Clemson University
TigerPrints

All Dissertations Dissertations

12-2013

Adaptive Transmission Protocols for Wireless Communication Systems with Fountain Coding

Jason Ellis
Clemson University, jellisy15@yahoo.com

Follow this and additional works at: https://tigerprints.clemson.edu/all_dissertations

Part of the Electrical and Computer Engineering Commons

Recommended Citation

Ellis, Jason, "Adaptive Transmission Protocols for Wireless Communication Systems with Fountain Coding" (2013). *All Dissertations*. 1245.
https://tigerprints.clemson.edu/all_dissertations/1245

This Dissertation is brought to you for free and open access by the Dissertations at TigerPrints. It has been accepted for inclusion in All Dissertations by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

ADAPTIVE TRANSMISSION PROTOCOLS FOR WIRELESS COMMUNICATION SYSTEMS WITH FOUNTAIN CODING

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Electrical Engineering

by
Jason D. Ellis
December 2013

Accepted by:
Dr. Michael B. Pursley, Committee Chair
Dr. Daniel L. Noneaker
Dr. Harlan B. Russell
Dr. Taufiquar Khan

ABSTRACT

We present low-complexity adaptive protocols for both unicast and multicast transmission in wireless communication systems that employ higher layer fountain codes. Our adaptive protocols respond to variations in channel conditions by adapting the modulation and channel coding of transmitted packets, and they provide efficient communication over wireless channels that experience fading, shadowing, and other time-varying propagation losses. The operation of our protocols is governed by simple receiver statistics that can be obtained during the demodulation of received packets. We present three adaptive protocols for fountain-coded unicast transmission, and compare the throughput performance of our protocols with that of fixed-rate systems, as well as hypothetical ideal protocols that are given perfect channel state information and use ideal fountain codes. We also present two adaptive protocols for fountain-coded multicast transmission. Our adaptive multicast transmission protocols operate with limited feedback from the destinations and provide scheduling to avoid collisions among the feedback messages. We compare the performance of our multicast protocols to systems with fixed modulation and coding, as well as hypothetical protocols that are given perfect channel state information. We demonstrate that our practical adaptive protocols for fountain-coded unicast and multicast transmission outperform fixed-rate coding schemes and provide throughput that is nearly as high as that achieved by hypothetical protocols that are given perfect channel state information.

DEDICATION

This dissertation is dedicated to my late father, Larry D. Ellis, for letting me be "Captain" even before I learned how to lead.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Michael B. Pursley, for all of his assistance in the preparation of this dissertation. I am very grateful for all of the guidance and support that he has given me during my graduate education. I would like to thank Dr. Daniel L. Noneaker, Dr. Harlan B. Russell, and Dr. Taufiquar Khan for their generous efforts in serving on my committee. I am very grateful for the support that they have shown me and the roles that they have played in my success here at Clemson.

I would also like to thank family and friends for their love, prayers, and support. I am truly blessed to have such wonderful people in my life.

TABLE OF CONTENTS

Page

APPENDICES

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

List of Figures (Continued)

Figure Page

CHAPTER 1
INTRODUCTION

Wireless communication systems require error-control coding at the physical layer in order to communicate over noisy channels, but they can also benefit from erasure correction coding at a higher layer. The combination of a good channel code with a CRC code can convert a noisy channel into an erasure channel by discarding packets that fail to decode. Recent advances have produced powerful fountain codes, such as Luby Transform (LT) codes and raptor codes [1]–[4], that may be used for erasure correction in communication systems. A fountain code is a forward-error-control code that can produce as many redundant packets as needed for packet erasure correction. Unlike automatic-repeat-request (ARQ) transmission, fountain coding does not require the destination to inform the source of the identities of the packets that are erased or even keep track of which packets are erased. We examine the use of fountain coding for both unicast and multicast transmission in packet radio systems, where communication takes place over time-varying channels with fading, shadowing, and other types of propagation losses.

In [4], Shokrollahi states, "A decoding algorithm for a Fountain code is an algorithm which can recover the original k input symbols from any set of n output symbols with high probability. For good Fountain codes the value of n is very close to k, ... Note that the number n is the same regardless of the channel characteristics between the sender and the receiver. More loss of symbols just translates to a longer waiting time to receive the n packets." Thus, for noisy wireless channels, the *waiting time*, and consequently the overall throughput performance of the fountain coding system, depends heavily on the selection of the channel code and the modulation format used to transmit the wireless signals. For time-varying channel conditions, good performance is not guaranteed by the use of a fixed modulation and coding scheme. Even if a powerful fountain code is employed for erasure correction, the number of packet erasures may be much larger than the number of excess packets required by the fountain code, and the system will still have poor performance. For either unicast or multicast transmission, if the information rate of the modulation and channel coding is too high for the link(s) to support, then excessive packet erasures will cause the performance of the system to suffer in poor channel conditions. If the conditions of one or more links are poor for a very long time (e.g., a link may have a continuing deep fade), then the number of packet erasures, and consequently the waiting time, may be indefinite. Alternatively, if the information rate of the modulation and channel coding is too low, then an excess amount of redundant information will be transmitted when channel conditions are good, and the system will not take advantage of the erasure correction provided by the fountain code. Adaptive transmission at the physical layer allows the system to take advantage of good channel conditions by transmitting at a high rate, yet maintain efficient communication by reducing packet losses in poor channel conditions.

In [5] and [6], the authors discuss the importance of optimizing the rate of the channel code in fountain coding systems; however, they do not consider any adaptation, and they assume that the average signal-to-noise ratio (SNR) of the channel is known prior to transmission. The goal of their approach is to choose the channel code of optimum rate based on knowledge of the average SNR and the parameters of the fading channel. Similarly, the results in [7] show the advantages of selecting the proper rate for the channel code in fountain coding systems, but adaptation is not considered. In [8], the authors present methods for adaptive-rate channel coding for fountain-coded transmissions, but the fade level is required to be known to the transmitter, and it must be constant for several packet transmissions. The "adaptation" does not occur from packet to packet in [8]. In practice, the fade level is typically unknown to the transmitter, and for most fading channels, the fade level varies over time and is not constant for several packet transmissions. Because the fade level varies over time, no fixed modulation or channel code is optimum for the entire session; however, unlike our protocols, the approaches of [5]–[8] do not attempt to adapt the modulation format or the channel code from packet to packet.

We present three low-complexity adaptive protocols for unicast transmission in packet radio systems with higher layer fountain coding. These protocols respond to time-varying channel conditions by adapting the modulation and channel coding of the transmitted signals. The parameters of the channel are unknown to our protocols, yet they require no pilot symbols, parameter estimates, or channel measurements. Instead, the protocols use only very simple receiver statistics for their control information to adapt the modulation and channel code. These receiver statistics can be easily obtained during the demodulation and decoding of received packets. We evaluate the throughput performance of our unicast protocols for time-varying channels with fading and shadowing, and we show that our adaptive protocols achieve better throughput performance than fountain-coded systems with fixed modulation and channel coding. We compare the throughput of our protocols with hypothetical ideal protocols that are given perfect channel state information and use ideal fountain codes that do not require any excess packet transmissions. We demonstrate that our protocols perform nearly as well as these ideal protocols. We present a modification of our adaptive transmission protocols that improves their performance at low SNR by permitting the protocols to transmit a short, known binary sequence in very poor channel conditions, when packet erasures are very likely. We also demonstrate that our unicast protocols can still provide good performance with delayed or intermittent feedback.

Adaptive multicast transmission presents challenges that are not encountered with unicast transmission. In order to effectively adapt the modulation and channel code of transmitted signals from packet to packet, the source must obtain feedback from multiple destinations. Also, different destinations may be experiencing different channel conditions that can support different rates of transmission. An adaptive multicast transmission protocol must be able to accommodate the needs of multiple destinations when selecting the modulation and channel code. In [9], we introduce an adaptive multicast transmission protocol that was designed with these challenges in mind. Our protocol provides scheduling for feedback messages from the destinations and a selection criterion for choosing a modulation and channel code to accommodate multiple destinations; however, we do not consider multicast systems that employ fountain coding. In [10], strategies for adaptive modulation and channel coding are presented for multicast systems with fountain coding, but the transmitter is assumed to have perfect knowledge of the channel state for all links of the multicast channel. We present two practical adaptive protocols for fountain-coded multicast transmissions that implement different selection criteria for adapting the modulation and channel code used to transmit the fountain-coded packets. Our protocols operate using the same simple receiver statistics that we employ for the control information of our unicast protocols. We compare the performance of our protocols to hypothetical protocols that are given perfect channel state information for all links in the multicast channel. We show that our adaptive protocols outperform fixed modulation and channel coding in multicast systems that employ fountain coding and perform nearly as well as the ideal protocols given perfect channel state information.

CHAPTER 2
UNICAST SYSTEM MODEL

2.1 Fountain Coding System

A *source* radio has a large number of information bits to be delivered to a single *destination* radio. We define an *information packet* to be a collection of information bits that make up one input symbol to the fountain code. The fountain code is applied across a collection of information packets that make up a *frame*, and it produces a potentially infinite number of fountain-coded packets, or *data packets*. Each data packet is further encoded by a channel code to form a *channel packet*, and channel packets are transmitted over the wireless channel to the destination. Once a sufficiently large set of data packets have been received, the destination is able to apply the fountain decoder to recover the frame.

The fountain coding system is illustrated in Fig. 2.1. The information bits are divided into frames with K information packets per frame. The information packets in a frame serve as input symbols to a fountain encoder to produce as many data packets as needed to decode the frame. The fountain encoder first chooses an integer d from the set $\{1, \ldots, K\}$ according to its degree distribution, then selects d of the K information packets at random in order to form a data packet. The data packet is the modulo-2 sum of the d information packets. Thus, if $\mathbf{s}_j$ denotes the jth information packet, the data packet is

$$\mathbf{b} = \sum_{j=1}^{K} a_j \mathbf{s}_j, \tag{2.1}$$

where exactly d of the coefficients a_j are equal to one and all others are equal to zero.

The data packets are further encoded by a channel encoder in order to mitigate the effects of noise on the wireless channel. A set of L binary block codes, $\{C_i : 1 \leq$

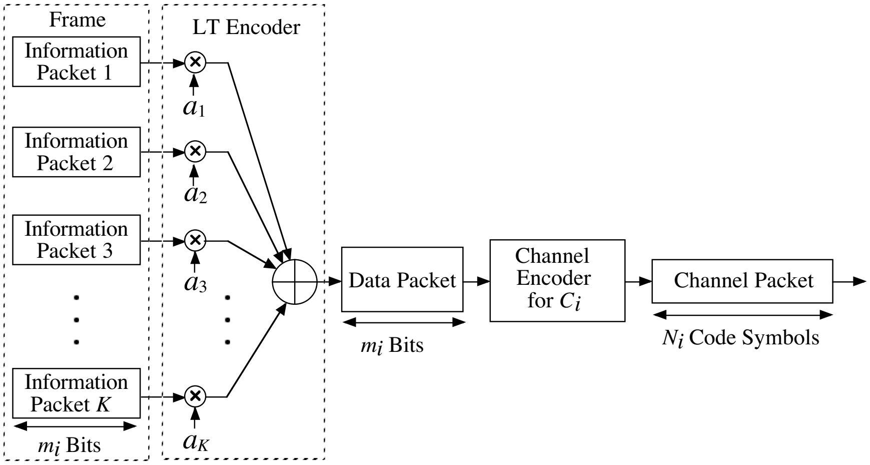


Figure 2.1: Fountain encoder and packet formation for channel code C_i.

$i \leq L\}$, indexed in order of increasing rates, is available to our adaptive transmission protocols for adaptive-rate channel coding. Code C_i has block length n_i with k_i information bits and rate $r_i = k_i/n_i$. The formation of a channel packet when channel code C_i is used is shown in Fig. 2.1. Each channel packet is a channel-encoded version of exactly one data packet. Thus, the length of each data packet and each channel packet depends on the channel code being used to transmit the packet. When code C_i is employed, there are Q_i codewords per channel packet. The length of each channel packet is $N_i = Q_i n_i$ binary code symbols, and the length of each data packet is $m_i = Q_i k_i$. Note that the length of each information packet is the same as the data packets, and there are m_i information bits per information packet. A high-rate CRC code (not shown) is used to verify the correct decoding of each channel packet; however, we ignore the small loss in rate due to the CRC code for our numerical results.

The fountain coding system generates as many data packets as needed for the destination to receive enough unerased packets to decode the frame. If the destination receives $K + \varepsilon$ unerased packets before the fountain decoder is able to recover the frame, then ε is the number of *excess packets* required to decode the frame. The fountain code illustrated in Fig. 2.1 is an LT code; however, any type of fountain code is suitable for the system. For all of our numerical results, the systematic 3GPP raptor code [11] is used for fountain coding. An LT code lies at the core of the raptor code and is easier to illustrate because the raptor code requires a pre-coding step between the formation of the frame of information packets and the application of the LT code. The 3GPP raptor code employed by our system is a systematic fountain code. For a systematic fountain code, the first K transmitted packets are the input symbols to the fountain code. Thus, if all of the initial K packets are received successfully by the destination, then there is no need for the fountain decoder, and there is also no overhead due to excess packet transmissions. In some implementations of systematic fountain codes, the initial K transmitted packets are fountain-coded packets that make up a decodable set. If all K packets are unerased, then the fountain decoder is able to recover the information packets without the need for excess packet transmissions. A systematic fountain code provides better efficiency in good channel conditions when packet erasures are less likely.

2.2 Modulation Formats and Channel Codes

Our adaptive protocols employ bit-interleaved coded modulation [12]. In addition to the set $\{C_i : 1 \leq i \leq L\}$ of channel codes, our protocols also have a set $\{\mathcal{M}_k : 1 \leq k \leq W\}$ of modulation formats available for the transmission of channel packets. We refer to a pair consisting of a modulation format and a channel code as a *code-modulation combination*. The sets of available channel codes and modulation formats allow for LW possible code-modulation combinations. A set of H_n code-modulation combinations are employed by the nth adaptive unicast transmission protocol ($n = 1, 2, 3$), and the set of combinations for the nth protocol is denoted by $\mathcal{B}_n = \{B_n(h) :$

$1 \leq h \leq H_n\}$.

