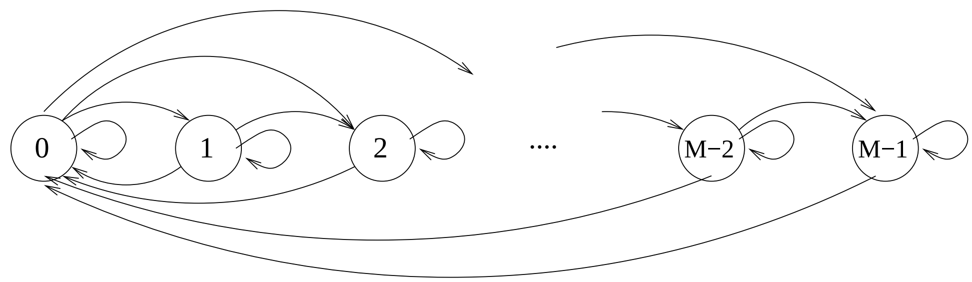

Figure 2.5: Receiver state Markov chain for M indistinguishable destinations.

as all destinations have received the packet, the source will instantaneously revert back to $r_n = 0$ and either begin serving the next packet in the queue of source n or become idle if the source is empty. We define the set $\mathcal{B}$ as the set of all sources that are backlogged at the time source n is attempting to transmit the packet at the front of its queue. Let β_n denote that probability that source n accesses the channel without interference, i.e.,

$$\beta_n = p_n \left(\prod_{l \in \mathcal{B} \setminus n} \overline{p_l} \right). \tag{2.62}$$

The service process conditioned on r_n can be described by

$$\Pr\{B_n = 1 | r_n = r\} = \beta_n {q_{n|n}}^{M-r}. \tag{2.63}$$

We develop a Markov chain model for r_n as shown in Fig. 2.5. In this model, transitions "upward" can occur between all pairs of states, however, transitions "downward" can only occur between a state and the 0 state. Additionally, each state has self-transitions. Let $\mathcal{P}$ denote the transition matrix for this Markov chain. When source n accesses the channel without collision, which happens with probability β_n, the transition probability matrix will be a matrix $\mathcal{P}^*$ which depends only on the reception probabilities $q_{n|n}$. Otherwise, a self-transition occurs, corresponding to a transition probability matrix equal to the identity matrix $\mathcal{I}$. Thus, we can describe

$\mathcal{P}$ as the convex combination of two probability matrices,

$$\mathcal{P} = \beta_n \mathcal{P}^* + (1 - \beta_n)\mathcal{I}. \tag{2.64}$$

Let $\boldsymbol{\pi}$ be the stationary distribution of $\mathcal{P}^*$, $\boldsymbol{\pi} = \boldsymbol{\pi}\mathcal{P}^*$. Clearly $\boldsymbol{\pi}$ will also be the stationary distribution of $\mathcal{P}$ since $\boldsymbol{\pi}\mathcal{P} = \beta_n \boldsymbol{\pi} + (1 - \beta_n)\boldsymbol{\pi}$. We can solve for $\boldsymbol{\pi}$ as follows. Let $p^*_{i,j}$ denote the probability of transition in r_n from i to j conditioned on s_n accessing the channel without collision. The transition probabilities are given as

$$p^*_{i,0} = \begin{cases} (1 - q_{n|n})^M + q^M_{n|n}, & i = 0 \\ q^{M-i}_{n|n}, & 0 < i < M. \end{cases} \tag{2.65}$$

$$p^*_{i,j} = \begin{cases} (1 - q_{n|n})^{M-i}, & i = j \\ \binom{M-i}{j-i} q^{j-i}_{n|n} (1 - q_{n|n})^{M-j}, & i < j \\ 0, & i > j. \end{cases} \tag{2.66}$$

In order to satisfy $\pi_i = \sum_j \pi_j p^*_{i,j}$ we have

$$\begin{aligned} \pi_i &= \sum_{j=0}^{i-1} \pi_j p^*_{i,j} + \pi_i p^*_{i,i} \quad i = 1, 2, \ldots, M-1 \\ &= \frac{\sum_{j=0}^{i-1} \pi_j p^*_{i,j}}{1 - p^*_{i,i}} \\ &= \frac{\sum_{k=1}^{i} \pi_{i-k} p^*_{i-k,i}}{1 - p^*_{i,i}}. \end{aligned} \tag{2.67}$$

Together with $\sum_{i=0}^{M-1} \pi_i = 1$, we can find the steady-state probabilities $\boldsymbol{\pi}$, which is the stationary distribution of $\mathcal{P}$.

Once the steady-state probabilities of the receiver Markov chain are found, the average service rate is given as follows.

$$\mu_n = \beta_n \sum_{i=0}^{M-1} \pi_i {q_{n|n}}^{M-i} \tag{2.68}$$

We further define

$$\alpha_n = \sum_{i=0}^{M-1} \pi_i {q_{n|n}}^{M-i}. \tag{2.69}$$

Thus the service rate can be represented in simplified form as

$$\mu_n = \beta_n \alpha_n. \tag{2.70}$$

This equation summarizes the natural relation between our multicast problem and the unicast collision channel problem [15]. The probability that source n completes transmission of a packet, given by μ_n, is equal to β_n, the probability that the source access the channel without collision, times α_n, which is the probability that all destinations receive the packet conditioned on collision-free access to the channel. In the unicast collision channel problem, we have $q_{n|n} = 1$ and $\alpha_n = 1$. Thus for $\alpha_n = 1$ we would expect the stable throughput region for our multicast problem to coincide with the results in [15] for the unicast collision channel. This is indeed the case as shown in the following section.

We define the empty and backlogged service rates as follows. When $\mathcal{B}$ contains all N sources, the broadcast service rate will take its minimum value μ_{nb} where

$$\mu_{nb} = p_n \left(\prod_{l \neq n} \overline{p_l} \right) \alpha_n. \tag{2.71}$$

The maximum value of μ_n is attained when only source n is backlogged and all other sources are empty. We denote this service rate by μ_{ne}

$$\mu_{ne} = p_n \alpha_n. \tag{2.72}$$

In deriving bounds on the stable throughput region, we will take advantage of the form of μ_n as expressed in 2.70 and the fact that $\mu_{nb} \leq \mu_n \leq \mu_{ne}$.

2.3.2 Bounds on the stable throughput

We follow the methodology outlined in [15] to develop bounds on the stable throughput region. These results are a generalization of the bounds on the stable throughput region for unicast given in [15]. To begin, by the dominant systems argument and Loynes' result, we can develop loose inner and outer bounds on the stable throughput region. First, if $\lambda_n < \mu_{nb}$, then since $\mu_{nb} \leq \mu_n$, queue n must be stable. Furthermore, if $\lambda_n < \mu_{nb}$ for all n, then the entire system is stable. Likewise, if $\lambda_n > \mu_{nb}$ for all n, then the system is unstable. This follows since $\lambda_n > \mu_{nb}$ corresponds to instability of all the queues in dominant system $\mathcal{S}^{[1]}$, in which case all of the queues grow to infinity and the dominant system $\mathcal{S}^{[1]}$ becomes indistinguishable from the original system $\mathcal{S}$ [13]. In order to improve upon these loose bounds, we make use of the *stability rank* of the queues as introduced in [15]. Let $\mathcal{S}^{[n]}$ denote a dominant system in which sources $s_n, s_{n+1}, \ldots, s_N$ transmit dummy packets when empty while sources $s_1, s_2, \ldots, s_{n-1}$ do not. The proof for the following theorem is the same as in the original [15] with the exception of the constant α_n.

Theorem 6. *Given $\boldsymbol{\lambda}_{\mathbf{N}}$ and $\mathbf{p}_{\mathbf{N}}$, we order the indices of the sources so that*

$$\frac{\lambda_1(1-p_1)}{\alpha_1 p_1} \leq \ldots \leq \frac{\lambda_N(1-p_N)}{\alpha_N p_N}. \tag{2.73}$$

Then in system $\mathcal{S}$ and any dominant system $\mathcal{S}^{[n]}$, if queue i is stable and $j < i$, then *queue j is also stable.*

We can find conditions for the stability of the system through the procedure outlined in [15]. After ordering the sources according to stability rank, we first check for stability of s_1 in system $\mathcal{S}^{[1]}$. If we find that s_1 is unstable, we can conclude that the entire system is unstable. However, if s_1 is stable, we proceed by examining s_2 in system $\mathcal{S}^{[2]}$. The queue at s_1 is known to be stable in $\mathcal{S}^{[2]}$ due to the stability rank. Given stability of s_1, we will check whether s_2 is stable in $\mathcal{S}^{[2]}$. The procedure continues in which the stability of s_n in system $\mathcal{S}^{[n]}$ is verified assuming that sources $s_1, s_2, \ldots, s_{n-1}$ are all stable in $\mathcal{S}^{[n]}$. If we can finally conclude that source s_N is stable in system $\mathcal{S}^{[N]}$, then this implies that the original system is stable.

Assuming that sources $s_1, s_2, \ldots, s_{n-1}$ are all stable, we will develop bounds on the average service rate $\mu_n^{[n]}$ for source n in system $\mathcal{S}^{[n]}$ in order to help determine whether s_n is stable. We can express $\mu_n^{[n]}$ as

$$\mu_n^{[n]} = \alpha_n p_n P_E^{[n]} \prod_{j=n+1}^{N} (1 - p_j), \tag{2.74}$$

where $P_E^{[n]}$ is the probability that none of the sources $s_1, s_2, \ldots, s_{n-1}$ transmit in the dominant system $\mathcal{S}^{[n]}$. One way to bound $\mu_n^{[n]}$ is by bounding $P_E^{[n]}$ when expressed as

$$\begin{aligned} P_E^{[n]} = 1 &- Pr\{\text{only one of } s_1, s_2, \ldots, s_{n-1} \text{ transmits}\} \\ &- Pr\{\text{more than one of } s_1, s_2, \ldots, s_{n-1} \text{ transmit}\}. \end{aligned} \tag{2.75}$$

The bounds on $\mu_n^{[n]}$ result in the following two theorems.

Theorem 7. *Sufficient condition. Given an N source, M destination random access system with $\boldsymbol{\lambda_N}$, $\mathbf{p_N}$ and the sources ordered according to the stability rank as in (2.73), if $\forall n$, $1 \leq n \leq N$, $\lambda_n < B_n$, where B_n is defined below, then the system is stable.*

$$\begin{aligned} B_n &= \max(C_n, D_n) \\ B_1 &= \alpha_1 p_1 \prod_{j=2}^{N}(1-p_j) \end{aligned}$$

$$C_n = \frac{\alpha_n p_n}{1-p_n}\left[\prod_{j=n}^{N}(1-p_j) - \frac{\sum_{i=1}^{k-1}\lambda_i}{\min\limits_{1\leq l\leq n-1}\alpha_l} - \frac{1}{2}\sum_{j=1}^{n-1}\left(\frac{\lambda_j p_j}{B_j}\prod_{i=n}^{N}(1-p_i) - \frac{\lambda_j}{\alpha_j}\right)\right]$$

$$D_n = \frac{\alpha_n p_n}{1-p_n}\prod_{j=1}^{N}(1-p_j)\left(1+\sum_{i=1}^{n-1}\left(1-\frac{\lambda_i}{B_i}\right)\frac{p_i}{1-p_i}\right)$$

Proof. Our proof follows the one given in [15]. A sufficient condition for stability corresponds to bounding $\mu_n^{[n]}$ from below. We obtain two separate lower bounds, C_n and D_n, and take their maximum to obtain the result. The bound C_n is derived by bounding the expression in (2.75). Since sources $s_1, s_2, \ldots, s_{n-1}$ are stable, the following holds.

$$\Pr\{\text{success by one of } s_1, s_2, \ldots, s_{n-1}\} = \sum_{i=1}^{n-1}\lambda_i$$

The above probability is lower bounded by

$$\left(\min_{1\leq l\leq n}\alpha_l\right)\Pr\{\textit{one} \text{ of } s_1, s_2, \ldots, s_{n-1} \text{ transmits}\}\prod_{j=n}^{N}(1-p_j),$$

therefore,

$$\Pr\{\textit{one} \text{ of } s_1, s_2, \ldots, s_{n-1} \text{ transmits}\} \leq \frac{\sum_{i=1}^{n-1}\lambda_i}{\left(\min\limits_{1\leq l\leq n}\alpha_l\right)\prod_{j=n}^{N}(1-p_j)}. \qquad (2.76)$$

This provides a bound for the second term in (2.75). For the third term, we first develop an expression for the probability that a packet sent by s_j, where $j \in [1, n-1]$, collides with at least one other packet sent by the other sources among $s_1, s_2, \ldots, s_{n-1}$. In doing so, we define the following sets of sources, where source s_j, $j \in [1, n-1]$, is excluded from the set.

$$\mathcal{A}_{n-1}^{j} = \{s_1, s_2, \ldots, s_{j-1}, s_{j+1}, \ldots, s_{n-1}\}$$

$$\mathcal{A}_{N}^{j} = \{s_1, s_2, \ldots, s_{j-1}, s_{j+1}, \ldots, s_{N}\}$$

Then

$$\begin{aligned}
&\Pr\{\text{packet from } s_j \text{ collides with others from } \mathcal{A}_{n-1}^{j}\} \\
&\qquad = \Pr\{s_j \text{ transmits}\} - \Pr\{s_j \text{ transmits, none from } \mathcal{A}_{n-1}^{j} \text{ transmit}\} \\
&\qquad = p_j \Pr\{s_j \text{ backlogged}\} - \frac{\Pr\{s_j \text{ transmits, none from } \mathcal{A}_{N}^{j} \text{ transmit}\}}{\prod_{i=n}^{N}(1-p_i)} \\
&\qquad = p_j \frac{\lambda_j}{\mu_j^{[n]}} - \frac{\lambda_j}{\alpha_j \prod_{i=n}^{N}(1-p_i)}.
\end{aligned}$$

We next make use of the result shown in [13] that for $j < n$, $\mu_j^{[j]} \leq \mu_j^{[n]}$. Thus, the third term in (2.75) is upper bounded as follows.

$$\Pr\{\text{more than one of } s_1, s_2, \ldots, s_{n-1} \text{ transmit}\} \leq \frac{1}{2} \sum_{j=1}^{n-1} \left(p_j \frac{\lambda_j}{\mu_j^{[j]}} - \frac{\lambda_j}{\alpha_j \prod_{i=n}^{N}(1-p_i)} \right) \tag{2.77}$$

By combining (2.75), (2.76), and (2.77) we obtain the lower bound

$$P_E^{[n]} \geq 1 - \frac{\sum_{i=1}^{n-1} \lambda_i}{\left(\min\limits_{1 \leq l \leq n} \alpha_l\right) \prod_{j=n}^{N}(1-p_j)} - \frac{1}{2} \sum_{j=1}^{n-1} \left(p_j \frac{\lambda_j}{\mu_j^{[j]}} - \frac{\lambda_j}{\alpha_j \prod_{i=n}^{N}(1-p_i)} \right). \tag{2.78}$$

Together with (2.74), this provides C_n, our first lower bound on $\mu_n^{[n]}$.

The other lower bound on $\mu_n^{[n]}$ is derived using an approach from [13]. We have adapted it for the multicast problem below and refer to it as D_n.

$$\begin{aligned}
\mu_n^{[n]} \geq \frac{\alpha_n p_n}{1-p_n} &\Pr\{\text{no other source transmits}\} \\
&+ \frac{\alpha_n p_n}{1-p_n} \Pr\{\text{one of } s_1, s_2, \ldots, s_{n-1} \text{ transmits but is empty}\} \\
&= \frac{\alpha_n p_n}{1-p_n} \prod_{j=1}^{N}(1-p_j) \left(1 + \sum_{i=1}^{n-1} \left(1 - \frac{\lambda_i}{\mu_i^{[i]}}\right) \frac{p_i}{1-p_i}\right) = D_n. \quad (2.79)
\end{aligned}$$

Note that we can express the exact value of $\mu_1^{[1]}$ as $\mu_1^{[1]} = \alpha_1 p_1 \prod_{j=2}^{N}(1-p_j)$. By beginning with $B_1 = \mu_1^{[1]}$ we can iterate through n values $2, \ldots, N$ to obtain the result. □

Theorem 8. *Necessary condition. Given an N source, M destination random access system with $\boldsymbol{\lambda_N}$, $\mathbf{p_N}$ and the sources ordered according to the stability rank as in (2.73), a necessary condition for stability of the system is that $\forall n$,*

$$\lambda_n \leq \frac{\alpha_n p_n}{1-p_n} \left(\prod_{j=n}^{N}(1-p_j) - \frac{\sum_{i=1}^{n-1} \lambda_i}{\max\limits_{1 \leq l \leq n-1} \alpha_l} \right).$$

Proof. The proof develops upper bounds on $\mu_n^{[n]}$ and thus on $P_E^{[n]}$. The technique is similar to the one used in finding C_n in the proof of Theorem 7. □

2.3.3 Numerical examples

We first observe the effect of the number of destinations on the stable and saturated throughput regions as shown in Fig. 2.6. These results are for $N = 2$ sources and $M = 2, 5$, and 15 destinations. The results are generated using the approach outlined for $N = 2$, $M = 2$ in Theorems 4 and 5 with the exception that

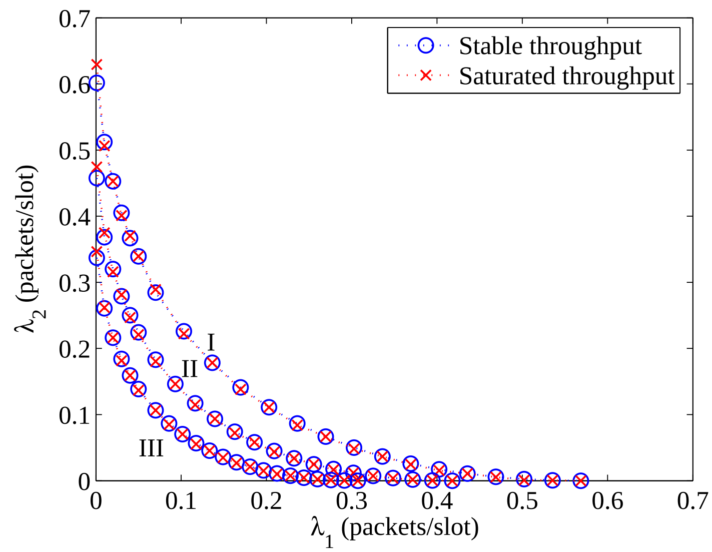

Figure 2.6: Stable and saturated throughput regions for N=2 sources and (I) 2, (II) 5, and (III) 15 destinations when using the retransmission scheme. The reception probabilities are $q_{1|1} = 0.7$, $q_{2|2} = 0.8$.

we use the backlogged and empty service rates for M destinations as given in (2.71) and (2.72). The results demonstrate that as the number of destinations increases, the throughput regions diminish in size. Additionally, we observe that the stable and saturated throughput regions coincide for $N = 2$ sources and arbitrarily many destinations, as indicated by Theorem 5.

A collection of results on stability and throughput for various values of N and M are shown in Tables 2.1 and 2.2. To generate these results, we fix $\lambda_1, \lambda_2, \ldots, \lambda_{N-1}$ and maximize λ_N over all $\mathbf{p_N}$ subject to $\lambda_N < \mu_{Nb}$. In Table 2.1 we observe that as the number of sources N increases, the throughput regions diminish in size. The values in Table 2.2 demonstrate the effect of the channel reception probabilities and the number of destinations on the throughput regions. The trends in these results are the same as those observed for a network of $N = 2$ sources. Furthermore, in all

Table 2.1: Saturated throughput (T) and bounds on the stable throughput (S) for a network of $N = 5, 10$ sources and $M = 10$ destinations. The reception probabilities are $q_{n|n} = 0.8\ \forall n$. The values of $\lambda_1, \ldots, \lambda_{N-1}$ are fixed and λ_N has been optimized over all $\mathbf{p}$.

N	$\lambda_1, \ldots, \lambda_{N-1}$	S-upper	S-lower	T
5	$[0.010, 0.010, 0.010, 0.010]$	0.2078	0.1939	0.1939
	$[0.070, 0.020, 0.010, 0.010]$	0.1051	0.0789	0.0789
	$[0.035, 0.035, 0.035, 0.035]$	0.0751	0.0362	0.0362
	$[0.050, 0.035, 0.035, 0.035]$	0.0602	0.0223	0.0223
10	$[0.010, 0.010, \ldots 0.010]$	0.1266	0.0912	0.0912
	$[0.070, 0.010, \ldots 0.010]$	0.0679	0.0252	0.0252
	$[0.017, 0.017, \ldots 0.017]$	0.0621	0.0137	0.0137
	$[0.020, 0.017, \ldots 0.017]$	0.0591	0.0108	0.0108

Table 2.2: Saturated throughput (T) and bounds on the stable throughput (S) for a network of $N = 4$ sources and $M = 8, 10$ destinations. The values λ_1, λ_2, λ_3 are fixed and λ_4 has been optimized over all $\mathbf{p}$.

M	$q_{1\|1}$	$q_{2\|2}$	$q_{3\|3}$	$q_{4\|4}$	λ_1	λ_2	λ_3	S-upper	S-lower	T
8	0.9	0.8	0.7	0.9	0.01	0.01	0.01	0.3648	0.3213	0.3213
					0.07	0.02	0.01	0.2125	0.1672	0.1672
					0.05	0.05	0.05	0.1363	0.0566	0.0566
					0.07	0.05	0.05	0.1090	0.0376	0.0376
8	0.8	0.8	0.8	0.8	0.01	0.01	0.01	0.2527	0.2433	0.2434
					0.07	0.02	0.01	0.1294	0.1090	0.1090
					0.05	0.05	0.05	0.0784	0.0428	0.0428
					0.07	0.05	0.05	0.0587	0.0254	0.0254
10	0.8	0.8	0.8	0.8	0.01	0.01	0.01	0.2329	0.2236	0.2236
					0.07	0.02	0.01	0.1153	0.0951	0.0951
					0.05	0.05	0.05	0.0651	0.0318	0.0321
					0.065	0.05	0.05	0.0503	0.0196	0.0196

cases, the saturated throughput values fall within the upper and lower bounds on the stable throughput values. As such, these results support the conjecture that the stable and saturated throughput regions coincide. Of special note, the lower bound for the stable throughput and the saturated throughput value appear to be equal in many cases. This is not entirely true. In the results shown here, the saturated throughput is in fact slightly larger than the lower bound on the stable throughput, but the difference is at most 10^{-6}.

2.4 Discussion

In this chapter we explored the stable throughput, saturated throughput, and information-theoretic capacity regions for a random access system in which nodes carry out multicast transmissions. The interest in exploring these three different notions of data rate stems in part from the comparisons drawn between them in previous work on random access for unicast transmission.

In Section 2.2 we characterized the throughput and capacity regions for a network with two source nodes and two destination nodes. We showed that in this small network, the stable throughput and saturated throughput regions coincide. We also compared two different transmission schemes and showed that if K, the number of packets drawn from the queue for random linear coding, is sufficiently large, then random linear coding provides a larger stable throughput region than a retransmission scheme. Finally, we saw that both the retransmissions and random linear coding schemes provide a stable throughput region that is outer bounded by the Shannon capacity region - the ultimate upper limit on performance.

In Section 2.3 we examined the throughput regions for random access multicast with N sources and M destinations, where both N and M are arbitrary but finite. For this arbitrarily large network, the stable throughput region is unknown for random access unicast; it remains so for multicast as well. However we developed bounds on the stable throughput region for the retransmission scheme, and our numerical examples of those bounds support the conjecture that the stable and saturated throughput regions coincide.

The implications of these results are clear. In order to obtain large data rates for random access multicast, the use of random linear coding is preferable to retransmissions - but only if K is sufficiently large. If K is chosen as large as possible, this will ensure a large rate region, but as a trade-off, will also lead to a large delay. The queueing delay for random linear coding, including a coding approach that adapts to traffic load, is explored in Chapter 4. Another implication of the results of the present chapter is that for multicast transmission, the saturated throughput region closely approximates (and possibly coincides with) the stable throughput region. This motivates the narrowing of our discussion in Chapter 3 to the saturated throughput, which can be characterized exactly.

Chapter 3

Proportionally fair multicast throughput for a random access network of general topology

3.1 Background

In this chapter we consider the multicast throughput in a large random access network with arbitrary connection topology. We focus our analysis on the saturated throughput, which as shown in Chapter 2, closely approximates and possibly coincides with both the stable throughput and the Shannon capacity for random access multicast. We propose distributed schemes to assign the random access transmission probabilities p_n in order to optimize a weighted proportional fairness objective function.

The model we consider is as follows. The network consists of a finite set $\mathcal{N}$ of nodes and each node n serves multiple multicast flows. Source n together with the one-hop neighbors in a multicast flow emanating from that source constitute a *multicast tree* and the set of receivers in the tree is denoted $\mathcal{D}_{nm}$, where m indexes the multicast tree. Time is slotted, and in each time slot, node n accesses the channel with probability p_n and chooses to transmit on multicast tree m with probability p_{nm}. Thus $p_n = \sum_m p_{nm}$. A transmission by node n interferes with reception at all nodes in a set $\mathcal{N}_n$; interference at the receiver causes destruction of any transmitted packets intended for that receiver. Interference is not assumed to be symmetric - for nodes $n, k \in \mathcal{N}$, $k \in \mathcal{N}_n$ does not necessarily imply that $n \in \mathcal{N}_k$. We consider two different multicast scenarios: non-guaranteed and guaranteed multicast. In both cases, we characterize the network throughput and find the access probabilities that maximize the proportional fairness objective. Our results in this chapter are summarized below.

- **Non-guaranteed multicast:** We assign a distinct link weight $w_{nmd} > 0$ for all destinations $d \in \mathcal{D}_{nm}$. The weighted proportional fairness problem is

$$\max_{\boldsymbol{\mu_{\mathcal{I}}}} \sum_{(n,\mathcal{D}_{nm},d)} w_{nmd} \log \mu_{nmd} \tag{3.1}$$

  where μ_{nmd} is the throughput obtained by receiver d on tree (n, D_{nm}), i.e. the number of packets successfully received by receiver d on tree (n, D_{nm}) per unit time, and $\boldsymbol{\mu_{\mathcal{I}}} = \{\mu_{nmd}\}$ is the vector of μ_{nmd} for all links in the network. This approach is of interest for applications in which it is beneficial, but not required, for each destination in the multicast tree to receive the transmitted packets.

- **Guaranteed multicast:** For applications that require that all destinations in the multicast tree *must* receive the transmitted packets, we assign weight $w_{nm} > 0$ to the multicast tree $(n, \mathcal{D}_{nm})$ and are interested in the problem

$$\max_{\boldsymbol{\mu_{\mathcal{T}}}} \sum_{(n,\mathcal{D}_{nm})} w_{nm} \log \mu_{nm} \tag{3.2}$$

  where μ_{nm} is the multicast throughput on tree (n, D_{nm}), i.e. the number of packets that have been successfully received by all receivers on the tree per unit time, and $\boldsymbol{\mu}_{\mathcal{T}} = \{\mu_{nm}\}$ is the vector of μ_{nm} for all trees in the network. We emphasize that although all receivers need to receive the packet in order to get credit, they do not have to receive it in the same timeslot. We characterize μ_{nm} for a retransmissions strategy as well as for a random packet (fountain) coding strategy. We then recast the weighted proportional fairness problem by introducing the constraint $\mu_{nm} \leq \mu_{nmd}$, $\forall d \in \mathcal{D}_{nm}$ and provide an algorithm that converges to the optimal set of access probabilities p_{nm}.

Previous work has considered optimal throughput allocation for *unicast* transmission in random access. In works by Kar et al. [29] and Gupta and Stolyar [30], the access probabilities are assigned in order to optimize a weighted proportional fairness objective function of the single-hop throughput. The access probabilities in this case can be computed using only the link weights in the local neighborhood (2-hop neighbors). In [31], Wang and Kar propose distributed algorithms to solve the weighted proportional fairness problem for end-to-end throughput. A family of objective functions which encapsulates proportional fairness, max-min fairness, and other types of fairness is presented by Mo and Walrand in [32]; this family of objective functions is applied to a random access network by Lee et al. in [33].

There has also been previous work in designing efficient multicast transmission strategies. In [34], Chaporkar and Sarkar propose a strategy in which the number of receivers that are available for reception is compared to a threshold in order to determine whether the source should transmit. The same authors study a multicast transmission strategy aimed at minimizing delay in [35]. In the models considered in

[34, 35], the ability of a node to receive a transmitted packet is determined by some exogenous process, i.e., they are given as part of the problem input; in contrast, in our work, the ability of a node to receive depends on the access probabilities of neighboring nodes as determined by our algorithm. In [36], Kar et al. study the end-to-end multicast rate control problem under the assumption that different nodes in the multicast group can receive at different rates, which is similar to the non-guaranteed multicast we consider here. The work in [36] treats the multicast problem for a wireline network, whereas in the present work we consider wireless transmission.

3.2 Model

An example of the setting we consider in this chapter is shown in Fig. 3.2. The network consists of a finite set $\mathcal{N}$ of nodes. Node $n \in \mathcal{N}$ has traffic to send to some subset of its neighbors. Let $\mathcal{D}_n \subseteq \mathcal{N}$ denote the set of nodes for which n has some traffic to send. The set $\mathcal{D}_n$ consists of subsets $\mathcal{D}_{nm}$, $m \in \mathcal{D}_n$, for which node n intends to send the same traffic, i.e., node n will *multicast* traffic to all nodes in the set $\mathcal{D}_{nm}$ where $|\mathcal{D}_{nm}| \geq 1$. This model can account for a mix of unicast and multicast traffic since unicast would correspond to $|\mathcal{D}_{nm}| = 1$.

We assume that time is slotted and in each time slot, node n can attempt transmission of a packet to (all nodes in) one of the sets $\mathcal{D}_{nm}$. We say that node n makes a transmission attempt on multicast tree $(n, \mathcal{D}_{nm})$. (This is a tree of depth

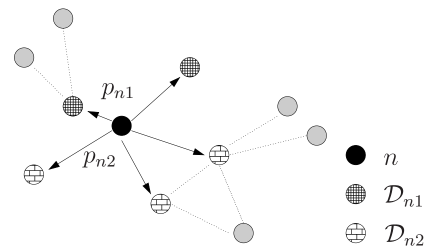

Figure 3.1: An example of multicasting in a network of general connection topology.

1.) Denote by

$$\mathcal{T} = \{(n, \mathcal{D}_{nm}) : n \in \mathcal{N}, \mathcal{D}_{nm} \subseteq \mathcal{D}_n\} \tag{3.3}$$

the set of all (depth-1) multicast trees in the network. Additionally, let $d \in \mathcal{D}_{nm}$ denote a leaf in the tree $(n, \mathcal{D}_{nm})$ and $(n, \mathcal{D}_{nm}, d)$ denote the link between n and d in the tree. The set $\mathcal{I}$ denotes the set of all multicast links in the network.

$$\mathcal{I} = \{(n, \mathcal{D}_{nm}, d) : n \in \mathcal{N}, \mathcal{D}_{nm} \subseteq \mathcal{D}_n, d \in \mathcal{D}_{nm}\} \tag{3.4}$$

We make the following assumptions with regard to transmissions and the channel. (I) A node n can transmit on at most one multicast tree $(n, \mathcal{D}_{nm})$ in each time slot. (II) A node can receive at most 1 packet per slot. (III) Any transmission attempt by node n will interfere with and destroy any attempt to receive a packet at any of the nodes in a set $\mathcal{N}_n$ where $\mathcal{D}_n \subseteq \mathcal{N}_n$ and $n \in \mathcal{N}_n$. Nodes in the network make use of a random access strategy as follows. In each time slot, node n transmits with probability p_n. When n transmits, it chooses a particular multicast tree to transmit on from among $(n, \mathcal{D}_{nm}), \mathcal{D}_{nm} \subseteq \mathcal{D}_n$, randomly with probability p_{nm}/p_n where $\sum_{m \in \mathcal{D}_n} p_{nm} = p_n$. We assume throughout that each multicast tree $(n, \mathcal{D}_{nm})) \in \mathcal{T}$ in the network has an infinite backlog of packets awaiting transmission on the tree; equivalently, we consider the saturated throughput of the network.

3.3 Non-guaranteed multicast

We first focus our attention on the optimization problem in Eqn. (3.1). Note that for a transmitting node n and receiving node d, the weight w_{nmd} and throughput μ_{nmd} may take different values depending on the multicast tree (or multicast flow) of interest, i.e., links $(n, \mathcal{D}_{nm}, d)$ and $(n, \mathcal{D}_{nl}, d)$, $m \neq l$, are distinct, with $w_{nmd} \neq w_{nld}$. For a given link $(n, \mathcal{D}_{nm}, d)$ it can be shown that μ_{nmd} is maximized by a policy in which each packet is transmitted on $(n, \mathcal{D}_{nm})$ *only once.* Once the packet has been transmitted on $(n, \mathcal{D}_{nm})$, it is removed from memory at n. Then the multicast transmission process on $(n, \mathcal{D}_{nm})$ consists of only a single transmission and the average throughput is given by

$$\mu_{nmd} = p_{nm} \prod_{k: d \in \mathcal{N}_k, k \neq n} (1 - p_k). \tag{3.5}$$

Let $\mathcal{L}_n$ denote the set of links $(l, \mathcal{D}_{lk}, j)$, $j \in \mathcal{D}_{lk}$, that either originate at n or are such that a transmission by n causes interference and destroys that on $(l, \mathcal{D}_{lk}, j)$.

$$\mathcal{L}_n = \{(l, \mathcal{D}_{lk}, j) : j \in \mathcal{D}_{lk}, j \in \mathcal{N}_n\} \tag{3.6}$$

Define further $\mathcal{L}_n^-$ as the set of links in $\mathcal{L}_n$ that do not originate at n.

$$\mathcal{L}_n^- = \{(l, \mathcal{D}_{lk}, j) \in \mathcal{L}_n : l \neq n\} \tag{3.7}$$

The following theorem describes the optimal throughput allocation for non-guaranteed multicast. Our analysis is based on techniques used in [30].

Theorem 9. *For arbitrary sets of positive real-valued weights* $\{w_{nmd}, (n, \mathcal{D}_{nm}, d) \in$

$\mathcal{I}\}$*, there exists a unique set of access probabilities that maximize the function*

$$\sum_{(n,\mathcal{D}_{nm},d)\in\mathcal{I}} w_{nmd} \log \mu_{nmd}. \tag{3.8}$$

The optimal probabilities are given by

$$p_{nm} = \frac{\sum_{d\in\mathcal{D}_{nm}} w_{nmd}}{\sum_{(l,\mathcal{D}_{lk},j)\in\mathcal{L}_n} w_{lkj}}. \tag{3.9}$$

Proof. The proof is similar to the proof of [30, Theorem 1]. In addition, we note that

$$\mathcal{L}_n = \mathcal{L}_n^- \bigcup \left\{ \bigcup_{m\in\mathcal{D}_n} \left\{ \bigcup_{d\in\mathcal{D}_{nm}} (n, \mathcal{D}_{nm}, d) \right\}\right\} \tag{3.10}$$

which gives rise to the sum $\sum_{d\in\mathcal{D}_{nm}} w_{nmd}$ in the numerator of (3.9). □

3.4 Guaranteed multicast

We now address the weighted proportional fairness problem in Eqn. (3.2). Here μ_{nm} denotes the throughput for the tree, or the number of packets per slot that can be successfully transmitted to *all* recipients on the tree. The multicast throughput μ_{nm} is in general difficult to describe. The difficulty arises in that the intended recipients on $(n, \mathcal{D}_{nm})$ have distinct sets of neighboring, interfering nodes and as such, the intended recipients have distinct probabilities of successfully receiving a packet transmitted by n. Node n might employ different strategies to ensure successful reception $\forall d \in \mathcal{D}_{nm}$, e.g., node n might retransmit the packet on $(n, \mathcal{D}_{nm})$ as long as the packet has not been received by some $d \in \mathcal{D}_{nm}$. Regardless of the transmission strategy that node n employs, due to the non-uniformity of the channels on the links $(n, \mathcal{D}_{nm}, d)$, $d \in \mathcal{D}_{nm}$, the process by which node n transmits on $(n, \mathcal{D}_{nm})$ to ensure successful reception $\forall d \in \mathcal{D}_{nm}$ is a process with memory.

In this section, we characterize μ_{nm} for a retransmission strategy as well as for a random packet (fountain) coding strategy. We then recast the weighted proportional fairness problem by introducing the constraint $\mu_{nm} \leq \mu_{nmd}$, $\forall d \in \mathcal{D}_{nm}$ and provide an algorithm that converges to the optimal set of access probabilities p_{nm}.

3.4.1 Throughput: retransmission strategy

We now consider a strategy whereby node n repeatedly transmits a packet on the multicast tree $(n, \mathcal{D}_{nm})$ until it has received a feedback acknowledgement of successful reception of the packet from all intended recipients in the tree, i.e., $\forall d \in \mathcal{D}_{nm}$. For destination $d \in \mathcal{D}_{nm}$, the probability of successful reception of a packet in any given time slot is equal to μ_{nmd}, the average unicast throughput on link $(n, \mathcal{D}_{nm}, d)$. The transmission time to destination d, or the number of slots that elapse from the time a packet is first available for transmission on tree $(n, \mathcal{D}_{nm})$ until the end of the slot in which destination d successfully receives the packet, is a geometric random variable X_d with parameter μ_{nmd}, where $\Pr(X_d > 0) = 1$. (Equivalently, X_d is the number of independent coin tosses needed up to and including the first toss that results in a heads, where heads occurs in each toss with probability μ_{nmd}.) The total transmission time, or the number of slots needed until all intended recipients on $(n, \mathcal{D}_{nm})$ receive the packet, has the expected value $E[\max_{d \in \mathcal{D}_{nm}} X_d]$. As noted earlier, the transmission process described above is a process with memory. Additionally, the random variables $X_d, d \in \mathcal{D}_{nm}$, may be correlated since the intended recipients may have overlapping sets of interfering nodes. In the following lemma, we bound the throughput for this retransmission strategy.