The set of channel codes for $\mathcal{B}_1$ and $\mathcal{B}_2$ are designed to provide a fixed number of binary code symbols per channel packet and a variable number of information bits per packet. The set of channel codes for $\mathcal{B}_3$ are designed to give a fixed number of information bits per packet and a variable number of binary code symbols per channel packet. The set $\mathcal{B}_1$ includes all LW code-modulation combinations; however, some of these combinations are of little or no value to the other two protocols. These combinations may not benefit the throughput that can be achieved by the protocol, so they are removed from the sets $\mathcal{B}_2$ and $\mathcal{B}_3$ to decrease the complexity of the protocols. The elimination of the these combinations is based on their performance over a static additive white Gaussian noise (AWGN) channel and do not require simulation of dynamic channels. For example, if the throughput of one combination is below that of another combination for all values of SNR, then it is not included in the set $\mathcal{B}_2$ or $\mathcal{B}_3$. The parameters for the sets $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ used to generate our numerical results are provided in Appendix A.

The modulation formats available to our protocols include binary phase-shift keying (BPSK), quadriphase-shift keying (QPSK), M-ary quadrature amplitude modulation (M-QAM), and M-ary biorthogonal modulation, also known as M-ary biorthgonal keying (M-BOK). These modulation formats are described in [13]. Each modulation format employed by our protocol uses a pulse waveform referred to as the *modulation chip* to transmit modulation symbols. For BPSK, QPSK, and M-QAM, there is one modulation chip per modulation symbol, and for M-BOK there are $M/2$ modulation chips per modulation symbol. The specific modulation formats employed by our protocols are BPSK, QPSK, 16-QAM, 16-BOK, and 64-BOK.

The performance measure used to evaluate our adaptive protocols is the *session throughput*. The average session throughput for a reliable unicast system with fountain coding is defined to be the average number of information bits in a frame divided by the average number of time units that the source is required to transmit channel packets until the destination's fountain decoder is able to decode the entire frame. For our numerical results, one time unit is a single modulation chip. The total transmission time includes all channel packets that are successfully decoded by the destination's channel decoder, as well as the packets that fail to decode. For our practical protocols, the transmission time also includes the time spent transmitting any excess packets that are required by the fountain decoder. The hypothetical protocols employ ideal fountain codes and do not require any excess packets. Though the session throughput is the performance measure for the adaptive protocol, the selection of the code-modulation combination is based on the expected throughput for a single packet transmission. A data bit in a channel packet is said to be *delivered* if it is contained in a channel packet that is decoded correctly by the destination's channel decoder. The average throughput for the transmission of a single channel packet is the average number of information bits delivered to the destination per unit time that the source spends transmitting the packet.

2.3 Channel Models

We evaluate the performance of our adaptive protocols in dynamic channel environments with fading and other time-varying propagation losses. We use finite-state Markov chains to model Nakagami-m fading channels [14]. A procedure for determining the steady-state probabilities and transition probabilities for a finite-state Markov chain model of a Nakagami fading process is demonstrated in [15]. We make the common assumption for slow fading that the fade level is constant for the duration of a packet transmission. Thus, the state of the Markov chain is fixed for the duration of a packet but may change from one packet to the next.

The bandwidth is unchanged when our protocols adapt the modulation format; therefore, the modulation chip rate is unchanged. The transmitter power is also held constant during adaptation by our protocols, so the chip-energy-to-noise-density ratio is the appropriate measure of SNR because it is unchanged when the modulation and channel coding are adapted. If $\mathcal{E}_c$ is the average energy per chip and N_0 is the one-sided power spectral density, then the chip-energy-to-noise-density ratio is denoted by $\mathcal{E}_c/N_0$. We let $\text{CENR} = 10\log_{10}(\mathcal{E}_c/N_0)$ denote the chip-energy-to-noise-density ratio in decibels (dB). Each state of the Markov chain corresponds to a value of CENR. We let CENR* be the chip-energy-to-noise-density ratio in the absence of fading. The values of CENR for adjacent states of the Markov chain differ by a fixed amount in dB, but transitions are not restricted to adjacent states only. We let G_j denote the channel gain for state j, and the value of CENR for state j of the Markov chain is given by

$$\text{CENR}_j = \text{CENR}^* + G_j. \tag{2.2}$$

The fading Markov chain is assumed to be in steady state, and we denote the probability of being in state j by π_j. For a J_f-state Markov chain, the probabilities π_j, $0 \le j \le J_f - 1$, are the *state probabilities.* The probability of transitioning from state j to state k in one step is $q(k|j)$. The values for the transition probabilities depend on the normalized Doppler frequency, $f_d T_s$, where f_d is the Doppler frequency and T_s is the average duration between the starting times for consecutive packets transmitted by the source. The correlation coefficient for samples of the Nakagami-m process that differ in time by T_s is given by $\rho = J_0^2(2\pi f_d T_s)$, where J_0 is the Bessel function of the first kind of order zero. As $f_d T_s$ increases, the fading becomes faster, and the probability of staying in the same state for consecutive packets decreases. Also, the probability of transitioning to a nonadjacent state in one step increases.

We also use finite-state Markov chains to model shadowing in addition to fading. The Markov chain used to model shadowing is independent of the fading Markov chain. We consider two shadowing scenarios. The first shadowing scenario is modeled by a two-state Markov chain where the difference in SNR between states is Δ dB. In other words, state 0 represents no shadowing and state 1 represents a Δ dB shadow loss. The transition probabilities are $q_1(1|0) = 0.002$ and $q_1(0|1) = 0.05$, so the system experiences a Δ dB shadow loss approximately 4% of the time. The second shadowing scenario is modeled by a three-state Markov chain with all three states being equally likely. The difference in SNR between states is Δ dB, so the system experiences no shadowing, a Δ dB shadow loss, or a 2Δ dB shadow loss. The transition probability matrix for this shadowing Markov chain is given by

$$q_2 = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.2 & 0.6 & 0.2 \\ 0 & 0.2 & 0.8 \end{bmatrix} . \qquad (2.3)$$

We let J_f be the number of states in the fading chain and J_s be the number of states in the shadowing Markov chain. To simplify the notation, we consider a single J-state Markov chain, where $J = J_f J_s$, that represents both independent Markov chains for fading and shadowing. Thus, if j_1 is the state of the fading Markov chain and j_2 is the state of the shadowing Markov chain, then $\mathbf{j} = (j_1, j_2)$ is the state vector for the composite Markov chain.

CHAPTER 3
INTER-FRAME ADAPTIVE CHANNEL CODING WITH INTRA-FRAME ADAPTIVE MODULATION

3.1 Protocol Description

Our first adaptive protocol for fountain-coded unicast transmission adapts the modulation and channel code in two separate stages. The first stage of the protocol consists of the selection of the channel code at the start of each frame. The channel code is only adapted at the start of a frame, which we refer to as *inter-frame adaptive channel coding*. In the second stage of the adaptive protocol, *intra-frame adaptive modulation*, the protocol utilizes a fixed channel code for the duration of a frame and adapts the modulation format from packet to packet throughout the frame. Thus, the protocol combines *adaptive modulation from packet to packet* (MP) with *adaptive channel coding from frame to frame* (CF), which we refer to as the AMPCF protocol.

At the start of a new frame, the adaptive protocol selects the channel code to be used for the duration of the frame. The AMPCF protocol is designed for systems that use fixed-length channel packets, so the length of the information packets in the frame depends on the channel code chosen for the frame. If code C_i is selected at the start of the frame from the set of available channel codes, then the frame is divided into K information packets containing m_i information bits per information packet, as illustrate in Fig. 2.1. The length of each data packet is also m_i data bits. The set of modulation formats available to the protocol is $\{\mathcal{M}_k : 1 \le k \le W\}$. The *initial modulation* format $\mathcal{M}_0 \in \{\mathcal{M}_k : 1 \le k \le W\}$ is used to transmit the first channel packet in the frame. This modulation is arbitrary and can be chosen by the protocol designer. For all of our numerical results, the initial modulation $\mathcal{M}_0$ is QPSK. For the remaining data packets in the frame, the AMPCF protocol responds to changes in the channel conditions by adapting the modulation format. When the destination has collected enough data packets to permit the fountain decoder to recover the frame, then the protocol selects the channel code for the next frame and adapts the modulation during transmission of the frame.

For the AMPCF protocol, demodulator statistics obtained during the reception of each channel packet provide the control information for the selection of the channel code for a frame and the adaptation of the modulation format during transmission of a frame. The demodulator statistic for a packet depends on the modulation format used to transmit the packet. For BPSK, QPSK, and M-QAM, the operation of the protocol is governed by the *distance statistic*. The distance statistic is the average Euclidean distance between the received point at the output of the demodulator and the closest point in the signal constellation, averaged over all modulation symbols in the channel packet. For M-BOK, the protocol relies on the *ratio statistic* for its operation. The ratio statistic is the magnitude of the second largest correlator output divided by the magnitude of the largest correlator output, averaged over all modulation symbols in the channel packet [16].

Prior to the formation of a frame, the protocol applies an interval test to the most recent demodulator statistic from the previous frame, and the result of this interval test dictates the selection of the channel code to be used for the impending frame. The decision intervals, $\{\mathcal{I}_f(\ell) : 1 \leq \ell \leq L\}$, depend on the modulation format, $\mathcal{M}_f$, used to transmit the final channel packet of the previous frame. If the modulation format for the final packet of the preceding frame is $\mathcal{M}_f$ and the demodulator statistic falls in interval $\mathcal{I}_f(\ell)$, then channel code C_ℓ is chosen as the channel code for the next frame, and the new frame is generated with m_ℓ information bits per packet. Once the channel code has been selected, the protocol transmits the first channel packet with the initial modulation format $\mathcal{M}_0$ and adapts the modulation format for the remaining data packets within the frame by applying an interval test to the demodulator statistic for the most recent channel packet. The result of the interval test governs the selection of the modulation format used to transmit the next channel packet. If code C_ℓ is the channel code used for the frame and $\mathcal{M}_p$ is the modulation for the previous channel packet, then the intervals $\{\mathcal{J}_{p,\ell}(k) : 1 \leq k \leq W\}$ serve as the decision intervals to adapt the modulation during transmission of the frame. If the demodulator statistic is in interval $\mathcal{J}_{p,\ell}(n)$, then modulation $\mathcal{M}_n$ is used to transmit the next channel packet. The details of the interval tests are shown in Appendix A.

The endpoints for the adaptation intervals are determined heuristically to provide the maximum throughput per packet transmission for a static AWGN channel with a known SNR. If accurate analytical performance results for the channel coding system do not exist, then simulations can be used to determine the packet error probability of the channel code as a function of the SNR. These probabilities serve two purposes, as they are also in the analytical expressions used to compute the throughput of our hypothetical adaptive protocols that serve as benchmarks for our practical protocols. Only simulations of static AWGN channels are required to obtain the packet error probabilities. Simulation of time-varying fading channels are not necessary, even though we evaluate the performance of our protocols for time-varying channels.

3.2 Performance Benchmarks

We compare the performance of our adaptive protocols to the performance of hypothetical ideal protocols that are given different levels of perfect channel state information. The first protocol is given *perfect state information for the next packet transmission*, which means it is told which state the Markov chain will be in when the next packet is to be transmitted. The hypothetical protocol uses its knowledge of the next channel state to perform adaptation of the modulation from packet to packet and the channel code from frame to frame, and it is denoted by AMPCF-N. This protocol employs an ideal fountain code that requires no excess packet transmissions, and it serves as the first benchmark for our practical AMPCF protocol.

The throughput performance for the AMPCF-N protocol can be determined analytically. We let $P_c[i,k|j]$ be the probability that a channel packet is decoded correctly by the destination if channel code C_i and modulation $\mathcal{M}_k$ are used to transmit the packet, and the channel is in state j. For a channel packet transmitted with C_i and $\mathcal{M}_k$, we let $N_I[i,k]$ denote the number of information bits in the packet and $\tau[i,k]$ denote the number of time units (modulation chips) required to transmit the packet. The expected throughput for a single channel packet transmitted with channel code C_i and modulation $\mathcal{M}_k$ when the channel is in state j is given by

$$s[i,k|j] = N_I[i,k]P_c[i,k|j]/\tau[i,k]. \tag{3.1}$$

For the AMPCF protocol, the transmission time $\tau[i,k]$ does not depend on the channel code because the protocol uses fixed-length channel packets, so without loss of generality, we can denote the time units by $\tau[k]$.

Now assume that the hypothetical protocol has chosen code C_ℓ to serve as the channel code for the next frame. For each data packet, the AMPCF-N protocol knows that the channel will be in state j when the packet is transmitted; consequently, it can determine the value of $P_c[\ell,k|j]$ for each k and compute $s[\ell,k|j]$. The protocol then chooses the modulation format with index $\kappa(\ell,j)$ to transmit the next channel packet in the frame if

$$s[\ell, \kappa(\ell, j)|j] = \max\{s[\ell, k|j] : 1 \le k \le W\}. \tag{3.2}$$

As a result, the average number of information bits delivered per packet by the AMPCF-N protocol for a frame that uses C_ℓ as the channel code is

$$\overline{I}_{N,1}(\ell) = \sum_{j=0}^{J-1} \pi_j s[\ell, \kappa(\ell, j)|j]\tau[\kappa(\ell, j)], \tag{3.3}$$

and the average number of time units that the source spends transmitting channel packets for the frame is

$$\overline{T}_{N,1}(\ell) = \sum_{j=0}^{J-1} \pi_j \tau[\kappa(\ell, j)]. \tag{3.4}$$

The resulting average throughput for the AMPCF-N protocol when it selects C_ℓ as the channel code for a frame is

$$\mathcal{S}_{N,1}(\ell) = \overline{I}_{N,1}(\ell)/\overline{T}_{N,1}(\ell). \tag{3.5}$$

The selection of the channel code for the AMPCF protocol is made only at the start of a frame and not packet to packet, so there are several packet transmissions that occur between each selection of the channel code. The AMPCF-N approximates the behavior of the adaptive channel coding stage of the AMPCF protocol by making a random selection of the channel code for each frame. The first channel packet of each frame is transmitted with the predetermined initial modulation $\mathcal{M}_0$. We let $s[i, 0|m]$ denote the throughput for the initial packet transmission if the the channel code is C_i and the channel is in state m when the packet is transmitted. The protocol draws at random from the state indexes $\{0, 1, \ldots, J-1\}$ according to the state probability distribution π_j, $0 \leq j \leq J-1$. If the result of the random drawing is m, then the protocol chooses the channel code whose index is $\ell(m)$ if $s[\ell(m), 0|m] = \max\{s[i, 0|m] : 1 \leq i \leq L\}$. The protocol employs channel code $C_{\ell(m)}$ for the entire frame, and performs adaptive modulation from packet to packet. The resulting average session throughput for the AMPCF-N protocol is given by

$$\overline{\mathcal{S}}_{N,1} = \frac{\sum_{m=0}^{J-1} \pi_m \overline{I}_{N,1}(\ell(m))}{\sum_{m=0}^{J-1} \pi_m \overline{T}_{N,1}(\ell(m))}. \tag{3.6}$$

In practice, no adaptive protocol is given information or able to make predictions about future channel conditions and must rely on information obtained during the reception of a packet for its operation. Thus, the second protocol that serves as a more realistic benchmark for the performance of our practical AMPCF protocol is given *perfect state information for the previous packet reception.* This protocol is told what state the Markov chain was in when the most recent channel packet was received by the destination. We refer to the hypothetical AMPCF protocol that is given knowledge of the previous channel state as the AMPCF-P protocol. This protocol does not have as much information about the state of the channel as the AMPCF-N protocol because its channel state information is slightly out of date.

For adaptation of the modulation from packet to packet, the AMPCF-P protocol still applies (3.2); however, the channel state j now represents the previous channel state rather than the next state. Because the Markov chain has a transition opportunity after each packet transmission, the modulation selected by the AMPCF-P protocol is chosen to maximize the expected throughput for state j, but the packet is transmitted when the channel is in state k. Thus, the average number of information bits delivered per packet by the AMPCF-P protocol for a frame transmitted with channel code C_ℓ is

$$\overline{I}_{P,1}(\ell) = \sum_{j=0}^{J-1} \pi_j \tau[\kappa(\ell,j)] \sum_{k=0}^{J-1} q(k|j) s[\ell, \kappa(\ell,j)|k], \tag{3.7}$$

the average number of time units that the source spends transmitting packets for the frame is

$$\overline{T}_{P,1}(\ell) = \sum_{j=0}^{J-1} \pi_j \tau[\kappa(\ell,j)], \tag{3.8}$$

and the average throughput for a frame that uses code C_ℓ is

$$\mathcal{S}_{P,1}(\ell) = \overline{I}_{P,1}(\ell)/\overline{T}_{P,1}(\ell). \tag{3.9}$$

Combining the throughput for adaptation of the modulation format with random selection of the channel code results in the average session throughput for the AMPCF-P protocol, which is given by

$$\overline{\mathcal{S}}_{P,1} = \frac{\sum_{m=0}^{J-1} \pi_m \overline{I}_{P,1}(\ell(m))}{\sum_{m=0}^{J-1} \pi_m \overline{T}_{P,1}(\ell(m))}. \tag{3.10}$$

When the frame size K is small, or the fading of the channel is very slow, the Markov chain may not reach steady-state over the course of a single frame. In this case, the choice of the channel code at the start of the frame becomes much more important, and the hypothetical protocols must make the choice in the same manner as the practical AMPCF protocol in order to be accurate benchmarks. We modify the AMPCF-N and AMPCF-P protocols to include a *deterministic* selection of the channel code at the start of a frame, and we refer to these modified protocols as the AMPCF-ND and AMPCF-PD protocols. The modified hypothetical protocols are given the same channel state information as our previous hypothetical protocols;

however, when selecting the channel code, the AMPCF-ND and AMPCF-PD protocols do not make a random selection, and instead emulate the practical AMPCF protocol. These protocols choose the channel code at the start of a frame that provides the maximum expected throughput for the QPSK-modulated packet that is the initial packet in the frame. The practical AMPCF protocol relies on a demodulator statistic to make its selection of the channel code; however, the AMPCF-ND protocol knows the initial channel state for the impending frame when choosing the channel code, and the AMPCF-PD protocol knows the state of the channel when the last packet of the previous frame was transmitted. Because the AMPCF-ND and the AMPCF-PD protocols operate in the same manner as the practical AMPCF protocol, their performance evaluations require simulation of the Markov chain models for the fading and shadowing on the link to the destination. The simulations of the Markov chains are combined with the values of $P_c[i,k|j]$ to compute the throughput for these modified hypothetical protocols. In contrast, all of the performance results for the AMPCF-N and AMPCF-P protocols can be obtained analytically, without any simulations of the fading channel or Markov chain models.

3.3 Performance Results

We evaluate the performance of our adaptive protocols in dynamic channel environments that experience fading and shadowing. Recall that the hypothetical AMPCF-N and AMPCF-P protocols make a random selection of the channel code according to the state distribution of the Markov chain model. When the fading is fast or the frame size employed by the fountain code is large, the analytical results for these benchmarks obtained from (3.6) and (3.10) provide good approximations for the performance of the practical AMPCF protocol. With fast fading or large frame

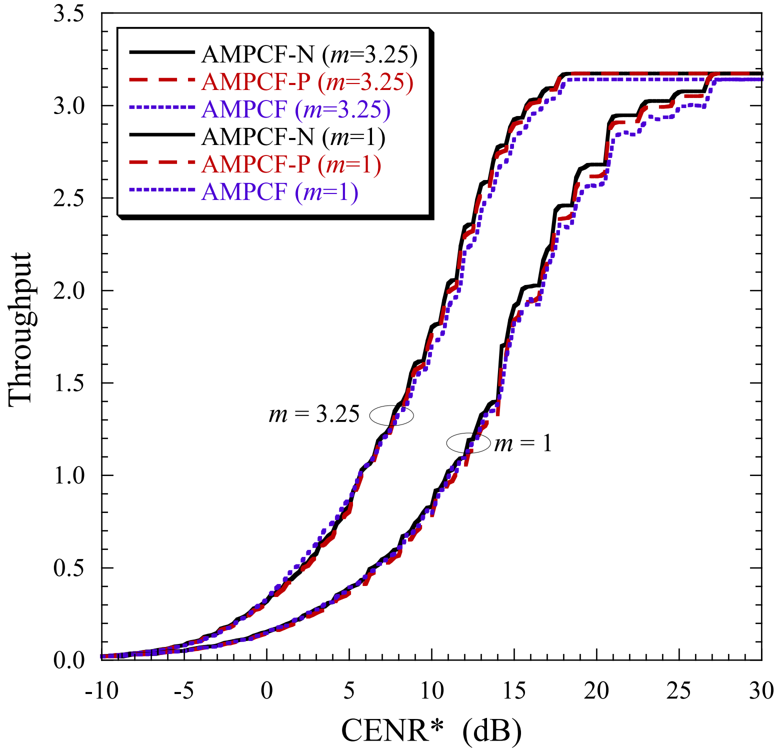


Figure 3.1: Throughput results for the analytical benchmarks and practical AMPCF protocols ($K = 100$, $f_dT_s = 0.02$).

sizes, we can expect the channel to reach steady state before the end of a frame; therefore, it is reasonable to select the channel code at random according to the steady state distribution. The throughput performance for the hypothetical AMPCF-N and AMPCF-P protocols and the simulated practical AMPCF protocol that relies on demodulator statistics are shown in Fig. 3.1 for Nakagami-m fading with $m = 1$, which is severe (Rayleigh) fading, and $m = 3.25$, which is less severe fading. The frame size is not large at $K = 100$ information packets per frame, but the normalized Doppler frequency of $f_dT_s = 0.02$ is relatively fast fading. Consequently, the analytical results for the hypothetical protocols are an accurate approximation of the simulated results for the practical protocol.

For slower fading, the selection of the channel code is much more important to the performance of the protocol because the channel does not reach steady state during the transmission of a single frame. As a result, the modified AMPCF-ND and AMPCF-PD protocols serve as better estimates of the performance of the practical AMPCF protocol. This is demonstrated in Fig. 3.2, where the performance of the hypothetical and practical AMPCF protocols is shown for Nakgami-m fading with $m = 1$ and a normalized Doppler frequency of $f_d T_s = 0.005$. When K is not large and the fading is slow, the analytical results for the AMPCF-N and AMPCF-P protocols do not agree with the simulated results for the practical protocol or the hypothetical protocols with deterministic selection of the channel code for CENR* from 2 dB to 16 dB. The analytical results for the AMPCF-N and AMPCF-P protocols are still of interest because they are easy to compute, and they provide a conservative estimate of the practical AMPCF protocol. However, if a more accurate prediction of the performance of the protocol is required, then the AMPCF-ND and AMPCF-PD protocols should be used.

The throughput for the AMPCF-N protocol computed with (3.6) does not depend on the transition probabilities for the fading Markov chain and thus, does not depend on $f_d T_s$. Therefore, the curves for the AMPCF-N protocol for $m = 1$ in Figs. 3.1 and 3.2 are identical. This means that the performance of the practical AMPCF protocol improved for the slower fading channel in Fig. 3.2. When selecting the channel code for the next frame, the practical protocol is able to take advantage of the higher correlation between consecutive states of the channel for slower fading because it obtains a demodulator statistic for the final packet of the previous frame. The AMPCF-N protocol is not able to take advantage of the statistical dependence of the prior states when selecting the channel code, so its performance does not improve in a slower fading channel. In contrast, the performance of the AMPCF-ND and AMPCF-PD protocols have very close agreement with the performance of the practical AMPCF protocol. Recall that these protocols operate in the same manner

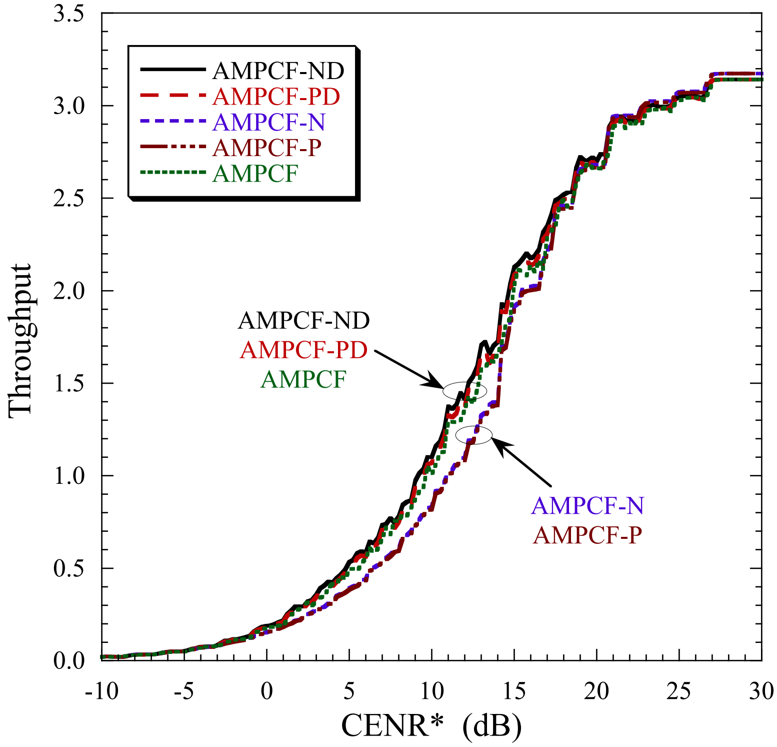


Figure 3.2: Throughput results for the hypothetical and practical AMPCF protocols ($K = 100$, $m = 1$, $f_d T_s = 0.005$).

as the practical protocol, but they are given perfect channel state information for the next or previous packet transmission, while our practical protocol is only given a demodulator statistic obtained during the reception of the previous packet.

In Fig. 3.3, we consider a system with a frame size that is four times larger than that of Fig. 3.2. The fading channel is unchanged from Fig. 3.2 to Fig. 3.3, but with a frame size that is much larger than the inverse of the normalized Doppler frequency, $(f_d T_s)^{-1}$, the channel can be expected to reach steady state during the transmission of a single frame. Thus, the random selection of the channel code based on the steady state distribution performed by the AMPCF-N and AMPCF-P protocols is a more reasonable estimate of the operation of the protocol. In Fig. 3.3 for a frame size of $K = 400$, we see that the analytical results for AMPCF-N and AMPCF-P protocols exhibit much better agreement with the performance of the practical protocol than

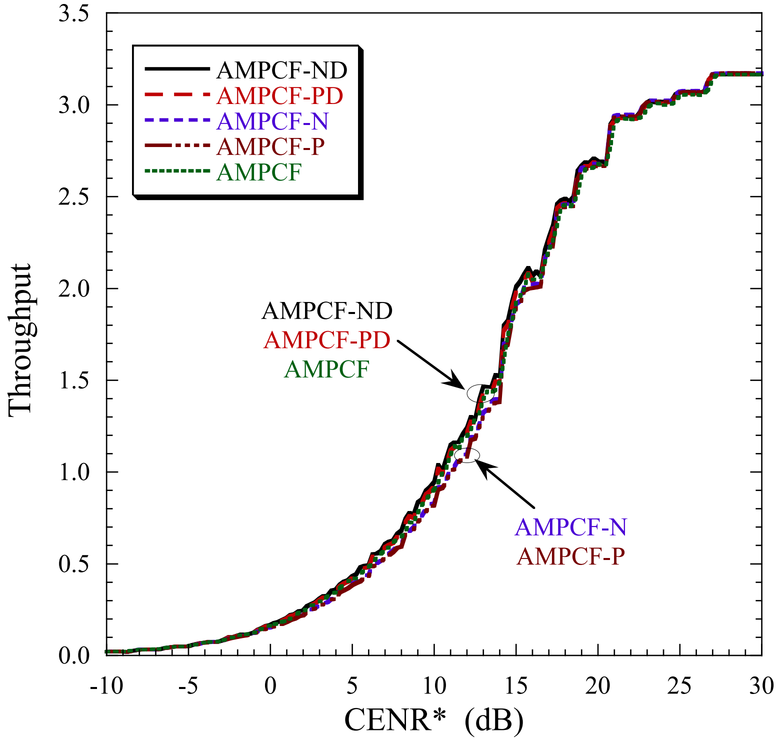


Figure 3.3: Throughput results for the hypothetical and practical AMPCF protocols ($K = 400$, $m = 1$, $f_dT_s = 0.005$).

with a frame size of $K = 100$ in the mid-range values of CENR*. They are still a conservative estimate of the practical protocol, but the maximum percentage increase in throughput for the AMPCF-ND, AMPCF-PD, and AMPCF protocols over the AMPCF-N and AMPCF-P protocols is reduced by approximately one-third when the frame size is increased from 100 to 400.