Lemma 10. *The guaranteed multicast throughput μ_{nm}^R for the retransmission strategy is bounded as*

$$\frac{\mu_{nmd}^{min}}{(1+\log|\mathcal{D}_{nm}|)\left(1+\frac{2}{e\left(1-\frac{1}{e|\mathcal{D}_{nm}|}\right)^2}\right)} \leq \mu_{nm}^R \leq \mu_{nmd}^{min} \tag{3.11}$$

where

$$\mu_{nmd}^{min} = p_{nm} \min_{d \in \mathcal{D}_{nm}} \prod_{k: d \in \mathcal{N}_k, k \neq n} (1-p_k) \tag{3.12}$$

Proof. The result follows by bounding the expected total transmission time $E[\max_{d \in \mathcal{D}_{nm}} X_d]$ and by noting that $\mu_{nm}^R = 1/E[\max_{d \in \mathcal{D}_{nm}} X_d]$. The upper bound in (3.11) can be shown as follows.

$$E[\max_d X_d] \overset{(a)}{\geq} \max_d E[X_d] = \max_d \frac{1}{\mu_{nmd}} = \frac{1}{\mu_{nmd}^{min}}$$

where (a) holds by Jensen's inequality since the max function is convex. To show the lower bound in (3.11), let c denote a fixed constant. The following sequence of inequalities holds for any $c > 1$.

$$
\begin{aligned}
E[\max_d X_d] &\leq \sum_{i=1}^{\infty} \frac{ic}{\mu_{nmd}^{min}} \Pr\left(\frac{(i-1)c}{\mu_{nmd}^{min}} < \max_d X_d \leq \frac{ic}{\mu_{nmd}^{min}}\right) \\
&\leq \sum_{i=1}^{\infty} \frac{ic}{\mu_{nmd}^{min}} \Pr\left(\frac{(i-1)c}{\mu_{nmd}^{min}} < \max_d X_d \leq \infty\right) \\
&\stackrel{(b)}{\leq} \frac{c}{\mu_{nmd}^{min}} + \sum_{i=2}^{\infty} \frac{ic}{\mu_{nmd}^{min}} \sum_d (1-\mu_{nmd})^{\frac{(i-1)c}{\mu_{nmd}^{min}}} \\
&\stackrel{(c)}{\leq} \frac{c}{\mu_{nmd}^{min}} \left(1 + |\mathcal{D}_{nm}| \sum_{i=1}^{\infty} (i+1) e^{-ci}\right) \\
&= \frac{c}{\mu_{nmd}^{min}} \left(1 + \frac{2|\mathcal{D}_{nm}| e^{-c} - |\mathcal{D}_{nm}| e^{-2c}}{(1-e^{-c})^2}\right)
\end{aligned}
$$

where (b) holds by the union bound and by using the fact that X_d is geometrically distributed and (c) holds since $(1-\mu_{nmd})^{(i-1)c/\mu_{nmd}^{min}} \leq e^{-(i-1)c\mu_{nmd}/\mu_{nmd}^{min}} \leq e^{-(i-1)c}$. By taking $c = 1 + \log |\mathcal{D}_{nm}|$ we obtain

$$
E[\max_d X_d] \leq \frac{1}{\mu_{nmd}^{min}} (1 + \log |\mathcal{D}_{nm}|) \left(1 + \frac{2}{e\left(1 - \frac{1}{e|\mathcal{D}_{nm}|}\right)^2}\right)
$$

This provides the lower bound in (3.11). □

Lemma 10 shows that when using the retransmission strategy, the guaranteed multicast throughput on tree $(n, \mathcal{D}_{nm})$ is upper bounded by the (unicast) throughput to the destination $d \in \mathcal{D}_{nm}$ that has the *smallest* (unicast) throughput among all destinations in the multicast tree.

3.4.2 Throughput: coded transmission strategy

We now consider a strategy whereby node n performs coding over groups of packets awaiting transmission on multicast tree $(n, \mathcal{D}_{nm})$. Let K denote the number of packets involved in encoding, or the number of inputs to the encoder. The coding strategy works as follows. For each transmission attempt on $(n, \mathcal{D}_{nm})$, the encoder at node n samples from a *degree distribution* to obtain a value w between 1 and K. The encoder then selects w packets randomly and uniformly from among the K input packets and forms the sum (modulo-2) of these w packets, which is the output of the encoder. The encoder output is then transmitted on the multicast tree $(n, \mathcal{D}_{nm})$. We assume that the coefficients of the random sum are transmitted in the header of the packet; these coefficients will be needed to perform decoding.

In the next transmission attempt on $(n, \mathcal{D}_{nm})$, the encoder follows the same procedure to independently and randomly form a new output for transmission on the tree. This process continues as the intended recipients on $(n, \mathcal{D}_{nm})$ collect coded versions of the K packets. Once a destination $d \in \mathcal{D}_{nm}$ has received N encoder outputs, where $N \gtrapprox K$, and is able to decode the original K packets, it sends a feedback acknowledgement to node n. As soon as node n has collected acknowledgements from all intended recipients in $\mathcal{D}_{nm}$, the transmission process is complete and node n can commence encoding and transmission of another group of K packets awaiting transmission on $(n, \mathcal{D}_{nm})$. The coding strategy described here is a general description of Fountain coding.

For an appropriately designed degree distribution, the value of N can be made arbitrarily close to K and decoding can be performed by belief propagation with low computational complexity. The LT-codes invented by Luby in [6] can be implemented through use of the Robust Soliton Distribution, for which $N = K + O(\sqrt{K}(\log K)^2)$ ensures that belief propagation decoding can be performed with error at most k^{-c}, $c > 0$ and decoding complexity on the order of $K \log K$. The Raptor codes described in [7], which were proposed by Shokrollahi and involve concatenating an error-correcting pre-code with an LT-code, provide better performance in the sense that N can be made even closer to K with small error probability and low-complexity decoding. For the purpose of describing the multicast throughput, we do not assume the use of a particular Fountain code. More generally, we assume that for a given error probability and decoding complexity, there exists a deterministic value N which is arbitrarily close to (but slightly larger than) K.

For destination $d \in \mathcal{D}_{nm}$, the probability of successful reception of an encoder output in any given time slot is μ_{nmd}. The transmission time to destination d, or the number of slots that elapse from the time a group of K packets is first available for transmission on tree $(n, \mathcal{D}_{nm})$ until the end of the slot in which destination d successfully receives the N^{th} encoder output, is given by $Y_d = X_{d,1} + X_{d,2} + \ldots + X_{d,N}$, where $X_{d,i}, 1 \leq i \leq N$ are independently, geometrically distributed with parameter μ_{nmd}. The geometric distribution of $X_{d,i}$ does not imply that a particular encoder output is retransmitted until received; it merely indicates that if a new encoder output is generated at each transmission attempt, then the number of slots that elapse until one encoder output is received is geometrically distributed. (Equivalently, if instead of tossing a single coin, we have a collection of coins with identical bias and choose a different coin for each toss, then the number of independent tosses until the first heads appears is geometric.) The total transmission time, or the number of slots needed for all destinations on $(n, \mathcal{D}_{nm})$ to receive N encoder outputs is

$E[\max_{d \in \mathcal{D}_{nm}} Y_d]$.

Lemma 11. *The guaranteed multicast throughput μ^C_{nm} for the coded transmission strategy is bounded as*

$$\mu^C_{nm} \leq \mu^{min}_{nmd} \tag{3.13}$$

where the bound holds with equality in the limit as $K \to \infty$.

Proof. We again make use of Jensen's inequality to show that

$$E[\max_{d \in \mathcal{D}_{nm}} Y_d] \geq \max_d E[Y_d] = \max_d \frac{N}{\mu_{nmd}} = \frac{N}{\mu^{min}_{nmd}}. \tag{3.14}$$

By using $\mu^C_{nm} = K/E[\max_d Y_d]$ and by the assumption that the Fountain code allows for N to be made arbitrarily close to K, the upper bound in the lemma follows.

To show when the upper bound holds with equality, we argue as follows. Let $\mathbf{Y}_d$ denote the vector of transmission times for $d \in \mathcal{D}_{nm}$ and let $\mathbf{Y}^a_d$ and $\mathbf{Y}^b_d$ denote two realizations of the vector $\mathbf{Y}_d$. The convexity of the max function means that for any realizations $\mathbf{Y}^a_d, \mathbf{Y}^b_d$

$$\max(\theta \mathbf{Y}^a_d + (1-\theta)\mathbf{Y}^b_d) \leq \theta \max \mathbf{Y}^a_d + (1-\theta) \max \mathbf{Y}^b_d \tag{3.15}$$

where $0 \leq \theta \leq 1$. As $K \to \infty$, by the strong law of large numbers, $Y_d \to N/\mu_{nmd}$ with probability 1 and (3.15) holds with equality. This is true even if the values of $\mu_{nmd}, d \in \mathcal{D}_{nm}$ are not distinct as long as ties are broken (i.e., a unique destination with the worst channel is selected) by an arbitrary but fixed rule. Since (3.15) holds with equality, Jensen's also holds with equality and the upper bound on the throughput is tight. ∎

Lemma 11 shows that as with the retransmission strategy, the guaranteed multicast throughput for the coded transmission strategy is upper bounded by the smallest unicast throughput among all destinations on the tree $(n, \mathcal{D}_{nm})$. However, unlike the retransmission strategy, the coded transmission strategy can reach the upper bound on the throughput, but at the cost of infinite delay.

3.4.3 Weighted proportional fairness for guaranteed multicast

We now return to the problem of maximizing the weighted proportional fairness for guaranteed multicast. We can formulate the problem as maximizing the weighted proportional fairness subject to a constraint on the throughput μ_{nm} given by Lemmas 10 and 11. Our constraint is

$$\mu_{nm} \leq \mu_{nmd}^{min} \iff \mu_{nm} \leq \mu_{nmd}, \quad \forall d \in \mathcal{D}_{nm} \tag{3.16}$$

where, as shown in Lemma 11, $\mu_{nm} = \mu_{nmd}^{min}$ is achievable by making use of the coded transmission policy. Our weighted proportional fairness problem $\mathbf{P}$ is stated below.

$$\begin{aligned}
\mathbf{P}: \quad \max \quad & \sum_{(n,\mathcal{D}_{nm}) \in \mathcal{T}} w_{nm} \log \mu_{nm} \\
\text{s.t.} \quad & \mu_{nm} \leq \mu_{nmd}, \quad \forall d \in \mathcal{D}_{nm}, \forall (n, \mathcal{D}_{nm}) \in \mathcal{T} \\
& \mu_{nmd} = c_{nmd}(\mathbf{p}), \quad \forall (n, \mathcal{D}_{nm}, d) \in \mathcal{I} \\
& 0 \leq p_{nm} \leq 1, \quad \forall (n, \mathcal{D}_{nm}) \in \mathcal{T} \\
& p_n \leq 1, \quad \forall n \in \mathcal{N} \\
& \mu_{nm} \geq 0, \quad \forall (n, \mathcal{D}_{nm}) \in \mathcal{T}
\end{aligned}$$

For link $(n, \mathcal{D}_{nm}, d)$, we define $c_{nmd}(\mathbf{p})$ as

$$c_{nmd}(\mathbf{p}) = p_{nm} \prod_{k:d \in \mathcal{N}_k, k \neq n} (1 - p_k) \tag{3.17}$$

where $\mathbf{p}$ is the vector of random access probabilities, $\mathbf{p} = \{p_{nm}, (n, \mathcal{D}_{nm}) \in \mathcal{T}\}$. The formulation of $\mathbf{P}$ for our guaranteed multicast problem is similar to the formulation for maximizing end-to-end proportional fairness in (unicast) random access networks, which is treated in [31] and [33]. Due to the form of $c_{nmd}(\mathbf{p})$, the problem is non-separable and non-convex in $\mathbf{p}$, which makes it difficult to provide a distributed algorithm that converges to the global optimum. However, due to the similarity of this problem to the end-to-end proportional fairness problem, we can easily adapt the Dual-Based Algorithm provided in [31], which overcomes these difficulties and is shown to converge to the optimal p_{nm}.

We proceed by decomposing $\mathbf{P}$ into two problems. The problem $\hat{\mathbf{P}}$ optimizes the same objective as $\mathbf{P}$ but is parameterized by the set of (unicast) link throughputs $\boldsymbol{\mu_{\mathcal{I}}} = (\mu_{nmd}, (n, \mathcal{D}_{nm}, d) \in \mathcal{I})$, meaning essentially that the random access probabilities $\mathbf{p}$ are assumed fixed. The problem $\hat{\mathbf{P}}$ is defined below.

$$\begin{aligned}
\hat{\mathbf{P}}: \quad \max \quad & \sum_{(n,\mathcal{D}_{nm}) \in \mathcal{T}} w_{nm} \log \mu_{nm} \\
\text{s.t.} \quad & \mu_{nm} \leq \mu_{nmd}, \quad \forall d \in \mathcal{D}_{nm}, \forall (n, \mathcal{D}_{nm}) \in \mathcal{T} \\
& \mu_{nm} \geq 0, \quad \forall (n, \mathcal{D}_{nm}) \in \mathcal{T}
\end{aligned}$$

Since $\mathbf{p}$ is fixed, the problem $\hat{\mathbf{P}}$ is equivalent to a rate allocation problem in a wired network. Since the log function in the objective function is strictly concave and the constraints are linear in μ_{nm}, $\hat{\mathbf{P}}$ is a convex problem and has no duality gap. Then

$\hat{\mathbf{P}}$ can be solved by the gradient projection method applied to its associated dual problem. This procedure is described below.

The solution to the problem $\hat{\mathbf{P}}$ will be a function of the link throughputs $\boldsymbol{\mu}_{\boldsymbol{\mathcal{I}}}$. Let $\hat{U}(\boldsymbol{\mu}_{\boldsymbol{\mathcal{I}}})$ denote the solution to $\hat{\mathbf{P}}$, defined as follows.

$$\hat{U}(\boldsymbol{\mu}_{\boldsymbol{\mathcal{I}}}) = \max\left\{ \sum_{(n,\mathcal{D}_{nm})\in\mathcal{T}} w_{nm}\log\mu_{nm} : \mu_{nm}\leq\mu_{nmd}, \right.$$
$$\left. \forall d\in\mathcal{D}_{nm}, \forall(n,\mathcal{D}_{nm})\in\mathcal{T} \right\} \quad (3.18)$$

The vector of link throughputs $\boldsymbol{\mu}_{\boldsymbol{\mathcal{I}}}$ will ultimately be a function of $\mathbf{p}$. We define the function $\widetilde{U}(\mathbf{p}) = \hat{U}(\mathbf{c}(\mathbf{p}))$, where $\mathbf{c}(\mathbf{p}) = (c_{nmd}(\mathbf{p}), (n,\mathcal{D}_{nm},d)\in\mathcal{I})$ and the problem $\tilde{\mathbf{P}}$ as follows.

$$\begin{aligned} \tilde{\mathbf{P}}: \quad \max \quad & \widetilde{U}(\mathbf{p}) \\ s.t. \quad & 0\leq p_{nm}\leq 1, \quad \forall(n,\mathcal{D}_{nm})\in\mathcal{T} \\ & p_n\leq 1, \quad \forall n\in\mathcal{N} \end{aligned}$$

Problem $\tilde{\mathbf{P}}$ is the problem we want to solve to determine $\mathbf{p}$. In the process of converging on the optimum $\mathbf{p}$, in each update we make we must optimize the vector of link throughputs $\boldsymbol{\mu}_{\boldsymbol{\mathcal{I}}}$, which is accomplished by solving problem $\hat{\mathbf{P}}$. The algorithm will work at two different time scales. Let t_1 denote time instants in the larger time scale in which we update $\mathbf{p}$. Let t_2 denote time in the smaller time scale, which is the time scale over which we solve $\hat{\mathbf{P}}$ through its dual problem.

We first describe the algorithm for solving the dual problem of $\hat{\mathbf{P}}$ when $\boldsymbol{\mu}_{\boldsymbol{\mathcal{I}}} = \mathbf{c}(\mathbf{p}^{(t_1,t_2)})$. The Lagrangian is given by $L^{(t_1,t_2)}(\boldsymbol{\mu}_{\boldsymbol{\mathcal{T}}}, \boldsymbol{\lambda})$, where $\boldsymbol{\mu}_{\boldsymbol{\mathcal{T}}} = (\mu_{nm}, (n,\mathcal{D}_{nm}\in\mathcal{T})$ denotes the vector of throughputs on the *multicast trees* in the network and

$\boldsymbol{\lambda} = (\lambda_{nmd}, (n, \mathcal{D}_{nm}, d) \in \mathcal{I})$ denotes the vector of Lagrange multipliers, or link prices. We have

$$L^{(t_1,t_2)}(\boldsymbol{\mu}_{\mathcal{T}}, \boldsymbol{\lambda}) = \sum_{(n,\mathcal{D}_{nm})\in\mathcal{T}} w_{nm} \log \mu_{nm} - \sum_{(n,\mathcal{D}_{nm},d)\in\mathcal{I}} \lambda_{nmd} \left(\mu_{nm} - \mu_{nmd}^{(t_1,t_2)}\right). \quad (3.19)$$

The solution to the dual problem of $\hat{\mathbf{P}}$ is

$$\boldsymbol{\lambda}^{*(t_1,t_2)} = \arg\min_{\boldsymbol{\lambda}\geq 0} \max_{\boldsymbol{\mu}_{\mathcal{T}}} L^{(t_1,t_2)}(\boldsymbol{\mu}_{\mathcal{T}}, \boldsymbol{\lambda}) \quad (3.20)$$

Note that the Lagrangian can be rewritten as

$$L^{(t_1,t_2)}(\boldsymbol{\mu}_{\mathcal{T}}, \boldsymbol{\lambda}) = \sum_{(n,\mathcal{D}_{nm})\in\mathcal{T}} \left(w_{nm} \log \mu_{nm} - \mu_{nm} \sum_{d\in\mathcal{D}_{nm}} \lambda_{nmd}\right) + \sum_{(n,\mathcal{D}_{nm},d)\in\mathcal{I}} \lambda_{nmd}\mu_{nmd}^{(t_1,t_2)} \quad (3.21)$$

and the first term is separable in μ_{nm}. The objective function for the dual problem of $\hat{\mathbf{P}}$ at $\boldsymbol{\mu}_{\mathcal{I}}$ is

$$\begin{aligned} D^{(t_1,t_2)}(\boldsymbol{\lambda}) &= \max_{\boldsymbol{\mu}_{\mathcal{T}}} L^{(t_1,t_2)}(\boldsymbol{\mu}_{\mathcal{T}}, \boldsymbol{\lambda}) \\ &= \sum_{(n,\mathcal{D}_{nm})\in\mathcal{T}} \max_{\mu_{nm}} \left(w_{nm} \log \mu_{nm} - \mu_{nm} \sum_{d\in\mathcal{D}_{nm}} \lambda_{nmd}\right) \\ &\qquad + \sum_{(n,\mathcal{D}_{nm},d)\in\mathcal{I}} \lambda_{nmd}\mu_{nmd}^{(t_1,t_2)} \end{aligned}$$

We can maximize $D^{(t_1,t_2)}(\boldsymbol{\lambda})$ over $\boldsymbol{\mu}_{\mathcal{T}}$ by setting

$$\mu_{nm} = \frac{w_{nm}}{\sum_{d\in\mathcal{D}_{nm}} \lambda_{nmd}} \quad (3.22)$$

The dual problem of $\hat{\mathbf{P}}$ can be solved using the gradient projection method, where λ_{nmd} are adjusted in the direction opposite to the gradient $\partial D^{(t_1,t_2)}(\boldsymbol{\lambda})/\partial\lambda_{nmd}$. The update is performed as

$$\lambda_{nmd}^{(t_1,t_2+1)} = \left[\lambda_{nmd}^{(t_1,t_2)} - \gamma \frac{\partial D^{(t_1,t_2)}(\boldsymbol{\lambda}^{(t_1,t_2)})}{\partial \lambda_{nmd}}\right]^+ \quad (3.23)$$

where

$$\frac{\partial D^{(t_1,t_2)}(\boldsymbol{\lambda}^{(t_1,t_2)})}{\partial \lambda_{nmd}} = \frac{-w_{nm}}{\sum_{d \in \mathcal{D}_{nm}} \lambda_{nmd}^{(t_1,t_2)}} + \mu_{nmd}^{(t_1,t_2)} \tag{3.24}$$

and $\gamma > 0$ is the step size.