A comparison of the throughput for our AMPCF protocol with the throughput for three fixed code-modulation combinations is shown in Fig. 3.4 for a Nakagami-m fading channel with $m = 1$, $f_dT_s = 0.02$, and a frame size of $K = 100$. As expected, our protocol achieves a throughput nearly as high as that of the hypothetical protocols given perfect channel state information. Our protocol also outperforms each of the fixed combinations for a large range of CENR* in Fig. 3.4. For example, 16-QAM with the rate 0.793 turbo product code (TPC) provides higher throughput

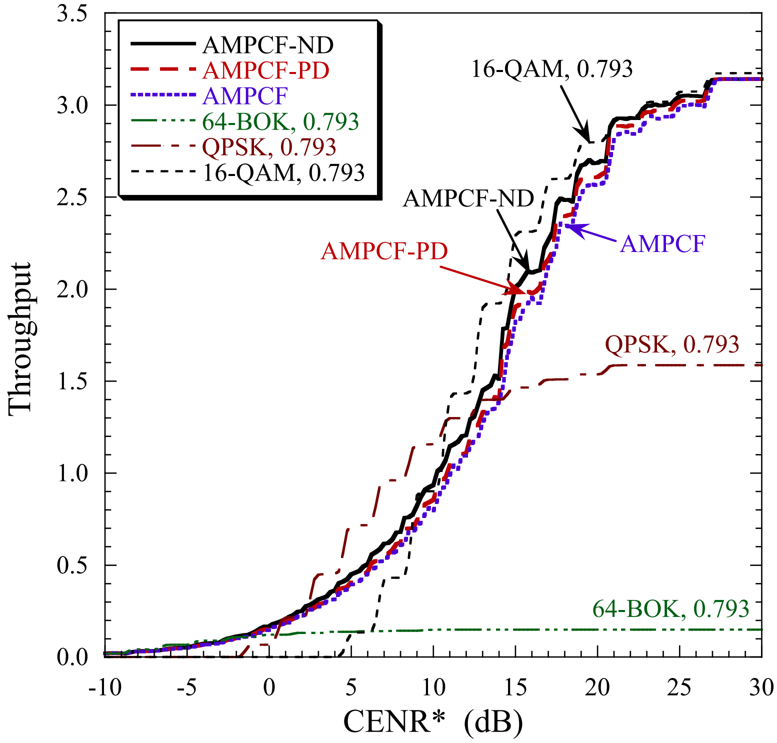
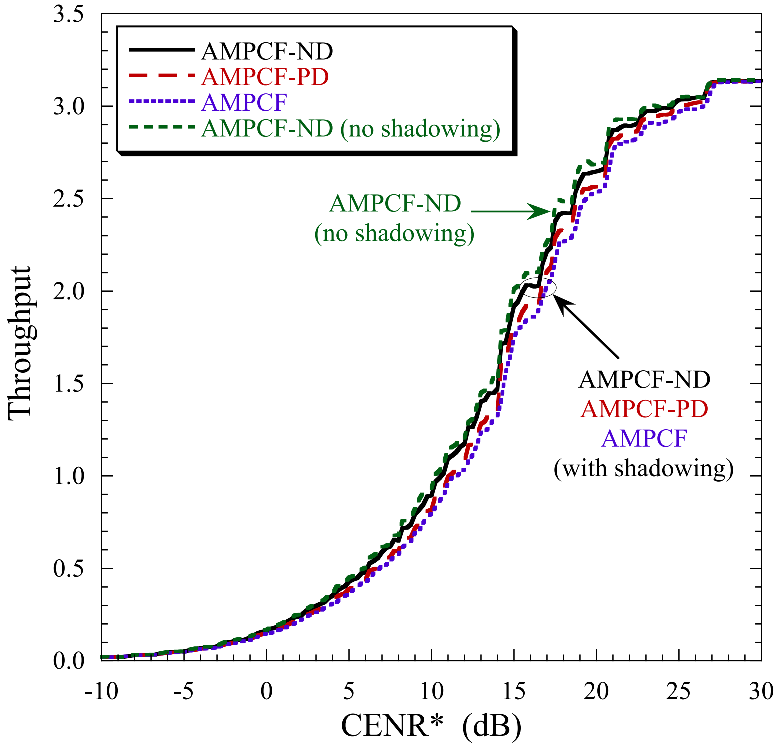


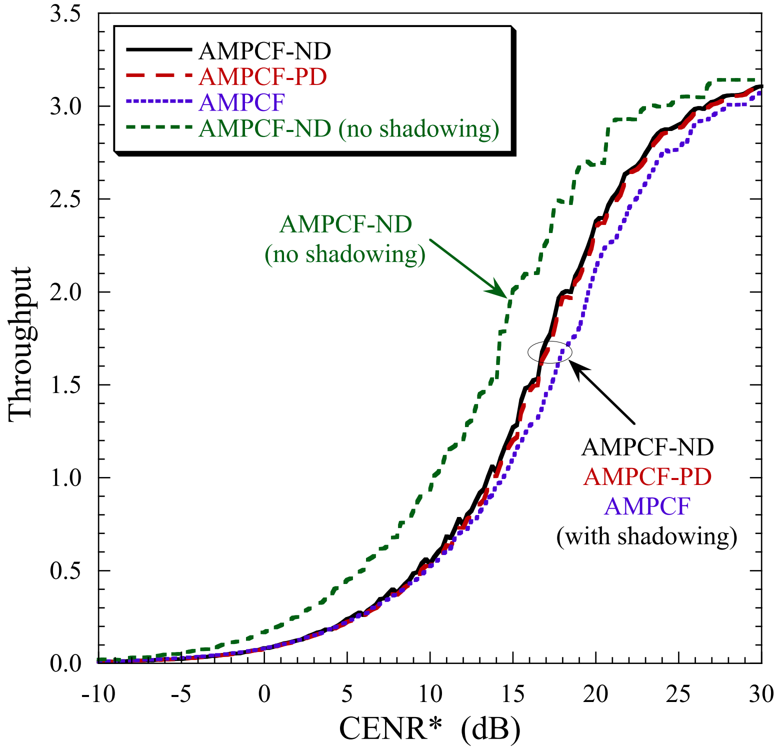
Figure 3.4: Throughput results for the AMPCF protocol and three fixed code-modulation combinations ($K = 100$, $m = 1$, $f_d T_s = 0.02$).

than our practical protocol for CENR$^* > 9\,$dB, but it provides zero throughput for CENR$^* < 4\,$dB, and our protocol achieves much better performance at low SNR. Also, QPSK with the 0.793 rate code outperforms our protocol at mid-range CENR*, but it provides zero throughput for CENR$^* < -2\,$dB, and our protocol improves upon the throughput of this combination at high CENR* by nearly 100%.

We also consider channels that experience shadowing in addition to fading. Our Markov chain models for shadowing are described in Sec. 2.3. In Fig. 3.5, the throughput for the AMPCF protocol is shown for a channel with Nakagami-m fading with $m = 1$ and $f_d T_s = 0.02$ as well as shadowing modeled by a two-state Markov chain (Fig. 3.5(a)) or a three-state Markov chain (Fig. 3.5(b)). For Fig. 3.5(a), the channel experiences a 5 dB shadow loss 4% of the time. Our practical AMPCF protocol achieves nearly the same performance as the hypothetical protocols in this



(a) Two-state Markov chain for shadowing ($\Delta = 5\,\text{dB}$).



(b) Three-state Markov chain for shadowing ($\Delta = 2.5\,\text{dB}$).

Figure 3.5: Throughput results for the AMPCF protocol for channels with fading and shadowing ($K = 100$, $m = 1$, $f_d T_s = 0.02$).

scenario. The throughput is also shown for the AMPCF-ND protocol for a system with fading, but no shadowing. The presence of shadowing 4% of the time has little effect on the performance of the adaptive protocol. In contrast, for the three-state Markov chain to model shadowing in Fig. 3.5(b) with all states equally likely, the protocol suffers much more performance degradation. The practical protocol also does not achieve the same level of performance in comparison to the hypothetical protocols in this scenario. The AMPCF-PD protocol outperforms the practical AMPCF protocol by up to 15% in Fig. 3.5(b); however, the AMPCF protocol has performance within 5% of the AMPCF-PD protocol for all other results in this section. The large variations in channel conditions from packet to packet due to the increased likelihood of shadowing, combined with the restriction of adaptive modulation only during the transmission of a frame, makes mistakes caused by the imperfect channel state information more costly to the performance of the practical AMPCF protocol. This is especially true if the practical protocol makes the wrong decision for the rate of the channel code at the start of a frame.

CHAPTER 4
INTRA-FRAME ADAPTIVE MODULATION AND CHANNEL CODING

4.1 Protocol Description

The AMPCF protocol described in Chapter 3 uses a fixed channel code for an entire frame and adapts only the modulation format from packet to packet. In some cases, adjusting the rate of the channel code may be more beneficial to the throughput performance of the system than adjusting the modulation. Our second adaptive protocol removes the restriction of a fixed channel code for a frame, and performs intra-frame adaptive modulation and channel coding or *adaptive modulation and channel coding from packet to packet*, which we refer to as the AMCP protocol.

The AMCP protocol is also designed for systems that have fixed-length channel packets, and the same L channel codes and W modulation formats available to the AMPCF protocol are also available to the AMCP protocol. This allows for LW possible code-modulation combinations, and the objective of the adaptive protocol is to select the combination of channel code and modulation that optimizes the throughput for the next packet transmission. The throughput achieved by one code-modulation combination may be outperformed by another combination for all signal-to-noise ratios, or it may add little to no benefit to the overall performance of the system. As a result, a subset of the LW possible combinations $\mathcal{B}_2 = \{B_2(h) : 1 \leq h \leq H_2\}$, where $H_2 \leq LW$, is chosen to serve as the library of code-modulation combinations available to the adaptive protocol. The details of the procedure for selecting this subset are described in [16]. Typically H_2 is much smaller than LW. For example, for the set of channel codes and modulation formats employed by our protocols, $LW = 25$ and $H_2 = 13$. As shown in Table A.1 in Appendix A, the use of 16-BOK is eliminated altogether from $\mathcal{B}_2$. We found that the use of 16-BOK with any of the channel codes always resulted in a combination whose performance was inferior to that of another combination in the set $\mathcal{B}_2$. The set $\mathcal{B}_1$ for the AMPCF protocol consists of all LW code-modulation combinations. The AMPCF protocol does not have the freedom to choose from any of the LW combinations after every packet transmission. It is restricted to a fixed channel code for the duration of a frame and adaptive modulation only for all channel packets within a frame, so it benefits from the availability of all W modulation formats while transmitting a frame. The AMCP protocol is permitted to choose any code-modulation combination for each packet transmission, so we are able to reduce the complexity of the system by reducing the number of possible combinations from LW to H_2 without any noticeable degradation in the performance.

The choice of the code-modulation combination is governed by receiver statistics for our practical AMCP protocol. The demodulator statistics described in Chapter 3 can be used to select the proper code-modulation combination for the next packet transmission. The *error count* (EC) is a lower-complexity statistic that may also be used to provide control information for the AMCP protocol. The EC is the number of hard-decision binary code symbols that are in error at the output of the demodulator. For binary or nonbinary modulation, the receiver makes hard decisions at the demodulator to obtain binary symbols for the channel packet and compares the results to the actual binary code symbols in the channel packet. The number of binary symbols that are in error at the demodulator output is the receiver's EC. Even though the channel decoder may use soft decisions from the demodulator to decode channel packets, the EC is based on hard decisions, and the errors occur at the output of the demodulator, not the channel decoder. Because the correct code symbols for the channel packet must be known to make the comparison, the EC can only be determined if a channel packet is correctly decoded. The EC does not provide the same level of accuracy as the demodulator statistics, so it is not suitable for the intra-frame adaptive modulation protocol in Chapter 3. It is also limited in the number of steps in code-modulation combinations it allows the protocol to take with each statistic; however, it is easier to compute than a demodulator statistic.

The choice of the code-modulation combination for the next packet is the result of an interval test applied to the most recent EC or demodulator statistic. If the AMCP protocol uses demodulator statistics, it is denoted by AMCP-DS. If it uses the EC, then it is denoted by AMCP-EC. Both the EC and demodulator statistics depend on the modulation format used to transmit the packet. As a result, the decision intervals, $\{\mathcal{I}_p(h) : 1 \leq h \leq H_2\}$, depend on the modulation format used to transmit the previous packet, $\mathcal{M}_p$. If the receiver statistic falls in $\mathcal{I}_p(h)$, then combination $B_2(h)$ is chosen as the code-modulation combination for the next packet transmission. In the event of a packet failure, the EC cannot be obtained, and the AMCP-EC protocol selects the code-modulation combination with the next-lowest information rate if there is one available. In other words, the AMCP-EC protocol selects combination $B_2(\hat{h})$ for the next packet transmission, where $\hat{h} = \max\{1, h-1\}$. Demodulator statistics can still be computed if the packet fails to decode, so the interval test is applicable for each packet reception. More details on the interval tests used for our numerical results for AMCP-DS and AMCP-EC protocols are given in Appendix A.

Adapting the channel code along with the modulation format while a frame is in progress increases the complexity of the system. The size of the data packets in a frame depends on the channel code selected by the adaptive protocol to transmit the channel packets, so a change in the channel code requires a change in the frame from which information packets are drawn. We let $S_i[n]$ denote the nth *source frame* formatted with code C_i as its channel code. At the start of a session, the AMCP protocol selects a code-modulation combination to transmit the first packet. If code

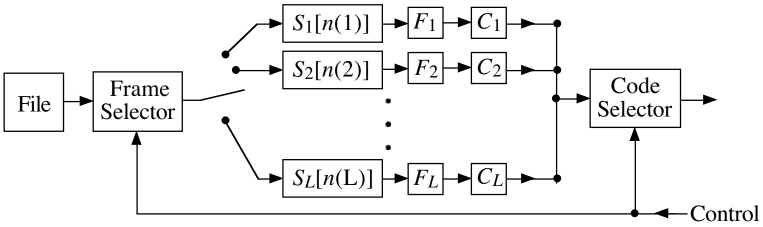


Figure 4.1: Channel code and frame selection for the AMCP protocol.

C_i is the channel code for that combination, then source frame $S_i[1]$ is generated with K information packets of length m_i. The fountain encoder is applied, and the source begins transmitting data packets to the destination. If the adaptive protocol elects to change to a combination whose channel code is C_ℓ before the destination has received enough data packets to recover frame $S_i[1]$, then the source stops transmitting packets from frame $S_i[1]$ and forms a new source frame, $S_\ell[1]$, containing K new information packets of length m_ℓ. These information packets now serve as the inputs to the fountain encoder, and the source begins transmitting data packets from source frame $S_\ell[1]$. If the adaptive protocol again elects to change the channel code to a code other than C_i or C_ℓ, then a new source frame is generated corresponding to the new channel code. Eventually, the AMCP protocol may be operating with as many source frames as there are channel codes. Each time the protocol chooses to change the channel code, the source transmits a data packet from the corresponding frame. Once the destination has received enough data packets to recover a source frame, then that source frame is discarded by the source, and a new source frame is generated the next time the channel code is selected.

This procedure is illustrated in Fig. 4.1. In the figure, $S_i[n(i)]$ represents the $n(i)$th source frame generated for channel code C_i, and source frames have been generated for each channel code. The control information for the adaptive protocol determines which channel code will be used for the next packet, and consequently, the source frame from which the next packet will be drawn. If the control information indicates that C_i should be used for the next packet transmission, then the frame selector activates source frame $S_i[n(i)]$, and the corresponding fountain encoder, F_i, encodes a data packet to be transmitted. The channel code selector activates the channel encoder for code C_i, and the channel packet is transmitted over the channel with the modulator chosen by the AMCP protocol. If the channel packet is decoded correctly by the channel decoder at the destination, then the data packet is stored in the *destination frame* $D_i[n(i)]$. If $D_i[n(i)]$ contains at least K data packets, then the fountain decoder F_i attempts to recover the source frame of information packets. If the fountain decoder successfully recovers the source frame, then the information bits are delivered to a higher layer and the destination frame $D_i[n(i)]$ is discarded. If the control information for the protocol indicates that code C_i should be used for the next packet, then the source generates frame $S_i[n(i) + 1]$, forms the data packet with encoder F_i and encodes it with C_i, and transmits the channel packet. If the control information indicates that the channel code should be changed, then the source switches to the corresponding frame and applies the fountain code.

At the end of the session when there are no more information bits in the file to be delivered, no new frames can be generated. There are a number of termination procedures that the protocol can implement to deliver any source frames that have yet to be delivered to the destination. For example, suppose that the protocol selects code C_i to be used, and the source frame $S_i[n(i)]$ is empty. If there is another file that needs to be transmitted, then one possibility is to use information bits from the next file to generate a new source frame for C_i. Alternatively, the protocol may elect to reduce the index of the combination until a combination can be found whose channel code has a nonempty source frame. If there is no such combination, but there are nonempty source frames for higher-rate combinations, then the protocol can reallocate information bits from nonempty source frames to source frames corresponding to a lower-rate combination. Alternatively, the protocol may switch to the intra-frame adaptive modulation stage of the AMPCF protocol from Chapter 3 to deliver any nonempty source frames. In our performance results, we do not account for the termination procedure at the end of the session because the resulting overhead is strongly dependent upon the file size, frame size, channel codes, and channel behavior, among other factors.

In addition to the increase in complexity required to orchestrate the transmission of multiple frames, another disadvantage of the AMCP protocol is the additional memory required to store the data for multiple frames. Also, switching frames may result in the information bits not being delivered to the destination in the order with which they were generated at the source. If the order in which data is delivered or memory constraints are of importance to a particular application, the the AMPCF protocol or the protocol described in Chapter 5 may be more suitable for the application, even if the AMCP protocol provides higher throughput.

4.2 Performance Benchmarks

The hypothetical AMCP protocols are given the same channel state information as those described in Section 3.2 and are told either the perfect channel state information for the *next* packet transmission (AMCP-N), or the perfect channel state information for the *previous* packet reception (AMCP-P). These benchmark protocols use their knowledge of the channel conditions to adapt the modulation and channel code from packet to packet. The session throughput performance for the AMCP-N and AMCP-P protocols can be computed analytically using a procedure similar to the analysis presented in (3.1)–(3.10). The AMCP protocols select a code-modulation combination from the set $\mathcal{B}_2 = \{B_2(h) : 1 \leq h \leq H_2\}$ to be used to transmit each packet, and the expected throughput for a packet sent with combination $B_2(h)$ when the channel is in state j is given by $s[h|j]$, where the channel code and modulation index pair (i, k) in (3.1) has been replaced with code-modulation combination index h. The AMCP-N protocol knows that the channel will be in state j when the next channel packet is transmitted, so it chooses combination $h(j)$ to transmit the next packet if

$$s[h(j)|j] = \max\{s[h|j] : 1 \leq h \leq H_2\}. \tag{4.1}$$

The resulting average session throughput for the AMCP-N protocol is given by

$$\mathcal{S}_{N,2} = \frac{\sum_{j=0}^{J-1} \pi_j s[h(j)|j]\tau[h(j)]}{\sum_{j=0}^{J-1} \pi_j \tau[h(j)]}. \tag{4.2}$$

The AMCP-P protocol knows that the channel was in state j when the previous packet was transmitted, and it employs (4.1) to make its decision for the combination index for the next packet transmission based on its knowledge of the previous state. However, the next packet will actually be transmitted when the channel is in state k; therefore, the average session throughput for the AMCP-P protocol is given by

$$\mathcal{S}_{P,2} = \frac{\sum_{j=0}^{J-1} \pi_j \tau[h(j)] \sum_{k=0}^{J-1} q(k|j) s[h(j)|k]}{\sum_{j=0}^{J-1} \pi_j \tau[h(j)]}. \tag{4.3}$$

4.3 Performance Results

The AMCP protocol is more complex than the AMPCF protocol described in Chapter 3, and it is important to determine whether the increase in complexity is justified by a potential increase in throughput. To accomplish this, it is sufficient to compare the analytical results for the hypothetical versions of the two adaptive

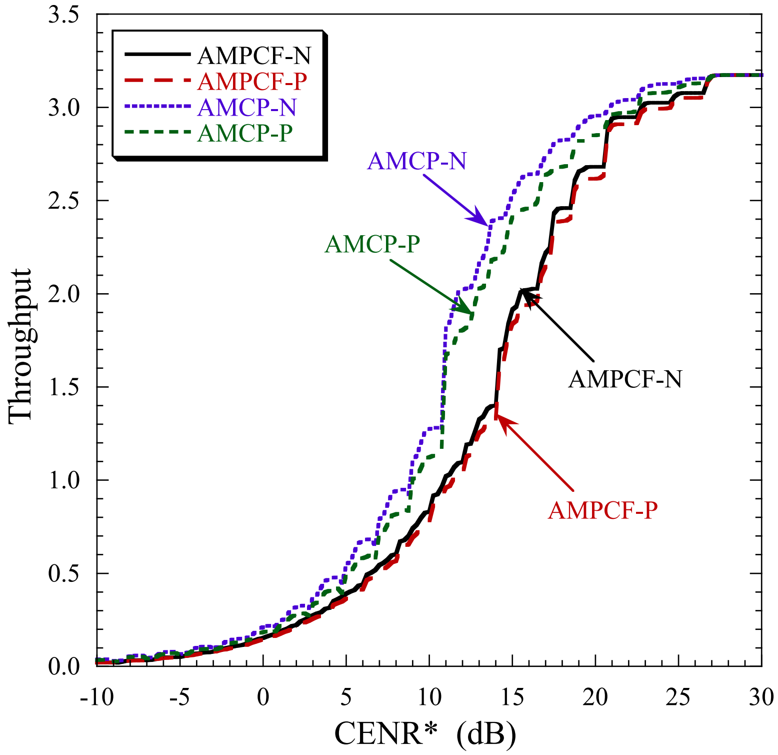


Figure 4.2: Comparison of the analytical results for the hypothetical AMCP and AMPCF protocols ($K = 100$, $m = 1$, $f_d T_s = 0.02$).

protocols. One of the key advantages of obtaining analytical results for the performance of the protocols is the ability to make such preliminary comparisons quickly and easily, and they are ideally suited for this scenario.

In Fig. 4.2, we compare the results obtained from (3.6) and (3.10) for the AMPCF protocol, with results obtained from (4.2) and (4.3) for the AMCP protocol for a Nakagami-m fading channel with $m = 1$ and $f_d T_s = 0.02$. The AMCP protocol provides a significant throughput gain over the AMPCF protocol for almost all values of CENR* considered in Fig. 4.2. For example, at CENR$^* = 11\,\text{dB}$, the AMCP-N protocol achieves a session throughput of approximately 1.82 bits per chip, and the AMCP-P protocol achieves a throughput of approximately 1.67 bits per chip. At the same CENR*, the AMPCF-N and AMPCF-P protocols achieve a throughput of approximately 1.02 bits per chip and 0.96 bits per chip respectively. That represents

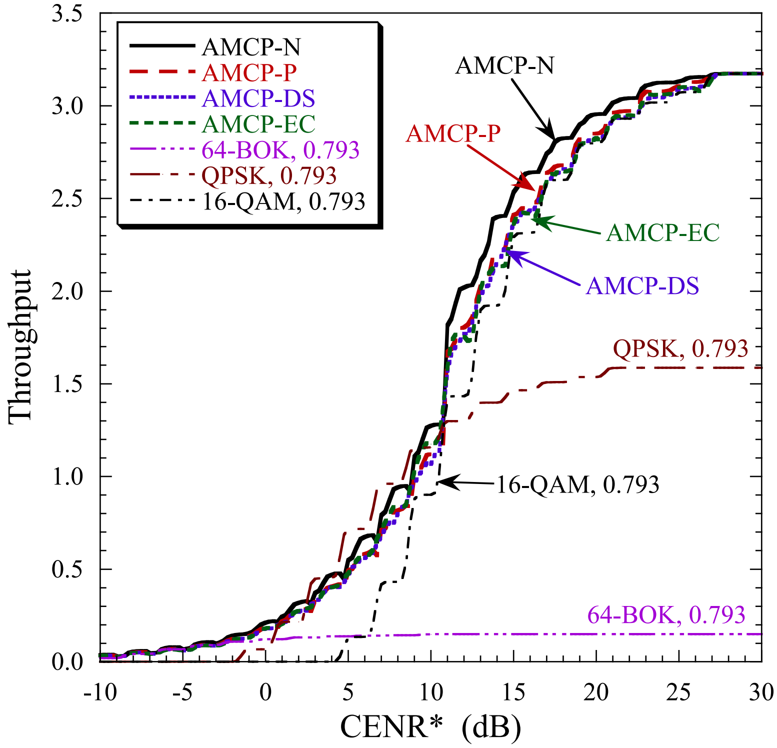


Figure 4.3: Throughput results for the AMCP protocol and three fixed code-modulation combinations ($K = 100$, $m = 1$, $f_d T_s = 0.02$).

a throughput gain of over 70% for the hypothetical AMCP protocols over the corresponding AMPCF protocols. In many applications, the benefits to the performance of the system may justify the additional complexity of the AMCP protocol over the AMPCF protocol.

The analytical results for the hypothetical AMCP protocols and the simulation results for the practical AMCP-DS and AMCP-EC protocols are shown in Fig. 4.3, along with the throughput for three systems that employ fixed code-modulation combinations from the set $\mathcal{B}_2$. The AMCP-DS and AMCP-EC achieve nearly the same performance as the hypothetical AMCP-P protocol, indicating that there is not much benefit to using more sophisticated techniques for estimating the channel conditions to obtain better channel state information. Fig. 4.3 also demonstrates the poor performance that can result from the use of fixed modulation and channel coding

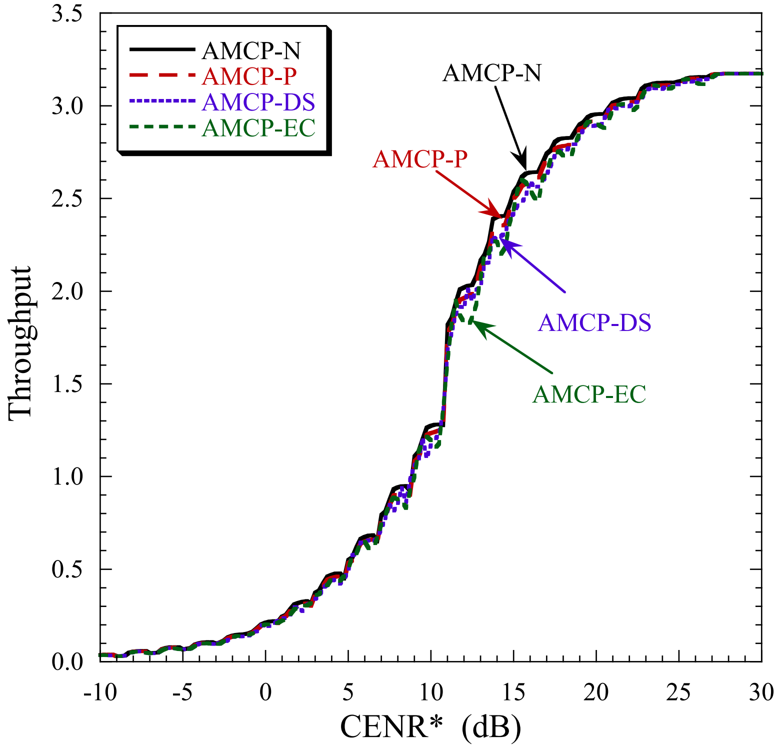
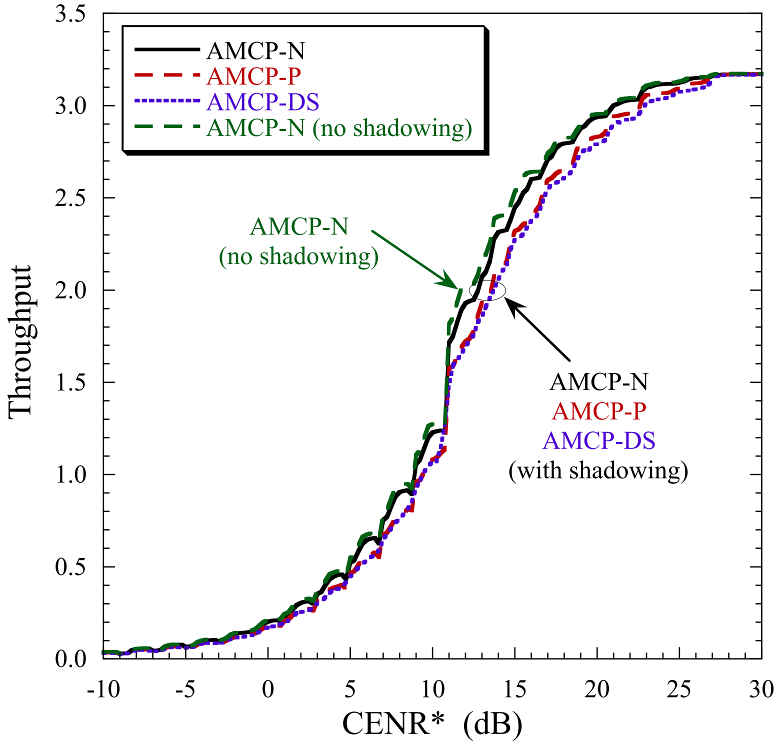


Figure 4.4: Throughput results for the AMCP protocol ($K = 100$, $m = 1$, $f_d T_s = 0.005$).