Once the gradient projection algorithm has converged to its solution $\boldsymbol{\lambda}^{*(t_1,t_2)}$, we can update the p_{nm} as follows.

$$p_{nm}^{(t_1+1,t_2)} = p_{nm}^{(t_1,t_2)} + \alpha \sum_{(l,\mathcal{D}_{lk},j) \in \mathcal{I}} \lambda_{lkj}^{*(t_1,t_2)} \frac{\partial c_{lkj}(\mathbf{p}^{(t_1,t_2)})}{\partial p_{nm}} \tag{3.25}$$

where

$$\frac{\partial c_{lkj}}{\partial p_{nm}} = \begin{cases} \prod\limits_{\substack{i:j \in \mathcal{N}_i, \\ i \neq l}} (1-p_i), & (n, \mathcal{D}_{nm}, d) = (l, \mathcal{D}_{lk}, j) \\ -p_{lk} \prod\limits_{\substack{i:j \in \mathcal{N}_i, \\ i \neq l, i \neq j}} (1-p_i), & j = n, j \in \mathcal{D}_{lk} \\ -p_{lk} \prod\limits_{\substack{i:j \in \mathcal{N}_i, \\ i \neq l, i \neq n}} (1-p_i), & j \in \mathcal{N}_n \backslash \{n\}, j \in \mathcal{D}_{lk} \\ 0, & \text{else} \end{cases} \tag{3.26}$$

and $\alpha > 0$ is the step size. Once $\mathbf{p}$ has been updated, we again run the algorithm in (3.23) to converge on the optimal link prices, and repeat.

We summarize the proposed algorithm for guaranteed multicast throughput allocation in the following steps.

(1) Pick arbitrary initial values for the access probabilities with $0 < p_{nm} < 1$.

(2) Find $\boldsymbol{\lambda}^*$ using the algorithm described in (3.23) and (3.24).

(3) Update the access probabilities according to (3.25) and (3.26).

(4) Return to step (2) and repeat until the access probabilities p_{nm} have converged.

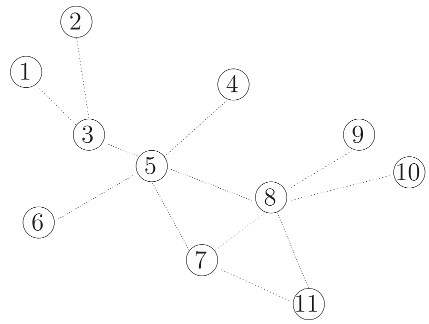

Figure 3.2: A simple example network to demonstrate our distributed throughput allocation algorithms for non-guaranteed and guaranteed multicast.

The convergence analysis of this algorithm follows directly from the proof of Theorem 1 in [31]. The analysis can be summarized as follows. Since the algorithm adjusts the access probabilities p_{nm} in the direction of the gradient, it yields a local optimal point for $\tilde{\mathbf{P}}$. This local optimal point for $\tilde{\mathbf{P}}$ is also a local optimal point for $\mathbf{P}$ since the two problems are equivalent. The analysis in [31] shows that the local optimal point for $\mathbf{P}$ is globally optimal.

3.5 An example network

We now consider an example network to demonstrate the non-guaranteed and guaranteed approaches to optimal throughput allocation for random access multicast. The example network we consider is shown in Fig. 3.2, where the numbers correspond to the node identities. The lines in the figure indicate interference; for simplicity, we assume that interference is symmetric, i.e., $k \in \mathcal{N}_n \iff n \in \mathcal{N}_k$. The multicast flows we consider are shown in Table 3.1, where nodes $3, 5$ and 8 act as source nodes and two multicast flows emanate from each source. We assume that multicast traffic traversing the center of the network is given higher priority; thus

Table 3.1: Multicast flows for the example network shown in Figure 3.2.

Source (n)	Tree (m)	Receivers
3	1	$\mathcal{D}_{31} = \{1, 2\}$
3	2	$\mathcal{D}_{32} = \{1, 2, 5\}$
5	1	$\mathcal{D}_{51} = \{3, 4\}$
5	2	$\mathcal{D}_{52} = \{6, 7, 8\}$
8	1	$\mathcal{D}_{81} = \{5, 7, 11\}$
8	2	$\mathcal{D}_{82} = \{9, 10\}$

in the examples we consider, we assign higher weights w_{nmd} and w_{nm} to multicast flows passing through node 5.

3.5.1 Non-guaranteed multicast

For non-guaranteed multicast, we assign a weight $w_{nmd} > 0$ to each link in the network and compute the optimal access probabilities according to (3.9). An example of the link weights, optimal access probabilities, and link throughputs μ_{nmd} as computed from (3.5) are shown in Table 3.2. Due to the large weights w_{nmd} associated with the multicast flows emanating from node 5, node 5 is assigned a large access probability with $p_5 > 0.9$. As a result, the value of the link throughput μ_{nmd} is smallest for links $(3, 2, 5)$, $(8, 1, 5)$, and $(8, 1, 7)$, where node 5 is either a receiver or causes interference. The value of the link throughput μ_{nmd} is largest for links on the edge of the network, which do not suffer interference.

Table 3.2: Optimal access probabilities and link throughput values for non-guaranteed multicast in the example network. A ” indicates that an entry is the same as the entry in the row above it.

Link	w_{nmd}	p_{nm}	μ_{nmd}
$(3, 1, 1)$	0.5	0.25	0.25
$(3, 1, 2)$	0.5	”	0.25
$(3, 2, 1)$	0.5	0.5	0.5
$(3, 2, 2)$	0.5	”	0.5
$(3, 2, 5)$	1.0	”	0.0154
$(5, 1, 3)$	2.0	0.4615	0.1154
$(5, 1, 4)$	1.0	”	0.4615
$(5, 2, 6)$	0.5	0.4615	0.4615
$(5, 2, 7)$	1.0	”	0.1846
$(5, 2, 8)$	1.5	”	0.1846
$(8, 1, 5)$	1.0	0.4	0.0077
$(8, 1, 7)$	0.5	”	0.0308
$(8, 1, 11)$	0.5	”	0.4
$(8, 2, 9)$	0.5	0.2	0.2
$(8, 2, 10)$	0.5	”	0.2

3.5.2 Guaranteed multicast

For guaranteed multicast, we assign a weight $w_{nm} > 0$ to each multicast tree and make use of the proposed algorithm to compute the random access probabilities. The link weights that we considered, along with the values of p_{nm} and μ_{nmd}^{min} that the algorithm reached at convergence, are shown in Table 3.3.

In implementing the algorithm, we set the step sizes at $\alpha = 5 \times 10^{-4}$ and $\gamma = 25$. We have assumed that the gradient projection algorithm has converged to its solution $\boldsymbol{\lambda}^{*(t_1,t_2)}$ once the variation in all λ_{nmd} values is less than 5×10^{-3}. With these parameters, we observed that the convergence of the gradient projection algorithm required at most 180 iterations, and typically required fewer than 20 iterations. As shown in Figs. 3.3 and 3.4, the algorithm to update p_{nm} converges on the optimal p_{nm} and μ_{nmd}^{min} for guaranteed multicast. In those two figures, the number of iterations (along the x-axis) refers to the number of updates to the values of p_{nm}, and we observe that at most 300 iterations are needed to reach convergence. For implementation in practical scenarios, the convergence speed can be adjusted by varying the step sizes, particularly by choosing the step size γ individually for each link $(n, \mathcal{D}_{nm}, d)$.

Once again, due to the large weights assigned to the multicast flows emanating from source 5, it is assigned a large access probability with $p_5 \approx 0.65$ and the guaranteed multicast throughput is largest for trees $(5, \mathcal{D}_{51})$ and $(5, \mathcal{D}_{52})$. The multicast flows on the edge of the network obtain a moderate throughput due to the lack of interference. The guaranteed throughput is smallest for flows $(3, \mathcal{D}_{32})$ and $(8, \mathcal{D}_{81})$

Table 3.3: Optimal access probabilities and minimum link throughput values for guaranteed multicast in the example network.

Tree	w_{nm}	p_{nm}	μ_{nmd}^{min}
$(3, \mathcal{D}_{31})$	1	0.0952	0.0952
$(3, \mathcal{D}_{32})$	2	0.3178	0.0718
$(5, \mathcal{D}_{51})$	3	0.2040	0.1198
$(5, \mathcal{D}_{52})$	3	0.4549	0.3012
$(8, \mathcal{D}_{81})$	2	0.2330	0.0466
$(8, \mathcal{D}_{82})$	1	0.1048	0.1048

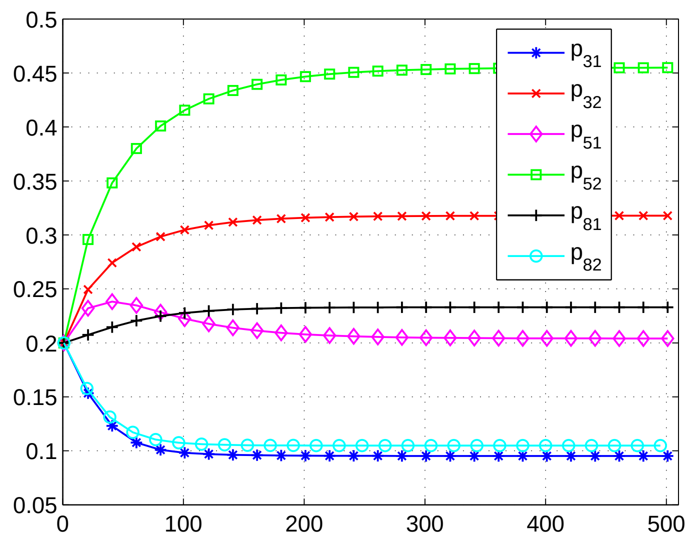

Figure 3.3: Values of p_{nm} versus iteration number for the proposed algorithm to compute random access probabilities for guaranteed multicast.

due to the moderate weight assigned to those flows coupled with the interference from node 5.

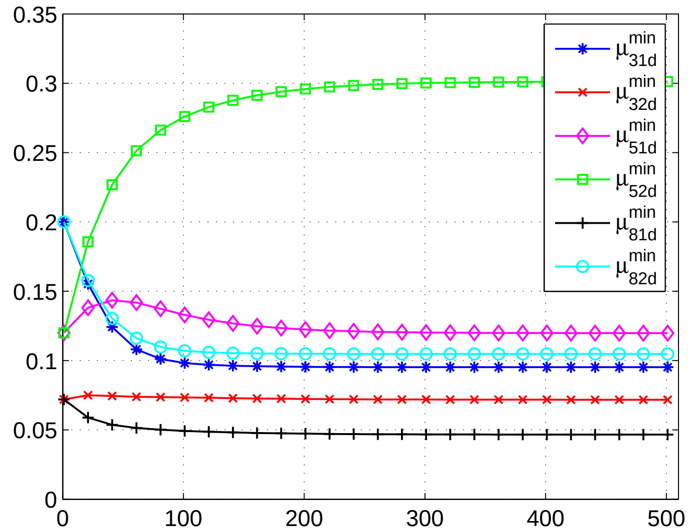

Figure 3.4: Values of μ_{nmd}^{min} versus iteration number for the proposed algorithm to compute random access probabilities for guaranteed multicast.

3.6 Discussion

In this chapter we treated the problem of multicast transmission in a random access network of general topology. We focused on the saturation throughput and proposed schemes to assign the random access probabilities to nodes in a wireless network in order to optimally allocate the *multicast* throughput. For applications where non-guaranteed multicast is sufficient, the optimal throughput allocation can be computed in a simple, distributed manner similar to the manner in which throughput is allocated for *unicast* transmission. If guaranteed multicast transmission is required, the throughput-optimal strategy involves random coding of packets and a distributed algorithm that converges to the optimal access probabilities.

For guaranteed multicast, we have assumed that random coding is performed with $K \to \infty$ in order to achieve the optimum throughput. However, coding with

$K \to \infty$ would yield an infinite delay, which is undesirable in practical systems. In the next chapter, we analyze the delay performance of a random coding scheme that adapts to the traffic load. The scheme we consider provides good performance in terms of both throughput and delay.

Chapter 4

Queueing delay

4.1 Background and model

In Chapters 2 and 3 we saw that random coding of packets can offer performance benefits in terms of multicast throughput. Given the well-documented tradeoff between the throughput and delay in a network, a natural question that arises is: does the throughput gain offered by random coding result in a penalty in terms of the delay? There has been relatively little attention given to this question. In [37], the authors examined the use of network coding for multicast transmission in a network and it was shown to offer improvements in delay performance. However, that work was carried out under the assumption that there is a fixed amount of data awaiting transmission at the source node, which means that queueing or waiting time at the source node is not considered.

In order to capture the effect on the delay performance of varying traffic load at the source node, it is useful to consider coding of packets that *randomly arrive* at the source node and analyze performance in a queueing framework. This has been explored in [38], where in each transmission opportunity, the source node sends a random linear combination of the packets queued in a finite-capacity buffer. Simulation studies are used in [38] to investigate the delay as a function of buffer size. Our intent is to develop analytical models and results on the delay performance of random coding, and this is the subject of the current chapter. In the following sections, the random linear coding scheme is analyzed as a discrete-time bulk-service queue. The bulk-service property of the queue captures the fact that packets are encoded, transmitted, and removed from the queue in groups. Bulk-service queueing is an appropriate model for performing coding of randomly arriving packets, and has been used in [39] to model forward error correction for transmission over a broadcast channel.

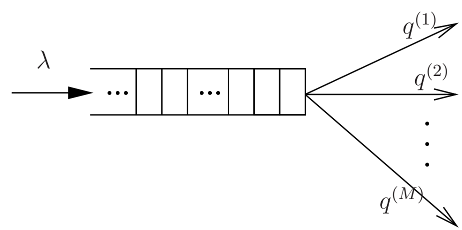

Figure 4.1: Packets arrive randomly to a single source node and are stored in an infinite-capacity buffer before multicast transmission to M destinations.

The system we consider is shown in Figure 4.1. Time is slotted; a slot corresponds to the time needed for a packet to be transmitted over the channel. A packet is a fixed-length vector of bits. *Data packets* arrive to a source node through a Bernoulli process with rate λ packets/slot. Thus λ represents the probability that a data packet arrives in a slot. Upon arrival, the data packet is placed in a buffer of infinite capacity in order to await transmission. The buffer forms a first-in-first-out (FIFO) queue. When a packet reaches the front of the queue, it is transmitted over the channel. The channel is a wireless (broadcast) medium, so a transmission has the potential to reach multiple destination nodes. There are M destination nodes and all of them must receive the data packets from the source node. The channel between the source and destination m is an independent erasure channel with reception probability in each slot given by

$$q^{(m)} = \Pr\{\text{coded packet received at destination } m\}, \quad 1 \leq m \leq M. \tag{4.1}$$

We assume that each source-destination pair has an independent, error-free feedback channel over which the destination node can send acknowledgement messages. We model the system shown in Figure 4.1 as a discrete-time queue. Much of the analysis is carried out through use of the probability generating function (pgf). For a non-negative discrete random variable with probability mass function (pmf) $f(n)$, the pgf, denoted $F(z)$, is the z-transform of $f(n)$ and is given by $F(z) = \sum_{n\geq 0} f(n)z^n$.

4.2 Retransmissions

We first present delay results for a scheme in which a data packet is retransmitted over the channel until received at all M destinations. As soon as a destination has received the packet under transmission, it sends an acknowledgement message to the source node. Transmission is complete when the source node has received acknowledgement messages from all M destinations. Let $T^{(m)}$ denote the number of slots needed for destination m to successfully receive the packet. According to the erasure channel model, $T^{(m)}$ is geometrically distributed with parameter $q^{(m)}$, i.e., $\Pr(T^{(m)} = t) = q^{(m)}(1 - q^{(m)})^{t-1}$. The time needed for all M destinations to receive the packet is denoted T and is given by $T = \max_m T^{(m)}$. In analyzing the average delay, we will be interested in the first and second moments of T. The problem of finding the moments of the maximum of independent, differently distributed geo-

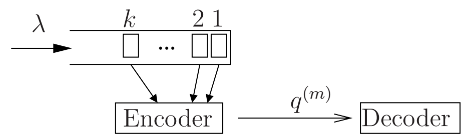

Figure 4.2: Random linear coding of randomly arriving packets. The first k packets at the front of the queue are passed to the encoder for encoding and transmission.

metric random variables can be solved numerically. (For approximations, we refer the reader to [40].) A numerical solution can be found by noting that the cdf of T is given as

$$\Pr(T \leq t) = \Pr(\max_m T^{(m)} \leq t) = \Pr(T^{(1)} \leq t) \times \Pr(T^{(2)} \leq t) \times \ldots \times \Pr(T^{(M)} \leq t). \tag{4.2}$$

The first and second moments, $E[T]$ and $E[T^2]$, can be found numerically using (4.2).

According to the Bernoulli arrival process and the distribution of T, the appropriate queueing model is a $Geo/Geo/1$ queue, for which a complete analysis is available in [41]. In particular, the queueing stability condition is given by $\rho_1 = \lambda E[T] < 1$, or equivalently, $\lambda < E[T]$. Additionally, the average delay (waiting time in the queue plus service time), which we denote D_1, is given as follows [41].

$$D_1 = E[T] + \frac{\lambda E[T^2] - \rho_1}{2(1 - \rho_1)}. \tag{4.3}$$

4.3 Random linear coding

We now analyze the random linear coding scheme depicted in Fig. 4.2 to determine the queueing delay involved. Let k denote the number of packets at the front of the queue that are simultaneously passed to the encoder. Random linear coding is performed as described in Chapter 1, where each of the k data packets are added into the modulo-2 sum with probability 1/2. The source node generates a new random linear combination in each time slot and transmits it over the channel. Let random variable $N_k^{(m)}$ denote the number of coded packets needed at destination m so that decoding can take place, i.e., so that there are exactly k linearly independent coded packets among the $N_k^{(m)}$. As the random linear combinations are generated by the same random distribution, $N_k^{(m)}$ are identically distributed with respect to m, and we refer to this commonly-distributed random variable as N_k. Although $N_k^{(m)}$ follow the same distribution, each destination may receive a different set of coded packets according to the realization of their independent channels. Because the coded packets are generated and transmitted by the same source to all M destinations, $N_k^{(m)}$ are correlated with respect to m. When a destination node has received N_k coded packets, decoding is performed to uncover the original k data packets. We assume that decoding occurs instantaneously and simultaneously on the k data packets. Subsequently, an error-free acknowledgement message is instantaneously sent from the destination to the source. Once the source has received an acknowledgement message from all M destinations, the service of the k packets is complete and they are removed from the front of the queue. The source then proceeds with encoding and transmission of the next set of data packets in the front of its buffer.

For random linear coding of randomly arriving packets, there are multiple strategies which might be employed. One strategy is to fix the number of packets used in encoding, or to fix the value of k as $k = K$. In this strategy, we enforce encoding over groups of *exactly* K, meaning that if there are fewer than K packets in the queue, then those packets must wait for more packets to arrive before encoding and transmission can begin. This strategy naturally leads to a delay penalty for lightly loaded systems in which packets arrive infrequently (i.e., small λ). Alternatively, the coding scheme could adapt to the amount of traffic buffered at the transmitting node by accepting k packets into the encoder when the channel becomes free, where k is a variable taking values between 1 and the upper limit of K. This type of strategy is proposed and investigated in [38]. In our work, we analyze both of these strategies using the model of a bulk-service queue. First, however, we analyze the pgf of the service time for random linear coding over k packets, which is used in the analysis for both strategies.