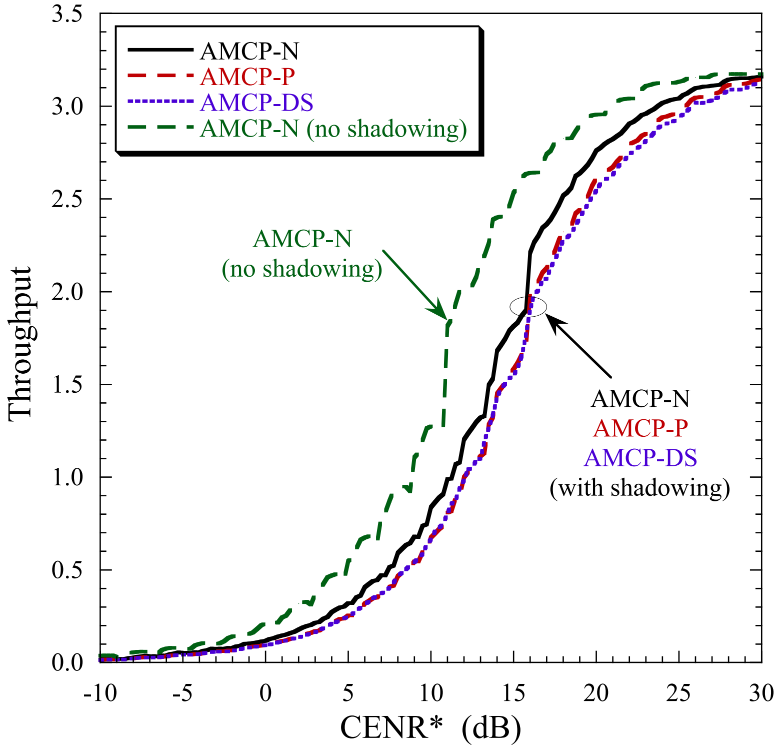
schemes. The fixed combinations are $B_2(3)$, $B_2(10)$, and $B_2(13)$ from Table A.1 in Appendix A, and each of the three combinations has poor performance for some range of CENR* in Fig. 4.3. For example, combination $B_2(10)$ consisting of QPSK modulation with the rate 0.793 TPC, provides high throughput in the range of CENR* from 1 dB to 11 dB, but compared to what is possible, it has poor performance for CENR* above 11 dB, and it provides zero throughput for CENR* below -2 dB.

With slower fading, as illustrated in Fig. 4.4, the AMCP protocol still achieves good performance for the full range of CENR*. Results are shown for the AMCP protocols for a Nakgami-m fading channel with $m = 1$ and $f_d T_s = 0.005$. The throughput gain for the AMCP-N protocol over the AMCP-P protocol and the practical protocols is reduced for slower fading. With a smaller value for the normalized Doppler frequency, the previous channel state is more highly correlated with the next channel state, so the benefit of having perfect information for the next state is reduced. As a result, the practical AMCP-DS and AMCP-EC protocols have approximately the same performance as the AMCP-P protocol, and their performance even approaches that of the AMCP-N protocol, even though they rely on imperfect information about the previous channel state.

Next we demonstrate the performance of our protocol for channels that experience fading and shadowing. In Fig. 4.5, the throughput for the AMCP protocol is shown for a channel with Nakagami-m fading with $m = 1$ and $f_d T_s = 0.02$ as well as our two shadowing scenarios described in Section 2.3. For Fig. 4.5(a), the channel experiences a 5 dB shadow loss 4% of the time. Our practical AMCP-DS protocol achieves nearly the same performance as the AMCP-P protocol in this scenario, and the presence of shadowing 4% of the time has little effect on the performance of the adaptive protocol, as demonstrated by the throughput curve for the AMCP-N protocol for a channel with no shadowing. For the three-state Markov chain to model shadowing in Fig. 3.5(b), with all states equally likely, the protocol suffers much more performance degradation in comparison to the channel with no shadowing. However, increasing the likelihood of shadowing does not disrupt the AMCP-DS protocol from keeping up with the variability in the channel conditions and achieving performance that agrees closely with the hypothetical AMCP-P protocol.



(a) Two-state Markov chain for shadowing ($\Delta = 5\,\text{dB}$).



(b) Three-state Markov chain for shadowing ($\Delta = 2.5\,\text{dB}$).

Figure 4.5: Throughput results for the AMCP protocol for channels with fading and shadowing ($K = 100$, $m = 1$, $f_dT_s = 0.02$).

CHAPTER 5
ADAPTIVE MODULATION AND CHANNEL CODING WITH FIXED-LENGTH DATA PACKETS

5.1 Protocol Description

The AMPCF and AMCP protocols described in Chapters 3 and 4 required all channel packets to have the same length. If that restriction is removed, the intra-frame adaptive modulation and channel coding protocol described in Chapter 4 can be simplified by allowing fixed-length data packets and variable-length channel packets. The fixed-length data packets in the system allows the protocol to adapt both the modulation and channel code after each packet transmission without requiring different source frames for each channel code. We refer to our third adaptive protocol for unicast transmission that performs *adaptive modulation and channel coding with fixed-length data packets* as the AMCFL protocol. The AMCFL protocol operates in the same manner as the AMCP protocol without the additional complexity of switching between multiple source frames, the additional memory required for storing multiple source frames and destination frames at the receiver, or the requirement of a termination procedure to complete the delivery of a file at the end of a session.

For AMCFL, we employ a new set of channel codes that can accommodate the fixed-length data packets. There are still L channel codes, $\{C_i : 1 \leq i \leq L\}$, indexed in order of increasing rate, available to the adaptive protocol for adaptive channel coding. Code C_i has rate $r_i = k_i/n_i$, where k_i is the number of information bits, and n_i is the block length. The number of information bits in each information packet and each data packet is η; therefore, for the formation of a channel packet depicted in Fig. 2.1, we set $m_i = \eta$. When code C_i is used, the number of information bits per codeword, k_i, is a divisor of η, and each data packet consists of information bits from $Q_i = \eta/k_i$ channel codewords. The resulting length of the channel packet is $N_i = Q_i n_i$.

The same modulation formats available to the AMPCF and AMCP protocols are also available to the AMCFL protocol. The W modulation formats and L channel codes form LW code-modulation combinations. Some combinations do not provide much benefit to the performance of the protocol, and the set of LW combinations is reduced to the set $\mathcal{B}_3 = \{B_3(h) : 1 \le h \le H_3\}$. The AMCFL protocol selects the code-modulation combination to transmit the next packet in the same manner as the AMCP protocol; however, the AMCFL protocol may use any channel code to deliver the η bits in each data packet, eliminating the need for multiple source frames. Demodulator statistics or the EC can be used to adapt the modulation and channel coding using the same type of interval test described in Chapter 4 for the AMCP protocol. The details of the interval test are given in Appendix A. The channel codes employed by the AMCFL protocol are TPCs; however, they differ from the channel codes employed by the AMCP protocol and are slightly weaker, which results in lower throughput compared to the AMCP protocol.

5.2 Performance Benchmarks

The hypothetical AMCFL protocols are given perfect channel state information and emulate the behavior of the practical AMCFL protocol. The AMCFL-N protocol is told what state the channel will be in when the *next* packet is transmitted, and the AMCFL-P protocol is told the state of the channel when the *previous* packet was received. These protocols use their information to maximize the expected throughput for the next packet transmission. The AMCFL-N and AMCFL-P protocols select the code-modulation combination with index $h(j)$ if

$$s[h(j)|j] = \max\{s[h|j] : 1 \leq h \leq H_3\}, \tag{5.1}$$

where the j in (5.1) represents the next state for the AMCFL-N protocol and the previous state for the AMCFL-P protocol. The same analytical expressions used to compute the session throughput for the hypothetical AMCP protocols can be used for the AMCFL protocols as well. The expression for the session throughput for AMCFL-N protocol, $\mathcal{S}_{N,3}$, is (4.2), and the expression for $\mathcal{S}_{P,3}$, the throughput for the AMCFL-P protocol, is (4.3).

5.3 Performance Results

The AMCFL protocol provides the benefit of adapting the modulation and channel coding after each packet transmission, without the complexity of transmitting multiple frames. A comparison of the throughput performance for all three of our adaptive protocols for unicast transmission is shown in Fig. 5.1 for a channel with Nakagami-m fading with $m = 1$ and $f_dT_s = 0.02$. Analytical results for the hypothetical protocols given perfect information for the previous channel state are shown along with simulation results for the practical protocols that use demodulator statistics for their operation. The EC provides nearly the same performance for the AMCP and AMCFL protocols; however, the EC is not suitable for the AMPCF protocol, so we chose to use demodulator statistics for the protocol comparison. The simulation results for the practical protocols agree closely with the analytical results for each of the three protocols. The AMCP protocol provides the highest throughput performance for almost all values of CENR* in Fig. 5.1, but it is also the most complex of the three protocols. The AMCFL operates in the same manner as the AMCP protocol, but it

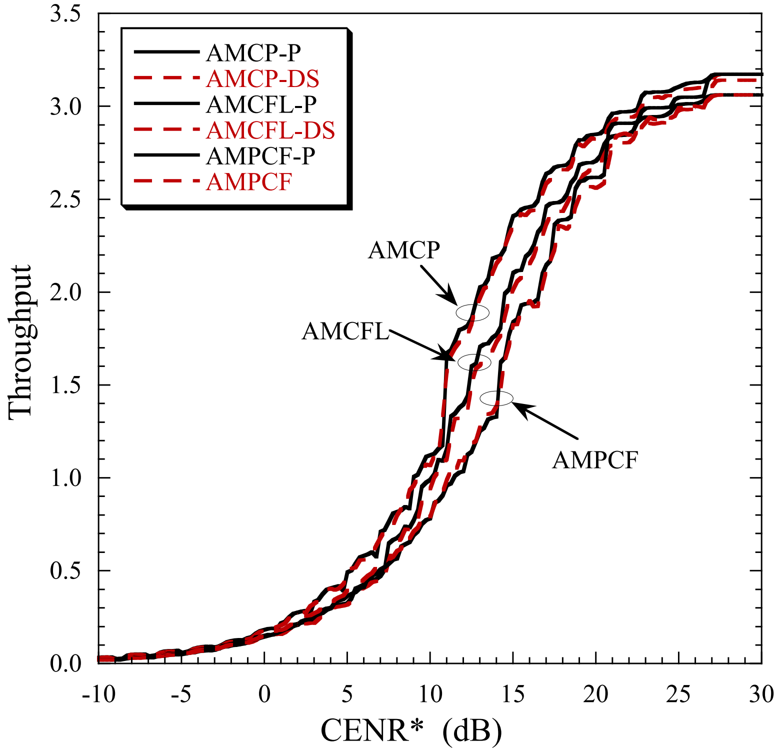


Figure 5.1: Analytical benchmarks and simulation results for all three adaptive protocols ($K = 100$, $m = 1$, $f_d T_s = 0.02$).

does not obtain the same level of performance, in part because the TPCs employed by the AMCFL protocol do not achieve performance as close to capacity as the TPCs employed by the AMCP protocol. However, the AMCFL protocol outperforms the AMPCF protocol, even with the weaker channel codes. Another disadvantage for the AMCFL protocol is the large transmission time required by the lower-rate codes with longer block lengths. The extra time that the source must spend transmitting packets due to the longer channel packets, particularly when using biorthogonal modulation at low SNR, can be detrimental to the session throughput performance of the protocol, even if the low-rate combination may provide the highest throughput for a single packet transmission. In Chapter 6 we present a modification to our adaptive protocols that addresses this issue.

In Fig. 5.2, the throughput performance is shown for the AMCFL protocol and

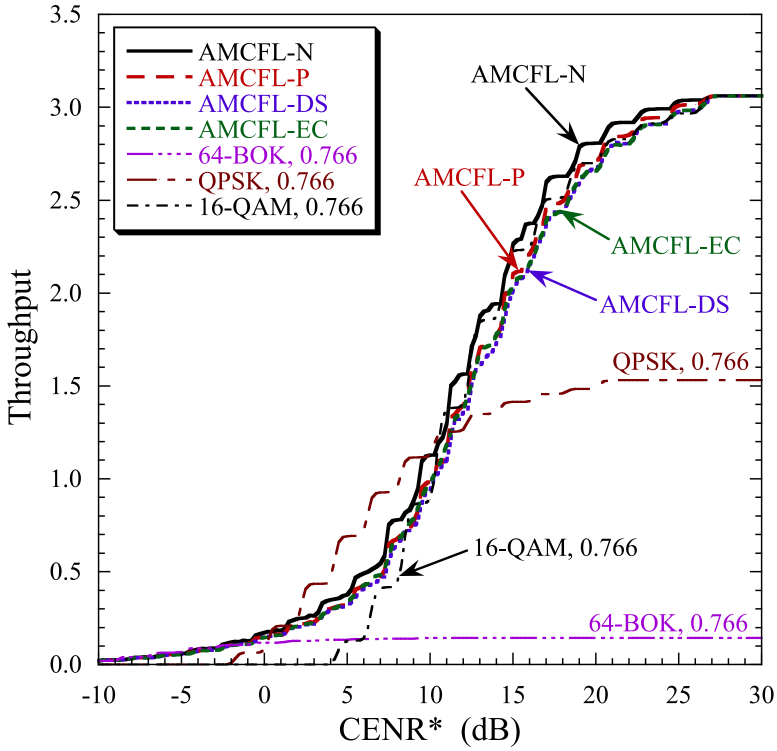


Figure 5.2: Throughput results for the AMCFL protocol and three fixed code-modulation combinations ($K = 100$, $m = 1$, $f_dT_s = 0.02$).

three of the fixed code-modulation combinations for $m = 1$ and $f_dT_s = 0.02$. The fixed combinations are $B_3(3)$, $B_3(10)$, and $B_3(13)$. Each of the fixed combinations has poor performance for a significant range of CENR*, while the AMCFL protocol performs well for the entire range of CENR*. The performance for the analytical benchmarks as well as the two practical protocols is shown in Fig. 5.2, and the simulation results for the practical protocols agree closely with the analytical results for the AMCFL-P protocol. The same is true for the performance of the practical protocols shown in Fig. 5.3 for a slower fading channel with $m = 1$ and $f_dT_s = 0.005$. With slower fading, the practical protocols perform nearly as well as the AMCFL-N protocol, even though they rely on the EC or demodulator statistics obtained during reception of the previous packet for their control information. In Figs. 5.2 and 5.3, there is little difference in throughput between the AMCFL-EC and AMCFL-DS protocols.