4.3.1 Service time

Let $\widetilde{T}_k^{(m)}$ denote the service time to destination m for random linear coding of k data packets, and let $B_{k,m}(z)$ denote the corresponding pgf. The service time constitutes the total number of slots that elapse from the initiation of service, when the k data packets are passed to the encoder to begin encoding and transmission, to the time that an acknowledgement message is sent by destination m. The function $B_{k,m}(z)$ will be given by a composite function of the distribution for success on the channel and the distribution of N_k. First, consider the time that elapses between two consecutive successful receptions at destination m of coded packets. According to our channel model, this time will be geometrically distributed with success probability $q^{(m)}$ in each slot. The pgf for this geometric random variable, which we denote $\Gamma_m(z)$, is given by

$$\Gamma_m(z) = \frac{q^{(m)}z}{1 - z(1 - q^{(m)})}. \tag{4.4}$$

The time periods that elapse between reception at destination m of consecutive coded packets will be independent, identically distributed according to $\Gamma_m(z)$. Since the destination must receive N_k such coded packets, the service time will be given by the sum of N_k independent random variables distributed according to $\Gamma_m(z)$. Letting $F_{N_k}(z)$ denote the pgf of N_k when k data packets are used in encoding, the pgf $B_{k,m}(z)$ can be shown (see, e.g., [11]) to be given by

$$B_{k,m}(z) = F_{N_k}(\Gamma_m(z)). \tag{4.5}$$

The distribution of N_k reflects the random linear coding, and as described in Chapter 1, is given by the probability that a random u-ary matrix is full-rank. Let $F_{N_k}(n)$ and $f_{N_k}(n)$ denote the cdf and pdf, respectively, of N_k. We have

$$F_{N_k}(n) = \begin{cases} \prod_{i=0}^{k-1}(1 - u^{-n+i}), & n \geq k \\ 0, & n < k \end{cases} \tag{4.6}$$

and

$$f_{N_k}(n) = F_{N_k}(n) - F_{N_k}(n-1). \tag{4.7}$$

We are now prepared to write an expression for the service time pgf $B_{k,m}(z)$ for random linear coding. Using the expression in (4.5), we can write $B_{k,m}(z)$ as follows.

$$B_{k,m}(z) = \sum_{n=k}^{\infty} f_{N_k}(n) \left(\frac{q^{(m)}z}{1 - z(1 - q^{(m)})} \right)^n. \tag{4.8}$$

Note that by evaluating $\frac{d}{dz}B_{k,m}(z)|_{z=1}$ from the above expression, we can obtain the expected service time to destination m for a group of k packets, which is $E[N_k]/q^{(m)}$.

4.3.2 Fixed expected coding rate

We first analyze a scheme whereby the number of data packets involved in random linear coding is a fixed parameter $k = K$. If there are fewer than K data packets in the buffer, then those packets must await the arrival of additional packets before encoding and transmission can begin. Once there are K data packets in the buffer, the first K packets are simultaneously passed to the encoder. We analyze this scheme as a *bulk-service queue with fixed capacity*, where the term "fixed capacity" refers to the fact that the service time consists of a single distribution corresponding to $k = K$.

This queueing system can be analyzed using the *method of stages* approach described in [11]. In this approach, the arrival process is said to involve K stages, where each stage corresponds to the arrival of one of the K packets involved in random linear coding. This approach leads to the Erlangian distribution [11] for continuous-time queues with an exponential distribution on the interarrival times for individual packets.

As stated previously, data packets arrive to the source according to a Bernoulli process with rate λ packets/slot. Accordingly, the time that elapses between arrivals of *individual* data packets is geometrically distributed. Let X_i denote the time in slots between the arrival of the $i - 1^{st}$ and the i^{th} packet in the buffer. The pmf of X_i is given by

$$a_i(x) = \Pr\{X_i = x\} = \lambda(1 - \lambda)^{x-1}, \quad x \geq 1 \tag{4.9}$$

and the pgf is given by

$$A_i(z) = \frac{\lambda z}{1 - z(1 - \lambda)}. \tag{4.10}$$

In applying the method of stages, we let X denote the *effective interarrival time*, which is essentially the time in slots between the arrival of every K^{th} packet. Thus X is given by the sum of K iid random variables distributed as X_i. The pgf of X, which we denote $A(z)$, can be expressed as

$$A(z) = (A_i(z))^K = \left(\frac{\lambda z}{1 - z(1 - \lambda)}\right)^K. \tag{4.11}$$

By taking derivatives of $A(z)$ with respect to z and evaluating at $z = 1$, we can find the mean and variance of the effective interarrival time as given below.

$$E[X] = \frac{K}{\lambda}, \quad \sigma_a^2 = Var[X] = \frac{K(1 - \lambda)}{\lambda^2} \tag{4.12}$$

The service process characterizes the time that elapses between the K^{th} data packet entering the encoder and receipt at the source of acknowledgement messages from all M destinations. Let $\widetilde{T}_K$ denote the service time, or the number of slots needed for all M destinations to receive N_K coded packets. Furthermore, let $\widetilde{T}_K^{(m)}$ denote the number of slots needed for destination m, $m = 1, 2, \ldots, M$, to receive N_K coded packets. Then we have

$$\widetilde{T}_K = \max_m \widetilde{T}_K^{(m)}. \tag{4.13}$$

Equation (4.13) accounts for the multicast nature of the transmission from the source. As shown in the previous section, the distribution of $\widetilde{T}_K^{(m)}$ is given by the pgf $B_{K,m}(z)$ in Eqn. (4.8). The pmf of $\widetilde{T}_K^{(m)}$, which we denote $b_{K,m}(t)$, is found by taking an inverse z-transform and is given by

$$b_{K,m}(t) = \begin{cases} \sum_{n=K}^{\infty} f_{N_k}(n)\alpha_n (q^{(m)})^n (1-q^{(m)})^{t-n}, & t \geq K \\ 0, & t < K \end{cases} \tag{4.14}$$

where

$$\alpha_n = \frac{(t-1)(t-2)\ldots(t-(n-1))}{(n-1)!}. \tag{4.15}$$

We will make use of $b_{K,m}(t)$ as shown above in our numerical computations. Finally, we can approximate the distribution of $\widetilde{T}_K = \max_m \widetilde{T}_K^{(m)}$ using the distribution of $\widetilde{T}_K^{(m)}$. Letting $B_K(t)$ denote the cdf of $\widetilde{T}_K$, we have

$$B_K(t) \approx \Pr\{\widetilde{T}_K^{(1)} \leq t\} \times \Pr\{\widetilde{T}_K^{(2)} \leq t\} \times \ldots \times \Pr\{\widetilde{T}_K^{(M)} \leq t\}. \tag{4.16}$$

The above expression is an approximation, and not an equality, since $T_K^{(m)}$ are not independent - due to the correlation of $N_k^{(m)}$ with respect to m. The mean and variance of $\widetilde{T}_K$, which we denote $E[\widetilde{T}_K]$ and σ_b^2 respectively, can be approximated from $B_K(t)$ as expressed above and are used in analyzing the stability and delay.

We next consider the conditions for stability of the queue, or more formally, the conditions for which the Markov chain representing the waiting time in the queue is ergodic. We define the traffic intensity ρ_2 as

$$\rho_2 = \frac{\lambda E[\widetilde{T}_K]}{K}. \tag{4.17}$$

For a standard queue, the queue is stable if and only if $\rho_2 < 1$ [11]. If this condition is not satisfied, then the delay will grow without bound.

In examining the delay, we note that the interarrival and service time distributions in the queueing model developed above do not allow for the application of standard delay results, such as the delay for the $M/G/1$ queue [11]. Instead, we will make an analytical approximation to the delay based on the mean and variances of the interarrival and service times. In doing so, we note that the delay will consist of two terms: the service time and the waiting time. The expected service time is given by $E[\widetilde{T}_K]$ and is constant over all arrival rates λ. The waiting time of a packet can be attributed to two phenomena: the time spent waiting to reach the front of the queue while other groups of K packets are being served, and the time spent waiting for additional packets to arrive to form a group of K packets for encoding and transmission.

For the time spent waiting to reach the front of the queue, we use the *heavy traffic approximation* provided for continuous-time systems, which is given in [42] by

$$\frac{\sigma_a^2 + \sigma_b^2}{2E[X](1-\rho_2)}. \tag{4.18}$$

This quantity will be relatively small for small values of λ but will be the dominant term as λ increases, approaching infinity as $\rho_2 \to 1$. For the time spent waiting for additional packets to arrive, we note that on average, a given packet will need to wait for $(K-1)/2$ additional packets to arrive, which will require a waiting time of $\frac{K-1}{2\lambda}$. This quantity will approach infinity as $\lambda \to 0$ but will diminish as λ increases. Our delay approximation assumes that the waiting time will be given by the maximum of the two quantities described above. Thus, our approximation for the average delay is given by

$$D_2 \approx E[\widetilde{T}_K] + \max\left(\frac{\sigma_a^2 + \sigma_b^2}{2E[X](1-\rho_2)}, \frac{K-1}{2\lambda}\right). \tag{4.19}$$

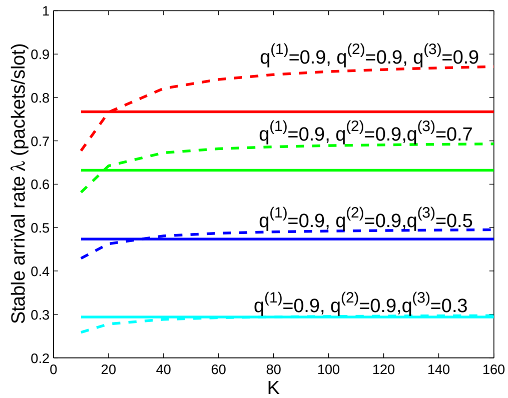

Figure 4.3: The maximum stable arrival rate $K/E[\widetilde{T}_K]$ for random linear coding over K packets (dashed line) and retransmissions (solid line) for four different channel models and transmission to M=3 destination nodes.

We now present some numerical examples of throughput and delay performance resulting from this bulk-queueing model for random linear coding. For these numerical results we have computed probability distributions, including $f_{N_K}(n)$ and $b_{K,m}(t)$, to within 10^{-5} of the total probability mass.

The maximum stable arrival rates for random linear coding and retransmissions are compared in Fig. 4.3 for $M = 3$ and four different sets of reception probabilities $(q^{(1)}, q^{(2)}, q^{(3)})$. The maximum stable arrival rate corresponds to the condition $\rho_2 < 1$ and is given by $K/E[\widetilde{T}_K]$. These results indicate that when one of the destinations has a poor channel ($q^{(3)}$=0.3) there is little or no benefit of random linear coding over retransmissions in terms of the stable arrival rate. Also, even for relatively good channels, the random linear coding scheme does not unconditionally outperform the retransmission scheme: if the value of K is not sufficiently large, then the retransmission scheme can support higher average arrival rates.

The effect of the number of destination nodes M on the stable arrival rate is

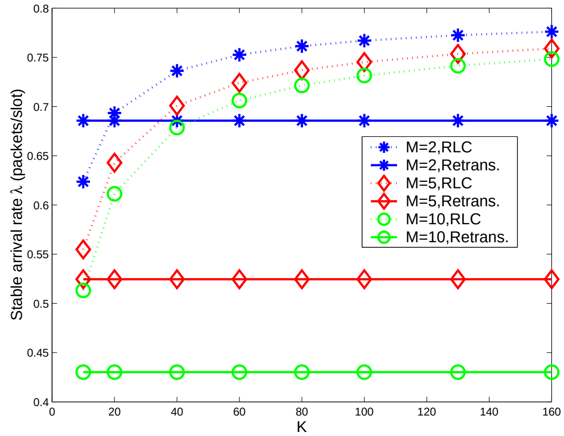

Figure 4.4: The maximum stable arrival rates for random linear coding over K packets (dashed line) and retransmissions (solid line) for channels with $q^{(m)} = 0.8, m = 1, 2, \ldots, M$ and different numbers of destinations.

shown in Fig. 4.4. These results are computed for relatively good channels with $q^{(m)} = 0.8$, $m = 1, 2, \ldots, M$, and random linear coding nearly always outperforms retransmissions. The arrival rate that the source can tolerate while maintaining a stable queue is higher when the source multicasts to fewer destinations. Additionally, we observe that random linear coding provides larger gains over retransmissions when the source multicasts to more destination nodes.

The delay performance of random linear coding is compared with retransmissions in Fig. 4.5 for a channel with $(q^{(1)}, q^{(2)}, q^{(3)}) = (0.9, 0.9, 0.9)$. In addition to computing D_2 and D_1, we have performed and plotted the results of a Monte Carlo simulation of the random linear coding scheme over 20,000 slots in order to compare the outcome with our approximation. Our approximation closely matches the simulation results for small values of λ, however, the two outcomes differ when the queues saturate. We note that, in contrast to typical queueing delay results, the delay for random linear coding is not a monotonic increasing function. The behavior

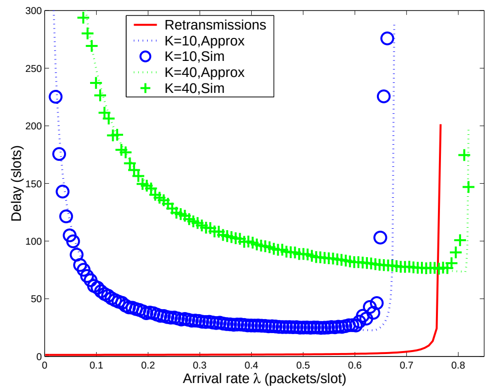

Figure 4.5: Delay versus arrival rate λ for retransmissions (solid line), the random linear coding approximation (dotted line), and the random linear coding simulation (o,+). Random linear coding is performed over $K = 10$ (o) and $K = 40$ (+) packets. The results are for transmission to M=3 destinations with $(q^{(1)}, q^{(2)}, q^{(3)}) = (0.9, 0.9, 0.9)$.

of the random linear coding delay for $\lambda \to 0$ is due to the time spent waiting for additional packets to arrive in order to form a group of K packets for encoding and transmission.

For random linear coding over $K = 10$ packets, the delay is higher than the retransmission delay for all values of λ. The fact that the random linear coding scheme with $K = 10$ saturates sooner than the retransmission scheme reflects the result shown in Figure 4.3. Thus, for $M = 3$, $K = 10$, and this particular channel model, random linear coding offers no benefit over retransmissions in terms of neither stable throughput nor delay. The random linear coding scheme over $K = 40$ packets can tolerate higher values of λ than the retransmissions scheme, however, for small values of λ, the random linear coding scheme with $K = 40$ provides significantly higher delay than retransmissions. This result points to a tradeoff between the stable throughput and the delay. Clearly there is a delay penalty for lightly loaded systems due to fixing the expected coding rate.

4.3.3 Variable expected coding rate adapted to traffic load

We now analyze a scheme whereby the expected coding rate is a variable that adapts the traffic load. In other words, k, the number of packets involved in forming a random linear combination, is a variable that takes values between 1 and K. The chosen value of k is dependent upon the number of packets waiting in the queue at the time when the service of a previous group of packets is completed. Thus, if there is one packet in the queue when the channel becomes free, then $k = 1$ and one packet is transmitted over the channel. If there are two packets in the queue when the channel becomes free, then $k = 2$ and encoded versions of two packets are transmitted over the channel, and so on. If there are K or more packets in the queue when the channel becomes free, then the K packets at the front of the queue are used in encoding and transmission. In the queueing literature, this policy is referred to as a *bulk-service queue with variable capacity*, where the term "variable-capacity" refers to the property that the service time distribution varies according to the value of k. For analytical tractability, we analyze this scheme in the simplified case of $M = 1$, or unicast transmission. We denote the channel reception probability $q^{(1)}$ as q.

For $k = 1$ packet in service, encoding is not performed, and instead the packet is retransmitted until received. In this case the service time distribution is geometric with success probability q in each slot. The service time pgf is given by

$$B_1(z) = \frac{qz}{1-z(1-q)}. \tag{4.20}$$

For $2 \leq k \leq K$, the encoder forms random linear combinations of the k packets.

The service time distribution follows the pgf $B_{k,m}(z)$ in Eqn. 4.8 for $2 \le k \le K$. Since we now consider only $M = 1$ destination node, we simply refer to this pgf as $B_k(z)$.

The pmf $f_{N_k}(n)$ appears in the expression for $B_k(z)$, and the finite product in (4.6) is difficult to deal with analytically. We would instead like to express the finite product as a finite sum. For this purpose, we can view (4.6) as a polynomial and apply Viète's formulas to obtain

$$\prod_{i=0}^{k-1}(1-u^{-n+i}) = 1 + \sum_{j=1}^{k}(-1)^j \alpha(j,k) u^{-nj}, \tag{4.21}$$

where, for $1 \le j \le k-1$,

$$\alpha(j,k) = \sum_{i_1=0}^{k-j} u^{i_1} \sum_{i_2=i_1+1}^{k-j+1} u^{i_2} \dots \sum_{i_j=i_{j-1}+1}^{k-1} u^{i_j}, \tag{4.22}$$

and

$$\alpha(k,k) = u^{\frac{1}{2}k(k-1)}. \tag{4.23}$$

The pmf of N_k can be written as follows.

$$f_{N_k}(n) = \begin{cases} 0, & n < k \\ 1 + \sum\limits_{j=1}^{k}(-1)^j \alpha(j,k) u^{-kj}, & n = k \\ \sum\limits_{j=1}^{k}(-1)^j \alpha(j,k) u^{-nj}(1-u^j), & n > k \end{cases} \tag{4.24}$$

From this expression for $f_{N_k}(n)$, we can write

$$\begin{aligned} B_k(z) = & \left(\frac{qz}{1-z(1-q)}\right)^k \left(1+\sum_{j=1}^{k}(-1)^j \alpha(j,k) u^{-jk}\right) \\ & + \left[\left(\frac{qz}{1-z(1-q)}\right)^{k+1} \sum_{j=1}^{k}(-1)^j \alpha(j,k) u^{-j(k+1)} \frac{(1-u^j)(1-z(1-q))}{1-z(1-q+qu^{-j})}\right]. \end{aligned} \tag{4.25}$$

We now apply the techniques in [43] to derive the distribution of the number of packets in the system in the steady-state. Let random variable S_t denote the number of packets in the system immediately after the completion of service and departure of the t^{th} group of packets. Precisely, we consider the number in the system at the boundary of a slot where a service of a group of packets has just been completed. The process $\{S_t, t \geq 0\}$ forms an irreducible, aperiodic Markov chain. Let P_k denote the steady-state probability of k packets in the system immediately after a departure instant.

$$P_k = \lim_{t \to \infty} \Pr(S_t = k), \quad k = 0, 1, 2, \ldots \tag{4.26}$$

The existence of the limit corresponds to stability of the queue, which as discussed in [43], holds as long as the arrival rate λ is less than K times the service rate for a group of K packets. In our random linear coding system this corresponds to $\lambda < qK/E[N_K]$. Note that this condition is equivalent to $\rho_2 < 1$ for $M = 1$, where ρ_2 is the traffic intensity for random linear coding with fixed expected coding rate, as defined in Eqn. (4.17). As such, the scheme currently under consideration, in which the coding rate is adapted to traffic load, supports the same maximum stable arrival rate as the scheme for which the expected coding rate is fixed. Then the results in Figs. 4.3 and 4.4 apply to the current scheme as well.