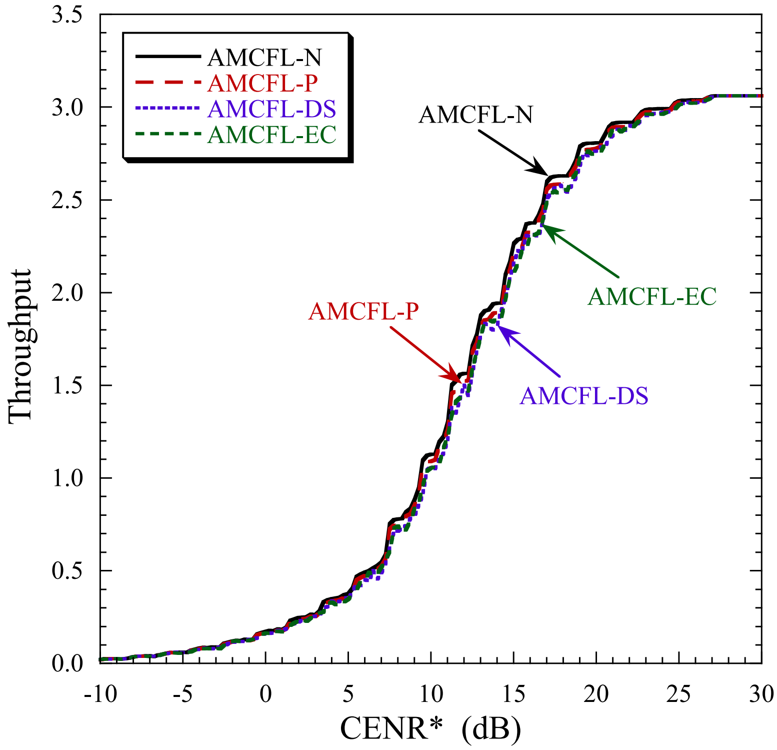


Figure 5.3: Throughput results for the AMCFL protocol ($K = 100$, $m = 1$, $f_d T_s = 0.005$).

When the fading is less severe, the AMCFL protocol achieves even better performance, as illustrated in Fig. 5.4 for a channel with $m = 3.25$ and $f_d T_s = 0.02$. The performance is shown for the analytical benchmarks, the practical AMCFL-DS protocol, and four fixed combinations. Each of the fixed combinations has poor performance for a large range of CENR*, and the AMCFL protocol outperforms the fixed-rate schemes for almost all values of CENR* considered in Fig. 5.4. Also, for less severe fading, we once again see close agreement between the performance of the AMCFL-DS protocol and the analytical results for the benchmark protocols. The same is true for channels that experience shadowing in addition to fading. In Fig. 5.5, the performance for the AMCFL protocol is shown for channels with Rayleigh fading and shadowing modeled by the two shadowing scenarios described in Section 2.3. As with our other adaptive protocols, our practical protocol is able to success-

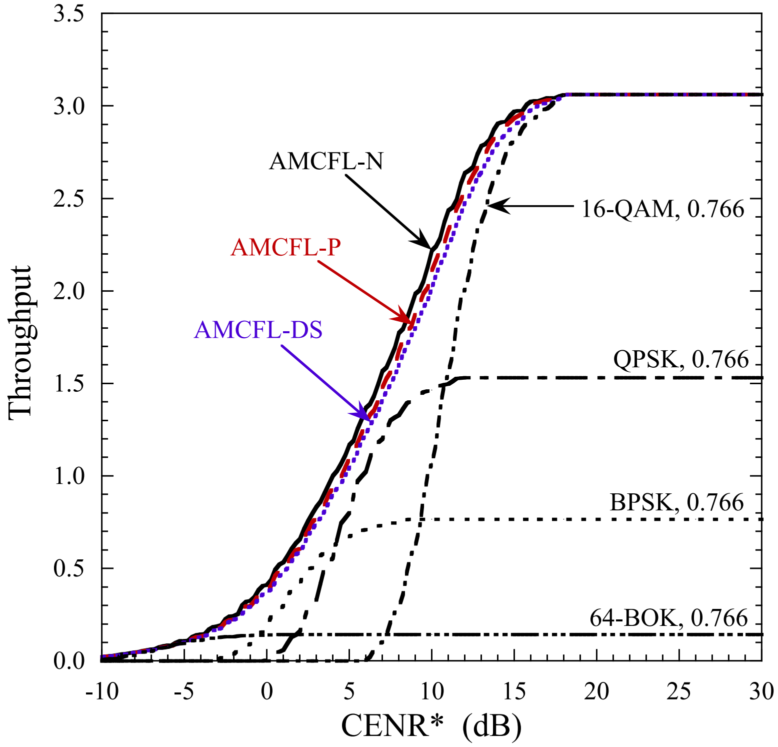
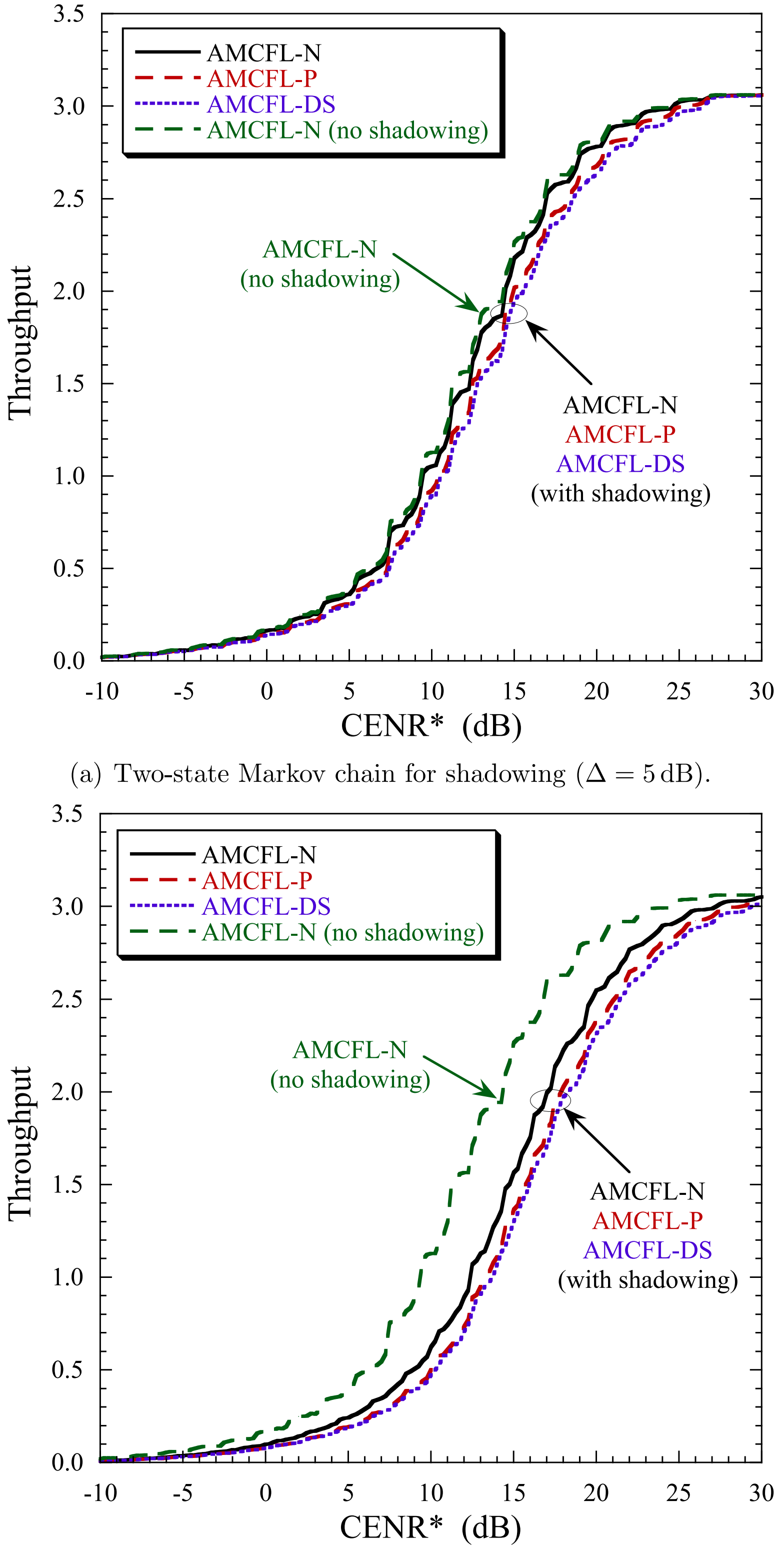


Figure 5.4: Throughput results for the AMCFL protocol and four fixed code-modulation combinations ($K = 100$, $m = 3.25$, $f_d T_s = 0.02$).

fully adapt the modulation and channel coding using only demodulator statistics, and it achieves performance that agrees closely with that of the hypothetical AMCFL-P protocol.



(a) Two-state Markov chain for shadowing ($\Delta = 5\,\text{dB}$).

(b) Three-state Markov chain for shadowing ($\Delta = 2.5\,\text{dB}$).

Figure 5.5: Throughput results for the AMCFL protocol for channels with fading and shadowing ($K = 100$, $m = 1$, $f_d T_s = 0.02$).

CHAPTER 6
GENERALIZATIONS AND MODIFICATIONS

6.1 Intermittent and Delayed Feedback

For all of our performance results presented in Chapters 3–5, we assume that the destination provides feedback after each packet transmission. Now suppose that the source receives *intermittent feedback*, and the destination only provides feedback every νth packet. This is a generalization of our prior results, where all of our previous results are for the case when $\nu = 1$. Intermittent feedback for our protocols is denoted by adding the suffix ν to the abbreviation. For example, AMCFL-P-ν indicates that the AMCFL protocol receives perfect channel state information for the previous packet after every νth packet transmission, and AMCFL-ν indicates that our practical protocol receives a demodulator statistic for the previous packet after every νth packet transmission.

The analytical expression for the throughput of the hypothetical AMCP-P and AMCFL-P protocols for $\nu = 1$ is (4.3). In this section, we generalize this expression to give the analytical expression for the throughput of the AMCP-P-ν and AMCFL-P-ν protocols for any positive integer ν. A hypothetical protocol that receives intermittent feedback knows that the channel state was j when packet $t - \nu$ was received by the destination, and it chooses combination $B_n(h(j))$ to transmit packets $t-\nu+1$, $t-\nu+2$, $\ldots$, t if

$$s[h(j)|j] = \max\{s[h|j] : 1 \leq h \leq H_n\}, \tag{6.1}$$

where $n = 2$ for the AMCP-P-ν protocol and $n = 3$ for the AMCFL-P-ν protocol. The code-modulation combination is only adapted every νth packet when the source receives feedback; however, the state of the channel may change after each packet transmission. For the generalized version of (4.3), we must determine the i-step transition probability function q_i, where $q_i(k|j)$ denotes the probability that the channel is in state k for packet t given that the channel was in state j for packet $t-i$. We let $\mathbf{Q}$ be the $J \times J$ matrix of one-step transition probabilities. We can obtain the i-step transition probabilities by the matrix multiplication of $\mathbf{Q}$. The i-step transition probabilities are the entries of the matrix $\mathbf{Q}^i$, which is the i-fold product of $\mathbf{Q}$ with itself. The average number of information bits delivered per packet by the hypothetical protocol whose perfect channel state information is updated every νth packet is given by

$$\overline{I}_{P,n}^{(\nu)} = \nu^{-1} \sum_{j=0}^{J-1} \pi_j \tau[h(j)] \sum_{k=0}^{J-1} s[h(j)|k] \sum_{i=1}^{\nu} q_i(k|j). \tag{6.2}$$

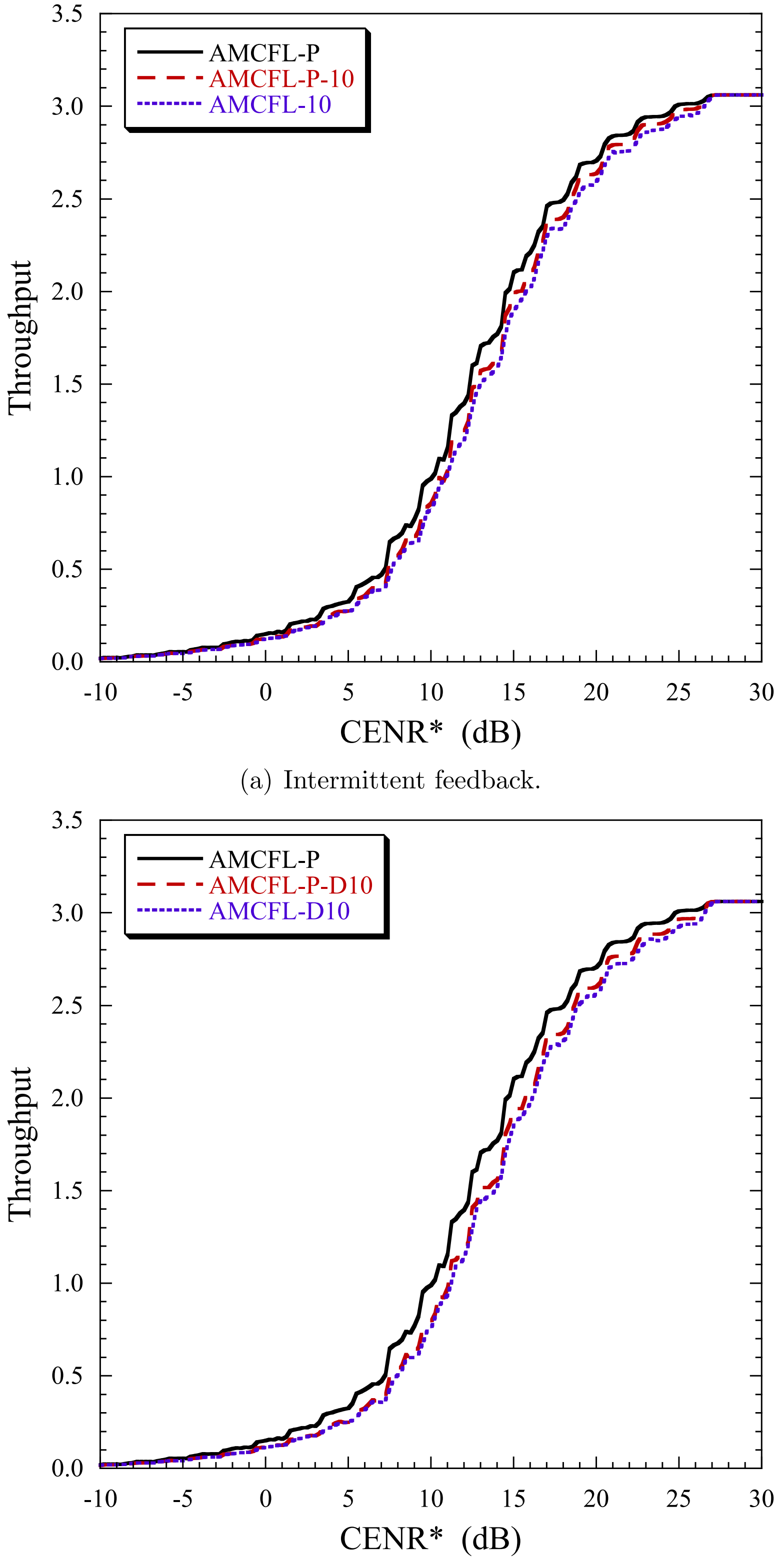
The resulting average session throughput for the AMCP-P-ν and AMCFL-P-ν protocols is

$$\mathcal{S}_{P,n}^{(\nu)} = \frac{\overline{I}_{P,n}^{(\nu)}}{\sum_{j=0}^{J-1} \pi_j \tau[h(j)]}. \tag{6.3}$$

A similar generalization of (3.10) can be used to obtain an analytical expression for the session throughput of the AMPCF-P-ν protocol.