Assuming that the stability condition holds, we derive the pgf of P_k, which we denote $P(z)$. Since S_t forms a Markov chain, we can use the transition probabilities of the Markov chain to find $P(z)$. The transition probabilities will depend on the length of the service period, given by $B_k(z)$, as well as the number of arrivals that occur during the service period. We define the random variable $\beta_{k,j}$, $1 \leq k \leq K$ as follows.

$$\beta_{k,j} = \Pr(j \text{ packets arrive during service period} \sim B_k(z)) \tag{4.27}$$

Furthermore, we make the assumption that if there are 0 packets in the queue when the channel becomes free, then there is a pause in service, where the duration of the pause is a random variable distributed according to $B_1(z)$. Following the pause, the channel becomes free again, and the next service can be initiated. Thus we define a service time distribution of zero packets as $B_0(z) = B_1(z)$. This assumption ensures the Markovian property of S_t and allows us to easily write the transition probabilities, which we denote P_{ij}, as follows.

$$P_{ij} = \begin{cases} \beta_{1,j}, & i = 0, 1 \\ \beta_{2,j}, & i = 2 \\ \vdots & \vdots \\ \beta_{K-1,j}, & i = K-1 \\ \beta_{K,j-i+K}, & K \leq i \leq j+K \\ 0, & i > j+K \end{cases} \tag{4.28}$$

We can derive the pgf of the number in the system as $P(z) = \sum_{j=0}^{\infty} P_j z^j$ by applying the balance equations $P_j = \sum_{i=0}^{\infty} P_i P_{ij}$. In doing so, we encounter the need to find the pgf of $\beta_{k,j}$. This is done by conditioning on the length of the service time

$\widetilde{T}_k$ as follows.

$$\begin{aligned}\beta_{k,j} &= \sum_{t=1}^{\infty} \Pr(j \text{ packets arrive in } t \text{ slots}) \Pr(\widetilde{T}_k = t) \\ &\overset{(a)}{=} \sum_{t=1}^{\infty} \binom{t}{j} \lambda^j (1-\lambda)^{t-j} \Pr(\widetilde{T}_k = t), \quad j \leq t\end{aligned}$$

In the expression above, (a) follows from the assumption that the arrivals follow a Bernoulli process of rate λ. The pgf of $\beta_{k,j}$ is given by

$$\sum_{j=0}^{\infty} \beta_{k,j} z^j = B_k(\lambda z + 1 - \lambda) \tag{4.29}$$

Thus $B_k(\lambda z + 1 - \lambda)$ expresses the distribution of the number of arrivals that occur during the service of a group of k packets. The distribution of the number in the system immediately after a departure is expressed as follows.

$$P(z) = \frac{\sum_{k=0}^{K-1} P_k \left(z^K B_k(\lambda z+1-\lambda) - z^k B_K(\lambda z+1-\lambda)\right)}{z^K - B_K(\lambda z+1-\lambda)}. \tag{4.30}$$

The expression (4.30) contains K unknown constants P_k, $0 \leq k \leq K-1$. These constants can be found in the following manner. Let the numerator and denominator in (4.30) be given by $N(z)$ and $D(z)$, i.e., $P(z) = N(z)/D(z)$. It can be shown by Rouché's theorem [43] that the denominator $D(z)$ has $K-1$ zeros *inside* the unit circle. Let z_j, $j = 1, 2, \ldots K-1$ denote these zeros, i.e., $D(z_j) = 0$. The zeros z_j can be computed numerically. Next, since the pgf $P(z)$ is an analytic function, the numerator $N(z)$ evaluated at z_j must also be equal to zero. We can set up a system of $K-1$ equations in the K unknowns P_k by the condition $N(z_j) = 0$, $j = 1, 2, \ldots K-1$ [44]. An additional equation is given by the condition $P(1) = 1$, which holds since $P(z)$ is a pgf. Thus we have a system of K equations in K unknowns and can solve for the constants P_k numerically.

We can obtain the expected value of the number in the system immediately after a departure by evaluating $\frac{d}{dz}P(z)|_{z=1}$. This results in the expression

$$\frac{f_1 g_2 - f_2 g_1}{2f_1^2} \tag{4.31}$$

where

$$\begin{aligned} f_1 &= K - \lambda E[\widetilde{T}_K] &(4.32)\\ f_2 &= K(K-1) - \lambda^2 E[\widetilde{T}_K(\widetilde{T}_K - 1)] &(4.33)\\ g_1 &= \sum_{k=0}^{K-1} P_k \left(K + \lambda E[\widetilde{T}_k] - k - \lambda E[\widetilde{T}_K]\right) &(4.34)\end{aligned}$$

$$g_2 = \sum_{k=0}^{K-1} P_k \Big(K(K-1) + 2K\lambda E[\widetilde{T}_k] + \lambda^2 E[\widetilde{T}_k(\widetilde{T}_k - 1)] - k(k-1)$$
$$-2k\lambda E[\widetilde{T}_K] - \lambda^2 E[\widetilde{T}_K(\widetilde{T}_K - 1)]\Big), \tag{4.35}$$

and the moments of $\widetilde{T}_k$ can be obtained from (4.25) and (4.20). Note that the distribution $P(z)$ provides the number in the system *immediately after a departure* and is not necessarily the same as the number in the system *at an arbitrary point in time.* However, we use the expected value in (4.31) as an *approximation* to the number in the system at an arbitrary point in time. We then apply Little's result [11] to obtain an approximation for the average time a packet spends in the system, which we refer to as average delay D_3. The average delay is given by

$$D_3 = \frac{1}{\lambda}\frac{f_1 g_2 - f_2 g_1}{2f_1^2}. \tag{4.36}$$

The average delay D_3 as given in (4.36) is computed and plotted as a function of arrival rate for two different values of $q = q^{(1)}$ in Figs. 4.6 and 4.7. As $\lambda \to 0$,

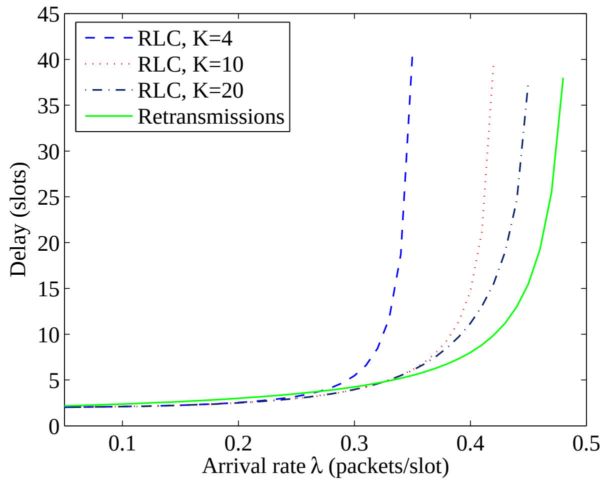

Figure 4.6: Average delay D_3 versus arrival rate λ for $M = 1$ and $q^{(1)} = 0.5$.

the system is nearly empty. The few packets that arrive do not experience any waiting time in the queue and are immediately transmitted over the channel with a service time distributed as $B_1(z)$ for both the random linear coding and retransmission schemes. In this case, the delay for all schemes is $1/q^{(1)}$. As the arrival rate λ increases, the random linear coding scheme provides the same delay as the retransmission scheme, and random linear coding performs the same regardless of the value of K. This is due to the fact that because the system is lightly loaded, the value of k rarely takes its maximum value K. In Fig. 4.7, at values around $\lambda = 0.5$, the retransmission scheme appears to suffer higher delay than the random linear coding schemes. We believe that this behavior is due to the fact that D_3 is an *approximation* and tends to underestimate the average delay.

As λ continues to increase, we see that the random linear coding scheme saturates sooner than the retransmission scheme. This is very different from the performance shown previously, where random linear coding provides higher stable arrival

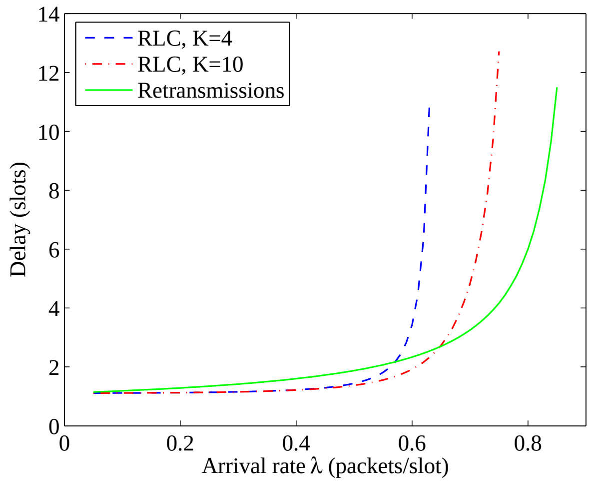

Figure 4.7: Average delay D_3 versus arrival rate λ for $M = 1$ and $q^{(1)} = 0.9$.

rates than retransmissions, and is due to the fact that we consider unicast transmission, $M = 1$. As K increases, the maximum stable arrival rate of random linear coding approaches the maximum stable arrival rate of retransmissions. This is due to the fact that as K increases, $K/E[N_K] \to 1$, so the stability condition for random linear coding, given by $\lambda < q^{(1)}K/E[N_K]$, approaches the stability condition for retransmissions, which is given by $\lambda < q^{(1)}$.

4.4 Discussion

In this chapter we examined the delay involved in random linear coding of randomly arriving data packets. We placed random linear coding into a queueing framework by considering bulk-service queues, where packets in the queue are serviced and removed from the queue in groups. This queueing framework naturally leads to a characterization of the delay.

We considered two different strategies for random linear coding. In the first strategy, the number of packets used in random linear coding, or the expected coding rate, is a fixed parameter, meaning that an additional delay may be incurred in waiting for enough packets to arrive before encoding and transmission can begin. In the second strategy, the number of packets used in random linear coding is a variable, and adapts to the number of packets waiting in the queue when the encoder and channel become free. If this adaptive coding scheme uses parameter K as an upper bound on the number of packets used in coding, and that same parameter K is used in the fixed expected coding rate scheme, then the two schemes support the same stable throughput. In terms of delay, the fixed expected coding rate scheme incurs a significant delay penalty for lightly-loaded systems. However, the adaptive coding rate scheme overcomes this difficulty, and the delay performance of random linear coding is superior to a retransmission scheme.

In conclusion, the throughput benefits offered by random linear coding *can* lead to a degradation in terms of delay, and this degradation will result if the expected coding rate is fixed. However, if the coding rate is adapted to the traffic load, then random linear coding offers benefits in terms of both throughput *and* delay. A primary consideration in designing practical systems must be the additional complexity and/or memory requirements for an adaptive coding scheme, which must be weighed with the delay benefits it may offer.

Chapter 5

Packet length, overhead, and effects of the wireless medium

5.1 Background

In previous chapters we focused on the throughput and delay performance of random coding while treating the wireless channel as an erasure channel with fixed, arbitrary erasure probability. We treated a data packet as a fixed-length sequence of bits without regard for its information content. Additionally, we have disregarded in our analysis the need to transmit overhead information containing the coefficients of the random code. In this chapter we revisit the performance of random coding for multicast transmission and specifically account for the packet length, overhead, and physical effects of the wireless channel.

Part of the motivation for the problem we describe below is to develop a wireless channel model within the framework of random coding of packets in order to better characterize performance. The notion of cross-layer design in wireless networks emphasizes this approach. Indeed, the physical behavior of the wireless channel has profound effects on data traversing the network in the form of packets. The analysis we present below demonstrates specific ways in which the physical medium impacts random coding of packets.

Our motivation also stems from interest in applying to wireless networks the technique of network coding, where packets are coded at bottleneck links in a multihop network [1, 2]. The random linear coding we consider throughout this thesis can be viewed as a localized (single-hop) version of random network coding [8]. It has been shown that by applying network coding for multicast transmission in a wired network, it is possible to achieve the maximum flow capacity in the limit as the symbol alphabet size (or field size) approaches infinity [1, 2]. For random network coding, the overhead inherent in communicating the coding coefficients becomes negligible only as the number of data symbols in the packet grows large. The "length" or information content of a transmitted packet (i.e., the number of bits conveyed by the packet) depends on both the symbol alphabet and the number of symbols per packet. From the previous works listed above, it is clear that transmitting sufficiently long packets is crucial to approaching the maximum flow capacity and to allowing random coding to operate with low overhead.

However, the long packets needed for network coding are more susceptible to noise, interference, congestion, and other adverse channel effects. In particular, we cite the following reasons that the erasure probability will increase with the length of the packet.

- If we try to fit more bits into the channel using modulation, then for fixed transmission power, points in the signal constellation will move closer together, making them more susceptible to noise and errors more likely.
- If we try to fit more bits into the channel by decreasing symbol duration, then we are constrained by bandwidth, a carefully-controlled resource in wireless systems.

- Longer packets take up more space in memory, leading to an increase in the chance of buffer overflows and dropped packets.

In this chapter, we model the erasure probability as a function of the packet length and investigate the implications on performance of random coding for multicast. Our emphasis will be on a wireless channel with a fixed bandwidth, for which we will associate erasure probability with the probability of symbol error for a given modulation scheme. We also scale the throughput to account for the overhead needed to perform random coding. We investigate the effects of packet length and overhead for two different channel models: a time-invariant channel and a time-varying channel.

5.2 Model

We consider the following setting. One source node will multicast data to M destination nodes. The channel from the source node is identically distributed (but not necessarily independent) with respect to the destination nodes. The channel is slotted but otherwise undefined for now; we will consider a time-invariant and time-varying channel model, which are clearly defined in the following sections. The source node has K units of information $\{s_1, s_2, \ldots, s_K\}$ that it wants to transmit. We will refer to each of these information units as data packets and let each packet be given by a vector of n u-ary symbols, where u is the symbol alphabet (the size of a finite field) and n is the number of symbols per packet. We consider values of u which are powers of 2. The length, or information content, of a packet is given by

$n \log_2 u$ bits.

The source generates random linear combinations by forming the sum $\sum_{i=1}^{K} a_i s_i$, where a_i are chosen randomly and uniformly from the set $\{0, 1, 2, \ldots, u-1\}$ and $\sum$ denote addition in the finite field $\mathbb{F}_u$. Note that the resulting random linear combination is a packet with length $n \log_2 u$ bits. When random linear combinations are formed, if the coefficients are all zero, i.e., if $a_i = 0$ for all $i = 1, \ldots, K$, then the source throws out that realization of a_i and generates a new realization. With each random linear combination transmitted, the source appends a packet header identifying a_i, which requires an additional $K \log_2 u$ bits of overhead with every $n \log_2 u$ bits of data transmitted. Once a random linear combination is generated and transmitted, it is discarded and a new random linear combination is generated for the next transmission. A receiver will collect these random linear combinations until it has received enough to decode; equivalently, the transmission of random linear combinations will continue until all M destinations have received K linearly-independent random linear combinations. Decoding will be performed by Gaussian elimination to recover the original K data packets.

Let $\widetilde{T}_K^{(m)}$ denote the number of slots needed for destination m to receive K linearly-independent random linear combinations, i.e., for destination m to be able to decode. The distribution of $\widetilde{T}_K^{(m)}$ will depend on the channel model, which is clearly defined in two separate cases below. As we assume that the channels are identically distributed with respect to destinations, $\widetilde{T}_K^{(m)}$ will be identically distributed with respect to m. The total time needed to complete transmission to all M destinations is given by $\widetilde{T}_K = \max_m \widetilde{T}_K^{(m)}$. The average number of data packets received per transmission is given by the ratio $K/E[\widetilde{T}_K]$. We account for the overhead by scaling the number of packets received per transmission by $n/(n+K)$, which is the ratio of number of information symbols to the total number of symbols (information plus overhead) sent with each transmission. We define throughput S as the effective portion of each transmission that contains message information; S is a measure of throughput in packets per transmission and takes values between 0 and 1.

$$S = \frac{K}{E[\widetilde{T}_K]} \frac{n}{n+K} = \frac{K}{E[\max_m \widetilde{T}_K^{(m)}]} \frac{n}{n+K} \text{ packets/transmission} \tag{5.1}$$

The difficulty in evaluating the throughput S is in solving for the expected maximum $E[\max_m \widetilde{T}_K^{(m)}]$, particularly because $\widetilde{T}_K^{(m)}$ are correlated random variables. To deal with this difficulty, we cite the following lemma from [45].

Lemma 12. *Suppose $X_1, \ldots, X_M$ are identically distributed, but not necessarily independent random variables. For any $t > 0$,*

$$E[\max(X_1, \ldots, X_M)] \leq \frac{1}{t} \left(\ln M + \ln E[e^{tX_1}] \right). \tag{5.2}$$

By applying Lemma 12 to our expression for the throughput in 5.1, we obtain a lower bound on the effective throughput as described in the following theorem.

Theorem 13. *The effective multicast throughput S in packets per transmission for random linear coding over groups of K packets is lower bounded, for any $t > 0$, as*

$$S \geq \frac{tK}{\ln M + \ln E[e^{t\widetilde{T}}]} \frac{n}{n+K} \tag{5.3}$$

where random variable $\widetilde{T}$ denotes the number of slots needed for successful recovery of the K packets at one of the M destinations and n denotes the number of information symbols per packet.

Next we consider two particular channel models and in each case, we describe $E[e^{t\widetilde{T}}]$, which allows us to characterize the throughput.

5.3 A time-invariant channel

In this section we assume that the behavior of the wireless channel to each of the M destination nodes is invariant with respect to time. In each time slot, we let $q(n,u)$ denote the probability that a transmitted packet (random linear combination) is correctly received. The channel is time-invariant in the sense that $q(n,u)$ takes the same value in every time slot. As discussed above, for a realistic wireless channel, the probability that a transmitted packet is dropped must increase with the length of the packet in bits. As such, $q(n,u)$ is decreasing in the symbols per packet n and the alphabet size u. For this time-invariant channel, the following result characterizes $E[e^{t\widetilde{T}}]$.

Theorem 14. *For a time-invariant channel with erasure probability $q(n,u)$ in each slot and random linear coding over K packets, the moment generating function of the number of slots needed for a destination node to recover the original K packets is given by*

$$E[e^{t\widetilde{T}}] = \prod_{j=0}^{K-1} \frac{q(n,u)\left(1-u^{-K}(u^j-1)\right)}{e^{-t}-1+\left(1-u^{-K}(u^j-1)\right)q(n,u)} \tag{5.4}$$

Proof. Let Y_j, $j = 0, 1, \ldots, K-1$ denote the number of slots needed for the destination to receive a linearly-independent random linear combination given that j linearly-independent random linear combinations have already been received at the destination. Then Y_j is geometrically distributed with parameter $(1 - u^{-K}(u^j - 1))q(n,u)$ since there are $u^j - 1$ possible ways to generate a new random linear combination that is linearly dependent on the j random linear combinations that have already been received. Furthermore, Y_j, $j = 0, 1, \ldots, K-1$ are independent. The total time needed for a destination to receive K linearly independent random linear combinations is given by $\widetilde{T} = Y_0 + Y_1 + \ldots + Y_{K-1}$. Then we have

$$
\begin{aligned}
E[e^{t\widetilde{T}}] &= E[e^{t(Y_0+Y_1+\ldots+Y_{K-1})}] && (5.5)\\
&= E[e^{tY_0}e^{tY_1} \times \ldots \times e^{tY_{K-1}}] && (5.6)\\
&\overset{(a)}{=} \prod_{j=0}^{K-1} E[e^{tY_j}] && (5.7)\\
&\overset{(b)}{=} \prod_{j=0}^{K-1} \frac{q(n,u)\left(1 - u^{-K}(u^j - 1)\right)}{e^{-t} - 1 + (1 - u^{-K}(u^j - 1))\, q(n,u)} && (5.8)
\end{aligned}
$$

where equality (a) is due to independence and (b) follows from the moment generating function of a geometric random variable. □

We obtain a lower bound on the effective throughput for the time invariant channel by applying the expression for $E[e^{t\widetilde{T}}]$ given in Theorem 14 to the bound provided in Theorem 13. Note that in order for $E[e^{t\widetilde{T}}]$ to be a positive quantity, we require that $t < -\ln(1 - q(n,u)(1 - u^{-1} + u^{-K}))$ in addition to $t > 0$ as stated in Theorem 13.

We now describe the means by which $q(n,u)$ is characterized. Let us assume that there is no channel coding or redundancy among the n symbols in a packet. In other words, $\{s_1, s_2, \ldots, s_K\}$ are uncoded information symbols. Then a random linear combination is received without error only if all n symbols are received without

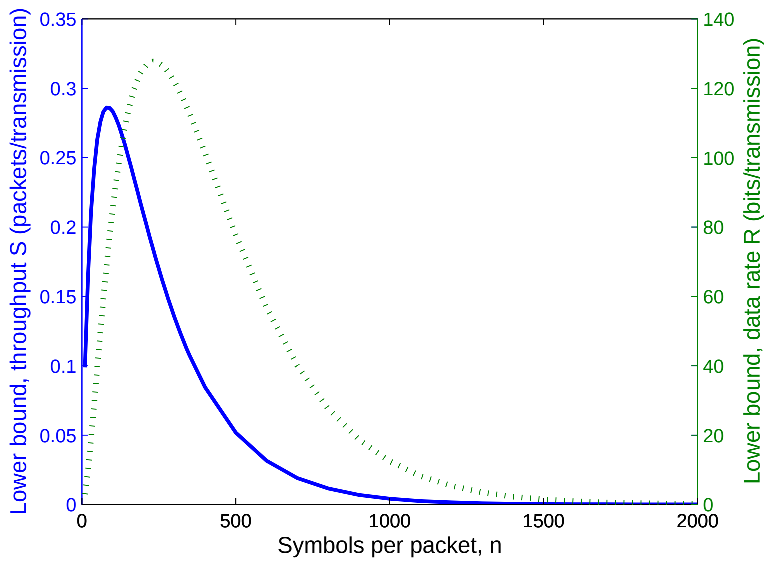

Figure 5.1: Lower bounds on the throughput S (solid line) and data rate R (dashed line) versus symbols per packet n for a time-invariant channel. Multicast transmission to $M = 10$ destination nodes with alphabet $u = 8$, QAM modulation, and SNR/bit 3.5 dB. Random linear coding is performed over $K = 80$ packets.

error. Then

$$q(n, u) = (1 - P_u)^n. \tag{5.9}$$

where P_u denotes the error probability for a u-ary symbol transmitted over the channel. We note that P_u is independent of n but depends on u as well as features of the channel such as pathloss and resulting signal-to-noise ratio (SNR). We will consider a wireless channel of limited bandwidth, for which modulation techniques such as pulse amplitude modulation (PAM), phase shift keying (PSK), and quadrature amplitude modulation (QAM) are appropriate. For these modulation techniques, $P_u \to 1$ as $u \to \infty$ [46], which implies that for fixed n, $q(n, u) \to 0$ as $u \to \infty$. Furthermore, it is clear from 5.9 that $q(n, u)$ approaches zero exponentially fast as n increases.

As a numerical example, we have plotted the lower bounds on the through-

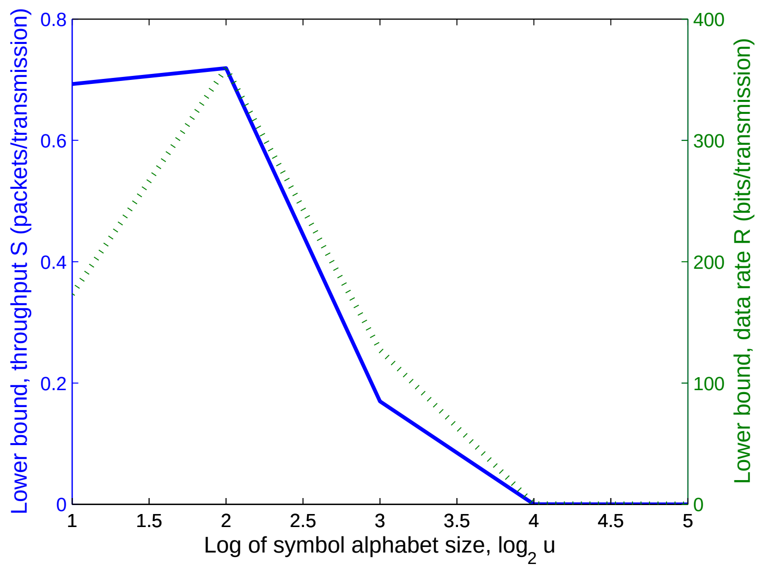

Figure 5.2: Lower bounds on the throughput S (solid line) and data rate R (dashed line) versus alphabet size u for a time-invariant channel. Multicast transmission to $M = 10$ destination nodes with $n = 250$ symbols per packet, QAM modulation, and SNR/bit 3.5 dB. Random linear coding is performed over $K = 80$ packets.

put in packets per transmission S and on the data rate $R = Sn \log_2 u$ in bits per transmission for QAM modulation over an additive white Gaussian noise (AWGN) channel. In this case the symbol error probability for the optimum detector is approximated by [46]

$$P_u \approx 1 - \left(1 - 2\left(1 - \frac{1}{\sqrt{u}}\right) Q\left(\frac{3\gamma_b \log_2 u}{u-1}\right)\right)^2 \tag{5.10}$$

where γ_b is the SNR per bit and Q is the complementary cumulative distribution function for the Gaussian distribution. The above expression for P_u holds with equality for $\log_2 u$ even. For these results the value of t in Theorems 13 and 14 has been numerically optimized subject to the criteria $0 < t < -\ln(1 - q(n,u)(1 - u^{-1} + u^{-K}))$. Example results are shown in three different forms. In Figure 5.1, we have displayed the throughput as a function of n for fixed values of u and K. Figure

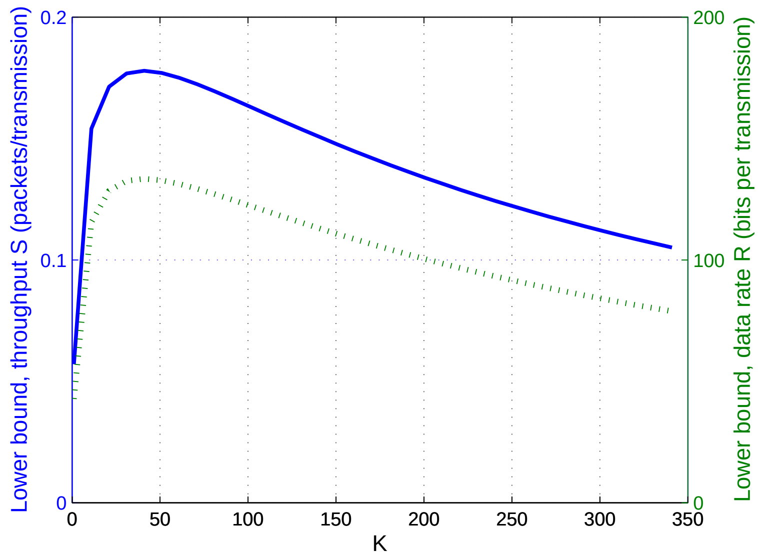

Figure 5.3: Lower bounds on the throughput S (solid line) and data rate R (dashed line) versus K for a time-invariant channel. Multicast transmission to $M = 10$ destination nodes with $n = 250$ symbols per packet, QAM modulation, alphabet size $u = 8$, and SNR/bit 3.5 dB.

5.2 shows throughput as a function of the alphabet size u when n and K are fixed. Finally, Figure 5.3 displays the throughput as a function of K for fixed n and u. In all three cases, we see that the throughput and data rate are concave functions, admitting optimum values for n, u, and K. We note that if any one of n, u, or K grows large as the other two values are fixed, the throughput approaches zero.

5.4 A time-varying channel

In this section we consider the multicast throughput performance of random coding over a channel for which the probability of reception $q(n, u)$ varies in time. We assume that the length or information content of a packet, given by the symbols per packet n and the alphabet size u, is fixed. Thus for a channel of fixed bandwidth,

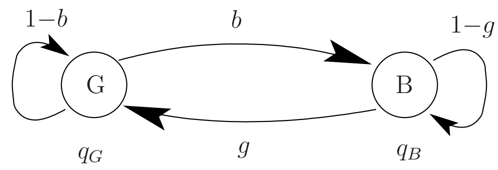

Figure 5.4: Packet erasure version of the Gilbert-Elliot channel model. The probability of packet reception is denoted q_G in state G and q_B in state B.

the duration of the time slot needed to transmit a packet is fixed. We further assume that the channel varies slowly with respect to the packet duration; equivalently, the coherence time of the channel is larger than the packet duration. This assumption is manifested by assuming that the reception probability $q(n, u)$ is fixed over a time slot. However the probability $q(n, u)$ will vary from slot to slot.

We will represent a time-varying channel by a packet-erasure version of the Gilbert-Elliot model [47]. We assume that in any time slot, the channel is in one of two possible states S: it is either in a "good" state or in a "bad" state. Thus the state variable S takes one of two possible values, $S \in \{G, B\}$. When in state G, a transmitted packet is correctly received with probability $q_G(n, u)$; when in state B, a transmitted packet is correctly received with probability $q_B(n, u)$. As stated above, we assume that n and u are fixed and simplify notation by denoting $q_G(n, u)$ as simply q_G and $q_B(n, u)$ as simply q_B. The state S of the channel evolves according to a Markov chain with transition probabilities g and b as shown in Fig. 5.4. We assume that the channels to each of the M destinations evolve according to the Gilbert-Elliot model with identical parameters q_G, q_B, g, and b.

To describe the multicast throughput of the random coding scheme over this time-varying channel, we develop techniques to compute $E[e^{t\widetilde{T}}]$ for each of the M

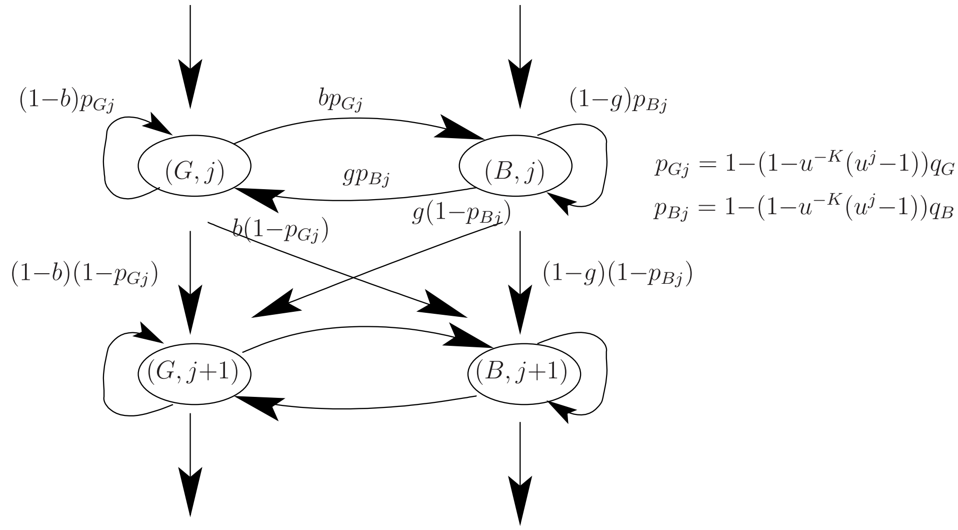

Figure 5.5: Markov chain model representing the process of transmission to one destination over the time-varying channel.

destination nodes and then apply Theorem 13. If we focus on a single destination node, the reception process can be represented by the augmented Markov chain shown in Fig. 5.5. The state of the chain is given by the pair (S, j), where $S \in \{G, B\}$ denotes the channel state, and $j \in \{0, 1, \ldots, K\}$ denotes the number of linearly independent packets the destination has received. The parameter K denotes the number of packets involved in forming random linear combinations. The Markov chain consists of $2(K + 1)$ states. The states (G, K) and (B, K) are absorbing states since the original K packets can be recovered from either of those states. The transition probabilities of the Markov chain are as shown in Fig. 5.5. The total transmission time $\widetilde{T}$ corresponds to the number of slots needed to reach state (S, K) from state $(S, 0)$. We make use of the following general result on the moment generating function of the hitting time for a discrete-time Markov chain in order to determine $E[e^{t\widetilde{T}}]$.

Theorem 15. *For a Markov chain with state space $\{1, 2, \ldots, n\}$ and transition probability matrix $\mathbf{P}$, let λ_i denote the eigenvalues of $\mathbf{P}$ and T_i denote the time needed to reach state n from state i, $i \in \{1, \ldots, n-1\}$. For all $t \notin \{-\log(\lambda_i) | \lambda_i$ is real and positive$\}$*

$$E[e^{tT_i}] = 1 + (e^t - 1)[(\mathbf{I} - e^t\mathbf{P})^{-1}\mathbf{g}]_i \tag{5.11}$$

where $[\mathbf{v}]_i$ denotes the i^{th} element of $\mathbf{v}$ and $\mathbf{g}$ is a column vector with $g(n) = 0$ and $g(i) = 1$ for $i \in \{1, \ldots, n-1\}$.

Proof. In terms of the transition probability matrix $\mathbf{P}$, we can express the distribution of T_i as follows.

$$\begin{aligned}
\Pr(T_i{=}j) &= \Pr(\text{transition } i \text{ to } n \text{ in } j \text{ steps}) - \Pr(\text{transition } i \text{ to } n \text{ in } j{-}1 \text{ steps}) \\
&= 1 - [\mathbf{P}^j\mathbf{g}]_i - (1 - [\mathbf{P}^{j-1}\mathbf{g}]_i) \\
&= [\mathbf{P}^{j-1}\mathbf{g}]_i - [\mathbf{P}^j\mathbf{g}]_i \qquad (5.12)
\end{aligned}$$

Then the moment generating function is given by

$$\begin{aligned}
E[e^{tT_i}] &= \sum_{j=1}^{\infty} e^{tj}\Pr(T_i = j) \\
&= \sum_{j=1}^{\infty} e^{tj}[\mathbf{P}^{j-1}\mathbf{g}]_i - \sum_{j=1}^{\infty} e^{tj}[\mathbf{P}^j\mathbf{g}]_i \\
&= \sum_{j=0}^{\infty} e^{t(j+1)}[\mathbf{P}^j\mathbf{g}]_i + 1 - \sum_{j=0}^{\infty} e^{tj}[\mathbf{P}^j\mathbf{g}]_i \\
&= 1 + \sum_{j=0}^{\infty} \left(e^{t(j+1)} - e^{tj}\right)[\mathbf{P}^j\mathbf{g}]_i \\
&= 1 + \left(e^t - 1\right)\sum_{j=0}^{\infty} [e^{tj}\mathbf{P}^j\mathbf{g}]_i. \qquad (5.13)
\end{aligned}$$

Next, let $\mathbf{z}(t) = \sum_{j=0}^{\infty}(e^t\mathbf{P})^j\mathbf{g}$. For $\mathbf{z}(t)$ the following holds.

$$\begin{aligned}
\mathbf{z}(t) &= \sum_{j=0}^{\infty}(e^t\mathbf{P})^j\mathbf{g} \\
&= \mathbf{g} + e^t\mathbf{P}\sum_{j=0}^{\infty}(e^t\mathbf{P})^j\mathbf{g} \\
&= \mathbf{g} + e^t\mathbf{P}\mathbf{z}(t) \qquad (5.14)
\end{aligned}$$

Then $(\mathbf{I} - e^t\mathbf{P})\mathbf{z}(t) = \mathbf{g}$. For any t such that $t \notin \{-\log(\lambda_i)|\lambda_i$ is real and positive$\}$, the matrix inverse $(\mathbf{I} - e^t\mathbf{P})^{-1}$ exists. In this case $\mathbf{z}(t) = (\mathbf{I} - e^t\mathbf{P})^{-1}\mathbf{g}$ and the moment generating function is given by

$$\begin{aligned}
E[e^{tT_i}] &= 1 + \left(e^t - 1\right)\sum_{j=0}^{\infty}[e^{tj}\mathbf{P}^j\mathbf{g}]_i \\
&= 1 + \left(e^t - 1\right)\left[\sum_{j=0}^{\infty}e^{tj}\mathbf{P}^j\mathbf{g}\right]_i \\
&= 1 + \left(e^t - 1\right)\left[(\mathbf{I} - e^t\mathbf{P})^{-1}\mathbf{g}\right]_i . \qquad (5.15)
\end{aligned}$$

□

We apply Theorem 15 to determine $E[e^{t\widetilde{T}}]$ for random coding over the time-varying channel in the following way. We assume that $q_B = 0$, so that a packet can only be received when the channel is in state G. This assumption ensures that the reception process always initiates in state $(G, 0)$. The generating function $E[e^{t\widetilde{T}}]$ is given by the time to reach state (G, K) from state $(G, 0)$. We let $\mathbf{P}$ denote the $2(K + 1) \times 2(K + 1)$ transition matrix of the Markov chain shown in Fig. 5.5. We make use of the column vector $\mathbf{g}$ whose elements are all 1 except for the elements corresponding to states (G, K) and (B, K), where $\mathbf{g}$ takes value 0. By making use of Theorem 15, we find $E[e^{t\widetilde{T}}]$ of the time needed for a destination node to receive K linearly independent coded packets. We then apply Theorem 13 to determine the multicast throughput. This computation can be done numerically and the value of t in Theorem 13 is optimized numerically subject to $t > 0$ and $t \notin \{-\log(\lambda_i) | \lambda_i \text{ is real and positive}\}$.

We compare the resulting bound on the effective multicast throughput for our time-varying channel with the corresponding bound for an equivalent time-invariant channel with reception probability

$$q = \frac{g}{g+b} q_G + \frac{b}{b+g} q_B. \tag{5.16}$$

An example of the results is shown in Fig. 5.6, which plots the effective multicast throughput versus K for three different time-varying channels (given by three different pairs (g, b)) and their equivalent time-invariant channel. We have assumed a fixed value of $q_G(n, u)$, given by QAM modulation over an AWGN noise channel as shown in Equations 5.9 and 5.10, and $q_B(n, u) = 0$. In these results, the effective throughput for the time-varying channel is smaller than the throughput for the equivalent time-invariant channel, indicating the extent to which the variation in the channel causes a degradation in performance.

5.5 Discussion

In this chapter, we explored the multicast throughput performance of random coding while taking account of the packet length, overhead, and physical effects of the wireless medium. We modeled the erasure probability as an increasing function of the packet length, given by the symbols per packet n and the symbol alphabet u.

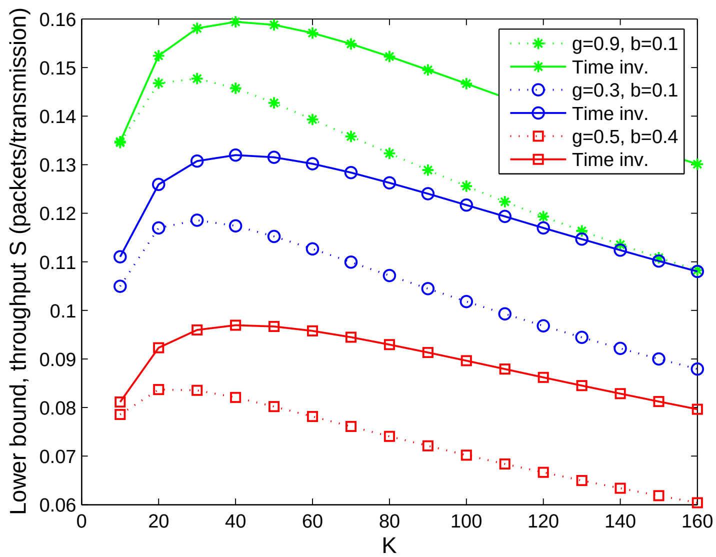

Figure 5.6: Lower bounds on the throughput S versus K for a time-varying channel. Multicast transmission to $M = 10$ destination nodes with $n = 250$ symbols per packet, QAM modulation, alphabet size $u = 8$, and SNR/bit 3.5 dB in the good state and $-\infty$ dB in the bad state. The dotted line shows throughput over a time-varying channel with given g and b values and the solid line shows throughput over the equivalent time-invariant channel.

Our results clearly indicate a tradeoff between the packet length and the throughput. As the number of symbols per packet n grows large, the effect of overhead becomes negligible, but the erasure probability grows and the transmissions are more likely to fail. Alternatively, as n approaches zero, transmissions are more likely to succeed, but the number of information symbols becomes negligible compared to the overhead symbols. In a similar manner, as the alphabet size u grows, the random linear coding becomes more efficient in the sense that the random coefficients a_i approach real numbers, for which we need only K random linear combinations in order to decode. However, a large alphabet size causes the erasure probability to increase and transmissions are likely to be unsuccessful.

We have also explored the effective throughput as a function of K, the number of packets involved in random linear coding. In contrast to our observations in Chapters 2 and 3, where we saw that the throughput increases with K, in this chapter we showed that when accounting for the overhead required for random coding, the throughput becomes a concave function of K.

Our results characterize the effective multicast throughput for both a time-invariant and a time-varying channel. For our packet erasure Gilbert-Elliot channel, we have quantified the degradation in throughput due to the time-varying nature of the channel. In all cases, we observed that the throughput is a concave function of n, u, and K. As such, system design for wireless multicast with random coding should involve the choice of the throughput-optimal n, u, and K.

We have argued in this chapter that for a practical channel, the erasure probability is an increasing function of the symbols per packet n and the symbol alphabet u. We considered as an example a simple case in which there is no error-control coding performed among the symbols in the packet prior to the random linear coding performed over multiple packets. In this example, all n symbols in the packet must be received without error and it's clear that the erasure probability approaches one exponentially fast as n increases. Alternatively, we could consider a concatenated coding approach, such as the one in [48], where error-correcting codes are applied to the symbols within a data packet before random linear coding is applied over multiple packets. In this case, the packet erasure probability will still approach one as n increases, albeit at a slower rate due to the error-correcting code. In this case, the behavior of the channel could be used to determine which of the two levels of coding should introduce more redundancy in order to maximize the overall throughput of the system.

Finally, our emphasis in this chapter on the multicast throughput performance as a function of the packet length has implications for network coding in multihop networks. We have shown that when accounting for overhead and the physical effects of the channel, over a single hop, the throughput approaches zero as either the symbols per packet n or the alphabet size u increases. This behavior has implications for a number of previous results, including [37, 49, 50], which consider network coding over a network of erasure links and analyze performance in the limit as the packet length grows without bound. Our results suggest a need to further examine the performance of network coding for finite-length packets.

Chapter 6

Conclusion

6.1 Summary of contributions

We first analyzed the multicast throughput in a random access network of finitely many nodes, each of which serves as either a source node or a destination node. We quantified the Shannon capacity region as well as the stable throughput and the saturated throughput for retransmissions and random linear coding. Our results indicate the extent to which the random linear coding scheme can outperform the retransmission scheme. We showed that for multicast transmission, the stable throughput region *does not* coincide with the Shannon capacity region; this result stands in stark contrast to a longstanding conjecture that the stable throughput and capacity regions coincide for unicast transmission. Finally, we provided inner and outer bounds on the stable multicast throughput for arbitrarily (but finitely) many source nodes; the saturated throughput was shown to fall between the inner and outer bounds. These contributions are described in Chapter 2 and appear in our published works [51, 52, 53, 54].

Next, we considered a random access network of general topology in which each node can act as a receiver or a sender for multiple multicast flows. We examined the multicast throughput both from the perspective of the receiver (non-guaranteed multicast) and from the perspective of the transmitter (guaranteed multicast). In both cases, we quantified the throughput and presented schemes for nodes in the network to compute their random access transmission probabilities in such a way as to maximize a weighted proportional fairness objective function of the throughput. Our approach to the guaranteed multicast problem involved the use of random coding. Chapter 3 describes our contribution to this problem, which has been published in [55].

We then turned our focus to the queueing delay performance of random linear coding. We proposed that random coding of packets be modeled as a bulk-service queueing system, where packets are served and depart the queue in groups. In the bulk-queueing framework, we analyzed two different random linear coding schemes. The first scheme involved a fixed expected coding rate (or fixed number of packets used in coding), which led to a delay penalty for lightly-loaded systems. The second scheme adapted to the traffic load by allowing for a variable number of packets used in coding, thereby removing the delay penalty at low loads. These contributions are presented in Chapter 4 and in [56, 57].

Finally, we returned to the question of multicast throughput and addressed the effects of packet length, overhead, and the time-varying nature of the wireless channel. We presented a packet erasure model for which the erasure probability increases as a function of the packet length, given by the number of symbols per packet and the symbol alphabet. We showed how the modulation scheme and noise level of the channel could be accounted for in the erasure probability. We also quantified the performance of random linear coding over a time-varying channel, which was modeled by a packet-erasure channel for which the erasure probability transitions between 'good' and 'bad' states. These results appear in Chapter 5 and portions of this work have been published in [58, 59].

6.2 Additional contributions and collaborations

In addition to the work presented in this dissertation, we have made contributions to other problems involving multicast transmission. In [60, 61], we presented results on the information-theoretic feedback capacity of the compound channel. The compound channel can represent either multicast transmission or channel uncertainty. Our contributions to this problem include a derivation of the feedback capacity for a compound channel with memory. Furthermore, in [62], we contributed to a study of the stable throughput for random linear coding over packets from disparate multicast sessions, or inter-session coding. We showed that there are instances in which inter-session coding, which involves destination nodes recovering data that is not intended for them, outperforms a retransmission scheme. This work provides a simple example demonstrating that the back-pressure algorithm [63], which is optimal for store-and-forward networks, can be inferior to a coding approach for multicast transmission.

6.3 Future work

The work outlined in this dissertation has provided valuable theoretical analysis and useful design techniques for multicasting in wireless networks. There remain a number of open questions in this arena of research, and we conclude by describing two of them.

The use of feedback in wireless multicast transmission requires further research both from a theoretical and a practical perspective. In this dissertation, we have assumed that instantaneous, error-free feedback can be sent from the destination nodes to the source node, and this assumption is key to implementing the retransmission and random coding schemes for reliable multicast. In reality, a feedback link may be difficult to establish (e.g., in satellite broadcast) and may be prone to errors and delay. The use of feedback in multicast communication presents a particularly challenging problem: requiring feedback from all destinations would result in a "feedback flood." Important avenues of future research include determining the loss in multicast data rate due to the failure of a particular destination node to send feedback as well as proposing scalable feedback schemes that allow a destination node to determine in a distributed way whether or not to send feedback.

The benefit of network coding in wireless multihop networks is another open area of research. It has been shown that the optimal (maximum flow) multicast throughput in a wired network can be obtained by use of network coding in the limit as the packet length grows to infinity. As we have demonstrated in Chapter 5, for a practical channel, particularly a wireless channel, the transmission of infinitely long packets is infeasible and would lead to a throughput of zero. Open questions remain as to the throughput benefit of network coding for finite-length packets and channels with error. The overhead required for random network coding is an important feature; we have taken the first steps in accounting for it, and important open questions on this issue remain.

Appendix A

Markov chain model for random linear coding

In the Markov chain analysis of random linear coding presented in Chapter 2, the state (i, j, k) represents i linearly independent coded packets received at destination 1, j linearly independent coded packets received at destination 2, and k linearly independent coded packets that have been received at both destinations, $k \leq \min(i, j)$. When source 2 is backlogged, the non-zero transition probabilities for source 1 are given as follows for $i, j = 0, 1, \ldots, K$.

- $(i, j, k) \rightarrow (i, j, k)$:

$$\overline{p}_1 + p_1 \Big[\overline{p}_2 \Big\{(1-q_{1|1}^{(1)})(1-q_{1|1}^{(2)}) + (1-q_{1|1}^{(1)})q_{1|1}^{(2)}u^{j-K} + q_{1|1}^{(1)}(1-q_{1|1}^{(2)})u^{i-K} + q_{1|1}^{(1)}q_{1|1}^{(2)}u^{k-K}\Big\}$$
$$+p_2 \Big\{(1-q_{1|1,2}^{(1)})(1-q_{1|1,2}^{(2)}) + (1-q_{1|1,2}^{(1)})q_{1|1,2}^{(2)}u^{j-K} + q_{1|1,2}^{(1)}(1-q_{1|1,2}^{(2)})u^{i-K} + q_{1|1,2}^{(1)}q_{1|1,2}^{(2)}u^{k-K}\Big\}\Big]$$

- $(i, j, k) \rightarrow (i+1, j, k)$:

$$p_1 \Big[\overline{p}_2 \Big\{q_{1|1}^{(1)}(1-q_{1|1}^{(2)})(1-u^{i-K}) + q_{1|1}^{(1)}q_{1|1}^{(2)}(u^j - u^k)u^{-K}\Big\}$$
$$+p_2 \Big\{q_{1|1,2}^{(1)}(1-q_{1|1,2}^{(2)})(1-u^{i-K}) + q_{1|1,2}^{(1)}q_{1|1,2}^{(2)}(u^j - u^k)u^{-K}\Big\}\Big]$$

- $(i,j,k) \to (i,j+1,k):$

$$p_1\left[\overline{p}_2\left\{(1-q^{(1)}_{1|1})q^{(2)}_{1|1}(1-u^{j-K})+q^{(1)}_{1|1}q^{(2)}_{1|1}(u^i-u^k)u^{-K}\right\}\right.$$
$$\left.+p_2\left\{(1-q^{(1)}_{1|1,2})q^{(2)}_{1|1,2}(1-u^{j-K})+q^{(1)}_{1|1,2}q^{(2)}_{1|1,2}(u^i-u^k)u^{-K}\right\}\right]$$

- $(i,j,k) \to (i+1,j+1,k+1):$

$$p_1\left[\overline{p}_2\left\{q^{(1)}_{1|1}q^{(2)}_{1|1}(1-(u^i+u^j-u^k)u^{-K})\right\}+p_2\left\{q^{(1)}_{1|1,2}q^{(2)}_{1|1,2}(1-(u^i+u^j-u^k)u^{-K})\right\}\right]$$

- $(i,K,k) \to (i,K,k) : \overline{p}_1+p_1\left[\overline{p}_2\left\{(1-q^{(1)}_{1|1})+q^{(1)}_{1|1}u^{i-K}\right\}+p_2\left\{(1-q^{(1)}_{1|1,2})+q^{(1)}_{1|1,2}u^{i-K}\right\}\right]$

- $(i,K,k) \to (i+1,K,k):$

$$p_1\left[\overline{p}_2\left\{q^{(1)}_{1|1}(1-(u^i+K-k)u^{-K})\right\}+p_2\left\{q^{(1)}_{1|1,2}(1-(u^i+K-k)u^{-K})\right\}\right]$$

- $(i,K,k) \to (i+1,K,k+1) : p_1\left[\overline{p}_2\left\{q^{(1)}_{1|1}(K-k)u^{-K}\right\}+p_2\left\{q^{(1)}_{1|1,2}(K-k)u^{-K}\right\}\right]$

- $(K,j,k) \to (K,j,k) : \overline{p}_1+p_1\left[\overline{p}_2\left\{(1-q^{(2)}_{1|1})+q^{(2)}_{1|1}u^{j-K}\right\}+p_2\left\{(1-q^{(2)}_{1|1,2})+q^{(2)}_{1|1,2}u^{j-K}\right\}\right]$

- $(K,j,k) \to (K,j+1,k):$

$$p_1\left[\overline{p}_2\left\{q^{(2)}_{1|1}(1-(u^j+K-k)u^{-K})\right\}+p_2\left\{q^{(2)}_{1|1,2}(1-(u^j+K-k)u^{-K})\right\}\right]$$

- $(K, j, k) \to (K, j{+}1, k{+}1) : p_1 \left[\overline{p}_2 \left\{q^{(2)}_{1|1}(K-k)u^{-K}\right\} + p_2 \left\{q^{(2)}_{1|1,2}(K-k)u^{-K}\right\}\right]$

Bibliography

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4), July 2000.

[2] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49:371–381, February 2003.

[3] A. Ephremides. Energy concerns in wireless networks. *IEEE Wireless Communications*, 9:48–59, August 2002.

[4] N. Abramson. The aloha system - another alternative for computer communications. In *Fall Joint Computer Conference (AFIPS)*, volume 37, pages 281–285, 1970.

[5] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *IEEE INFOCOM*, 2000.

[6] M. Luby. Lt codes. In *IEEE Symposium on the Foundations of Computer Science*, pages 271–280, 2002.

[7] A. Shokrollahi. Raptor codes. *IEEE Transactions on Information Theory*, 52(6):2551–2567, June 2006.

[8] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *IEEE International Symposium on Information Theory*, 2003.

[9] D. J. C. MacKay. Fountain codes. In *Fourth IEE Workshop on Discrete Event Systems (WODES98)*, 1998.

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

[11] L. Kleinrock. *Queueing Systems, Volume I: Theory*. Wiley, New York, 1975.

[12] B.S. Tsybakov and V.A. Mikhailov. Ergodicity of a slotted aloha system. *Problems in Information Transmission*, 15:301–312, 1979.

[13] R.R. Rao and A. Ephremides. On the stability of interacting queues in a multiple-access system. *IEEE Transactions on Information Theory*, 34:918–930, September 1988.

[14] W. Szpankowski. Stability conditions for some distributed systems: Buffered random access systems. *Advances in Applied Probability*, 26:498–515, June 1994.

[15] W. Luo and A. Ephremides. Stability of n interacting queues in random-access systems. *IEEE Transactions on Information Theory*, 45:1579–1587, July 1999.

[16] V. Naware, G. Mergen, and L. Tong. Stability and delay of finite user slotted aloha with multipacket reception. *IEEE Transactions on Information Theory*, 51(7):2636–2656, July 2005.

[17] J. L. Massey and P. Mathys. The collision channel without feedback. *IEEE Transactions on Information Theory*, 31:192–204, March 1985.

[18] J. Y. N. Hui. Multiple accessing for the collision channel without feedback. *IEEE Journal on Selected Areas in Communications*, 2:575–582j, July 1984.

[19] G. Sh. Poltyrev. Coding in an asynchronous multiple access channel. *Problems on Information Transmission*, 19(3):184–191 (12–21 in Russian version), 1983.

[20] J. Y. N. Hui and P. A. Humblet. The capacity of the totally asynchronous multiple-access channel. *IEEE Transactions on Information Theory*, 31(2):207–216, March 1985.

[21] S. Tinguely, M. Rezaeian, and A. Grant. The collision channel with recovery. *IEEE Transactions on Information Theory*, 51:3631–3638, October 2005.

[22] V. Anantharam. Stability region of the finite-user slotted aloha protocol. *IEEE Transactions on Information Theory*, 37:535–540, May 1991.

[23] J. Luo and A. Ephremides. On the throughput, capacity and stability regions of random multiple access. *IEEE Transactions on Information Theory*, 52(6):2593–2607, June 2006.

[24] R. Ahlswede. The capacity region of a channel with two senders and two receivers. *Annals of Probability*, 2(5):805–814, 1974.

[25] R. G. Gallager. Basic limits on protocol information in data communication networks. *IEEE Transactions on Information Theory*, 22(4):385–398, July 1976.

[26] R.M. Loynes. The stability of a queue with non-independent interarrival and service times. *Proc. Cambridge Phil. Soc.*, 58, 1962.

[27] Y. E. Sagduyu and A. Ephremides. On broadcast stability region in random access through network coding. In *Allerton Conference on Communication, Control, and Computing*, September 2006.

[28] P. J. Grabner and H. Prodinger. Maximum statistics of n random variables distributed by the negative binomial distribution. *Combinatorics, Probability and Computing*, 6:179–183, 1997.

[29] K. Kar, S. Sarkar, and L. Tassiulas. Achieving proportionally fair rates using local information in aloha networks. *IEEE Transactions on Automatic Control*, 49(10):1858–1862, 2004.

[30] P. Gupta and A.L. Stolyar. Optimal throughput allocation in general random-access networks. In *Conference on Information Sciences and Systems (CISS)*, Princeton, March 2006.

[31] X. Wang and K. Kar. Cross-layer rate control for end-to-end proportional fairness in wireless networks with random access. In *Proc. MobiHoc*, Urbana-Champaign, May 2005.

[32] J. Mo and J. Walrand. Fair end-to-end window-based contention control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, 2000.

[33] J.-W. Lee, M. Chiang, and A. R. Calderbank. Jointly optimal congestion and contention control based on network utility maximization. *IEEE Communications Letters*, 10(3):216–218, 2006.

[34] P. Chaporkar and S. Sarkar. Wireless multicast: theory and approaches. *IEEE Transactions on Information Theory*, 51(6):1954–1972, 2005.

[35] P. Chaporkar and S. Sarkar. Minimizing delay in loss-tolerant mac layer multicast. *IEEE Transactions on Information Theory*, 52(10):4701–4713, 2006.

[36] K. Kar, S. Sarkar, and L. Tassiulas. Optimization based rate control for multirate multicast sessions. In *Proc. INFOCOM*, Alaska, 2001.

[37] A. Eryilmaz, A. Ozdaglar, and M. Medard. On delay performance gains from network coding. In *Conference on Information Sciences and Systems*, March 2006.

[38] D. S. Lun, P. Pakzad, C. Fragouli, M. Medard, and R. Koetter. An analysis of finite-memory random linear coding on packet streams. In *Fourth International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt06)*, April 2006.

[39] KCV K. S. Sayee and U. Mukherji. Stability of scheduled message communication over degraded broadcast channels. In *IEEE International Symposium on Information Theory*, June 2006.

[40] D. R. Jeske and T. Blessinger. Tunable approximations for the mean and variance of heterogeneous geometrically distributed random variables. *The American Statistician*, 58(4):322–327, 2004.

[41] H. Takagi. *Queueing Analysis, Volume 3: Discrete-Time Systems.* North-Holland, Amsterdam, 1993.

[42] L. Kleinrock. *Queueing systems, Volume II: Computer applications.* Wiley, New York, 1976.

[43] M.L. Chaudhry and J.G.C. Templeton. *A first course in bulk queues.* Wiley, New York, 1983.

[44] M. L. Chaudhry. private communication, 2007.

[45] R. Cogill. Randomized load balancing with non-uniform task lengths. In *Allerton Conference on Communication, Control, and Computing*, September 2007.

[46] J. G. Proakis. *Digital Communications, Third Edition.* McGraw-Hill, Boston, 1995.

[47] E. O. Elliot. Estimates of error rates for codes on burst noise channels. *Bell Systems Technical Journal*, 42:1977–1997, September 1963.

[48] M. Vehkapera and M. Medard. A throughput-delay trade-off in packetized systems with erasures. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, Adelaide, Australia, September 2005.

[49] D. S. Lun, M. Medard, R. Koetter, and M. Effros. Further results on coding for reliable communication over packet networks. In *IEEE Information Theory Symposium (ISIT)*, September 2005.

[50] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros. Capacity of wireless erasure networks. *IEEE Transactions on Information Theory*, 52(3), 2006.

[51] B. Shrader and A. Ephremides. Broadcast stability in random access. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, Adelaide, September 2005.

[52] B. Shrader and A. Ephremides. The capacity of the asynchronous compound multiple access channel and results for random access systems. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, Seattle, July 2006.

[53] B. Shrader and A. Ephremides. Random access broadcast: stability and throughput analysis. *IEEE Transactions on Information Theory*, 53:2915–2921, August 2007.

[54] B. Shrader and A. Ephremides. On the shannon capacity and queueing stability of random access multicast. *IEEE Transactions on Information Theory*, May 2007. submitted.

[55] B. Shrader and M. Andrews. Optimal multicast throughput in random access networks of general topology. In *Proc. Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Berlin, April 2008.

[56] B. Shrader and A. Ephremides. On the queueing delay of a multicast erasure channel. In *Proc. IEEE Information Theory Workshop (ITW)*, Chengdu, October 2006.

[57] B. Shrader and A. Ephremides. A queueing model for random linear coding. In *Proc. Military Communications Conference (MILCOM)*, Orlando, October 2007.

[58] B. Shrader and A. Ephremides. On packet lengths and overhead for random linear coding over the erasure channel. In *Proc. International Wireless Communications and Mobile Computing Conference (IWCMC)*, Honolulu, August 2007.

[59] R. Cogill, B. Shrader, and A. Ephremides. Stability analysis of random linear coding across multicast sessions. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, Toronto, July 2008.

[60] B. Shrader and H. H. Permuter. On the compound finite state channel with feedback. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, Nice, June 2007.

[61] B. Shrader and H. H. Permuter. Feedback capacity of the compound channel. *IEEE Transactions on Information Theory*, November 2007. submitted.

[62] R. Cogill, B. Shrader, and A. Ephremides. Stable throughput for multicast with inter-session network coding. In *Proc. IEEE Military Communications Conference (MILCOM)*, San Diego, November 2008. submitted.

[63] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12), 1992.