For a second generalization of our adaptive protocols, we assume that the destination provides feedback after each packet, but there is a delay of μ packets before the source receives the feedback from the destination. For a protocol that receives *delayed feedback*, when the source prepares to transmit packet t, it receives feedback for the destination for packet $t-\mu$. We denote a feedback delay of μ packets by the suffix Dμ, so AMCFL-P-Dμ indicates that the AMCFL protocol receives perfect channel state information for packet $t-\mu$ before transmitting packet t, and AMCFL-Dμ indicates that the AMCFL protocol receives the demodulator statistic for packet $t-\mu$ before packet t is transmitted. The analytical expressions for the session throughput of the AMPCF-P-Dμ, AMCP-P-Dμ, and the AMCFL-P-Dμ protocols can easily be obtained by substituting the μ-step transition probability function q_μ into (3.10) and (4.3) for the one-step transition probability; thus, the term $q(k|j)$ becomes $q_\mu(k|j)$, which is the probability that the channel is in state k for packet t given that it was in state j for packet $t - \mu$.

The throughput performance for the AMCFL protocol with intermittent and delayed feedback is shown in Fig. 6.1. The results for the AMPCF and AMCP protocols are similar. We compare the performance of these protocols to the hypothetical AMCFL-P protocol that is always given the state of the channel for the previous packet transmission (i.e., $\nu = \mu = 1$). In Fig. 6.1(a), results are shown for the analytical benchmark and the practical AMCFL protocol using demodulator statistics with intermittent feedback for $\nu = 10$, and in Fig. 6.1(b), we show the performance of the protocol with a feedback delay of $\mu = 10$ packets. Comparisons between the AMCFL-P and AMCFL-P-10 protocols illustrate the reduction in throughput due to intermittent feedback. There is some degradation in performance due to the intermittent feedback, but the AMCFL protocol still works reasonably well, even with less feedback information from the destination. The practical AMCFL-10 protocol that uses demodulator statistics for the its operation still performs nearly as well as the hypothetical AMCFL-P-10 protocol that has perfect channel state information. Comparisons between the AMCFL-P and AMCFL-P-D10 protocol in Fig. 6.1(b) show the degradation due to delayed feedback information. The protocol has slightly more degradation in its performance with delayed feedback than with intermittent feedback; however, the performance is still competitive with the AMCFL-P protocol, even though all of its feedback information is outdated. Also, the performance of the AMCFL-D10 protocol agrees closely with that of its hypothetical counterpart.



(a) Intermittent feedback.

(b) Delayed feedback.

Figure 6.1: Analytical benchmarks and simulation results for the AMCFL protocol with intermittent and delayed feedback ($K = 100$, $m = 1$, $f_d T_s = 0.02$).

6.2 Pseudorandom Sequence Transmission

Biorthogonal modulation is included in the sets $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ to provide an acceptable packet error probability when the channel conditions are very poor, so that the source is able to maintain communication with the destination. A single packet transmitted with biorthogonal modulation requires several time units of transmission time, while only providing a small number of delivered information bits. As a result, sending a packet with biorthogonal modulation may provide the highest throughput for that packet transmission, but it could be detrimental to the session throughput. For example, 64-BOK requires 5.33 modulation chips per binary symbol, while QPSK requires only 0.5 modulation chips per binary symbol, so the amount of time required to transmit a packet with 64-BOK is over ten times the amount of time to transmit the same packet with QPSK. Thus, given the same channel code, a system using QPSK will achieve higher throughput than a system using 64-BOK, even if every 64-BOK-modulated packet is successful and only one out of every ten QPSK-modulated packets is successful. From this analysis, we may conclude that for some channel conditions, the session throughput performance is actually improved by using higher-rate modulation schemes and transmitting shorter packets, even if many of these packets fail to decode. One example of such channel conditions is a severe fading channel, such as Rayleigh fading. In fact, in terms of session throughput, it would be better for the source not to transmit at all when channel conditions are so poor that biorthogonal modulation must be used to have an acceptable packet error rate. However, this is not feasible because if the source is not transmitting, then the destination is not computing receiver statistics, and the source will not know when channel conditions have improved to the point where it should continue transmission.

We investigated a modification to our adaptive protocols that permits the source to transmit a pseudorandom sequence (PRS) of information bits to maintain communication in poor channel conditions without wasting resources. This sequence should be of short duration and known by both the source and destination. For our numerical results, we eliminate all code-modulation combinations that employ biorthogonal modulation to obtain the modified sets of combinations $\mathcal{B}'_1$, $\mathcal{B}'_2$, and $\mathcal{B}'_3$. The eliminated combinations are replaced by a PRS of 1024 binary digits that is transmitted with BPSK. For example, the BPSK-modulated PRS replaces the first three entries in Table A.1 of Appendix A to give the modified sets of code-modulation combinations. Because the PRS is known to the transmitter and receiver, either demodulator statistics or the EC may be used for the control information for the AMCP and AMCFL protocols. The AMPCF protocol requires demodulator statistics for intra-frame adaptive modulation. For all three protocols, if the receiver statistic falls in an interval that corresponds to a combination that employs biorthogonal modulation, then the PRS is transmitted instead. The next time the receiver statistic falls in the interval for a higher-rate combination, indicating that channel conditions have improved, each of the protocols continues transmitting channel-encoded data packets using the selected code-modulation combination.

The performance of the raptor code with AMCFL and pseudorandom sequence transmission (AMCFL-PRS) is shown in Fig. 6.2, along with the performance of the AMCFL protocol that uses the full set $\mathcal{B}_3$. The channel is a Nakagami-m fading channel with $m = 1$, which is severe fading, and $f_d T_s = 0.02$. Both protocols in Fig. 6.2 use demodulator statistics for their operation. The PRS transmission significantly improves the throughput of the system at low SNR, and the protocol's performance nearly achieves the upper envelope of the fixed-rate coding schemes that use combinations $B_3(3)$, $B_3(10)$, and $B_3(13)$. Similar improvement to the throughput can be achieved by employing PRS transmission with the AMPCF and AMCP protocols.

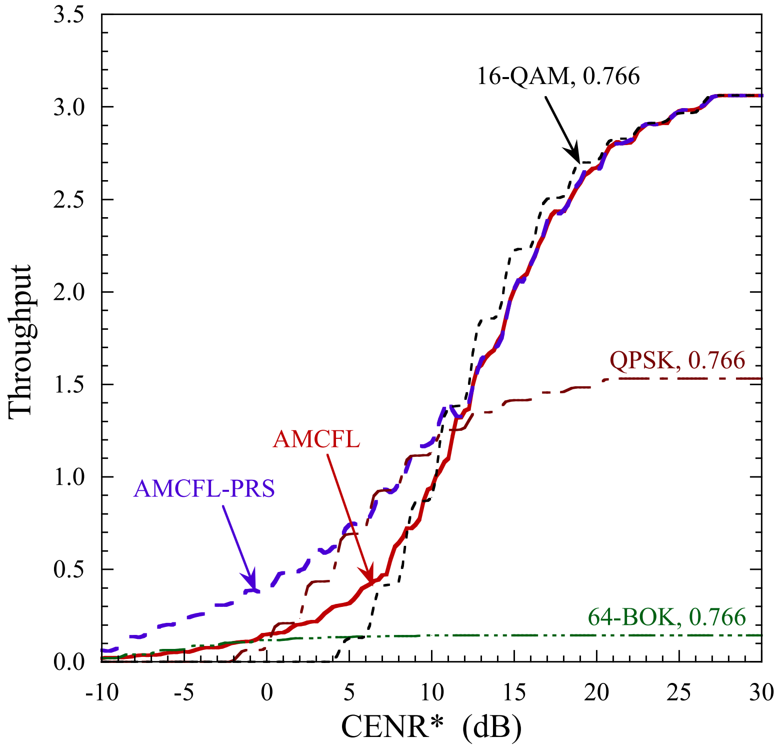


Figure 6.2: Throughput comparison for the AMCFL-PRS and AMCFL protocols and three fixed combinations ($K = 100$, $m = 1$, $f_d T_s = 0.02$.

Biorthogonal modulation is useful if channel conditions are very poor for a very long time; however, if channel conditions are not likely to be poor for very long, the modification of PRS transmission can improve the session throughput performance of the adaptive protocol. It should be noted that a backup procedure should be used in the event that the protocol will send the PRS for several consecutive packets. In practice, the channel conditions may not be accurately modeled by a Nakagami-m fading channel, or any other fading process. If channel conditions are poor for a very long time (e.g., the entire duration of a frame), the PRS will provide zero throughput, and the protocol must return to its original set of combinations to ensure that communication is not disrupted.

CHAPTER 7
MULTICAST SYSTEM MODEL

7.1 Adaptive Multicast Transmission

In the design of adaptive transmission protocols, reliable multicast transmission presents many challenges that do not arise in unicast transmission. One challenge inherent in multicast transmission for any wireless network is the result of communicating over multiple links with time-varying channels. At any given time, these links may be experiencing very different channel conditions. One destination may be experiencing good channel conditions and will request a modulation and coding scheme with a high information rate. The communication link for another destination may be suffering from poor channel conditions at the same time, so that destination will request a code-modulation combination with a low information rate. As with unicast transmission, a fixed modulation and channel coding scheme does not provide efficient communication in a multicast system, even if fountain coding is employed for erasure correction. An adaptive transmission protocol for reliable multicast transmission must adapt to dynamic channel conditions on multiple links, while ensuring that all destinations successfully receive the intended information.

Another challenge in the design of adaptive multicast transmission protocols is balancing the the amount of necessary feedback information for the adaptation with the amount of time it takes to obtain feedback from multiple destinations. For half-duplex packet radios, the source radio cannot receive feedback from any destinations while it is transmitting packets; therefore, the time spent gathering feedback messages is overhead time in which the source is not transmitting packets. One extreme is to allow all destinations to reply to the source after each packet transmission. This is referred to as *full reporting*, and it provides the maximum amount of feedback information but also results in the maximum overhead time. The amount of time required to obtain feedback from all destinations is proportional to the number of destinations in the multicast network, so for a large number of destinations, the amount of time obtaining feedback could easily exceed the amount of time spent transmitting packets. Full reporting would also require a channel-access protocol for the feedback channel to avoid collisions amongst the reply messages. A channel-access protocol for the feedback channel would increase the complexity of the system and may be difficult to implement, even for a moderate number of destinations.

In order to avoid the added complexity and overhead time of full reporting, our protocols employ *single-destination reporting*. With single-destination reporting, only one destination radio is allowed to report feedback after each packet transmission. This minimizes the amount of overhead time required to obtain feedback; however, it also reduces the amount of feedback information supplied to the adaptive protocol. As the number of destinations increases, the amount of time between consecutive reports from each destination also increases. Single-destination reporting also requires a scheduling protocol to determine which destination responds to which packet transmission.

We will present two adaptive protocols for multicast transmission of fountain-coded packets that are designed with these challenges in mind. Our protocols respond to variations in the channel conditions for multiple links by adapting the modulation and channel coding on a packet-by-packet basis. The selection of the modulation and channel code for one of our protocols is based on a technique often utilized in multicast systems with no fountain coding. With no erasure correction coding at a higher layer, all destinations in the multicast network must receive a packet correctly; otherwise, it must be retransmitted. As a result, our adaptive multicast transmission protocol attempts to choose a modulation and coding scheme that satisfies the *most disadvantaged destination*, or the destination whose link is experiencing the worst channel conditions [9]. If the most disadvantaged destination is able to receive packets successfully, then the other destinations with higher capacity links will be able to as well. If fountain coding is employed for packet erasure correction, then a successfully decoded packet may be useful to a destination even if other destinations were not able to decode the packet. Focusing on the most disadvantaged destination results in the protocol transmitting at low information rates too often in a system that employs fountain coding. Thus, our second protocol is designed to maximize the amount of data being delivered to all destinations, regardless of which destinations are receiving the data. We will show that our adaptive protocol utilizing this technique can improve upon the performance of our protocol that satisfies the most disadvantaged destination in systems that employ fountain coding. Our performance results demonstrate that both of our adaptive protocols are capable of achieving high throughput in multicast systems with a moderate number of destinations whose communication links experience large, independent variations in channel conditions. We also demonstrate that single-destination reporting with simple scheduling is sufficient to provide the necessary control information for the operation of our adaptive protocols.

7.2 Multicast Channel Model

For a multicast session in a packet radio network, a source radio must deliver a large file to D destinations over D time-varying communication links. The multicast channel is illustrated in Fig. 7.1 with D forward links. There are also D reverse links in the multicast system that enables the source to collect feedback information from each of the destinations. The fountain code is implemented in the same manner as described in Section 2.1. The source divides the file into frames of K information packets. The multicast session concludes when all destinations have notified the source that they have successfully received enough data packets to decode the complete file or the last frame in the file.

For multicast transmission, our protocols perform adaptive modulation and channel coding with fixed-length data packets. Thus, each information packet and data packet has length η, and $m_i = \eta$ in Fig. 2.1 for each i. The adaptive protocols with fixed-length channel packets described in Chapters 3 and 4 are not suitable for multicast transmission. The restrictions of adaptive modulation only within a frame and adaptive channel coding between frames for the AMPCF protocol will cause the throughput to suffer even more in a multicast system than in a unicast system. The AMCP protocol becomes too complex for multicast systems. The source must keep track of all destinations that have incomplete data frames for all channel codes, which requires more memory. Also, the source is not able to generate a new data frame for a given channel code until all destinations have decoded the current frame corresponding to that channel code. This restriction eliminates the advantage of being able to adapt both the modulation and channel code after each packet. The AMCFL protocol does not have these disadvantages and can easily be implemented in a multicast system. The set of code-modulations combinations available to our adaptive multicast transmission protocols is the set $\mathcal{B}_3 = \{B_3(h) : 1 \leq h \leq H_3\}$ of combinations for the AMCFL protocol.

For our numerical results, the D communication links in our multicast channel are modeled by independent identical J-state Markov chains that model Nakagami-m fading on each of the links. These are the same Markov chain models developed in [15] that we employed for unicast transmission. Independent Markov chains are used to model the links in the multicast channel because independently fading links creates

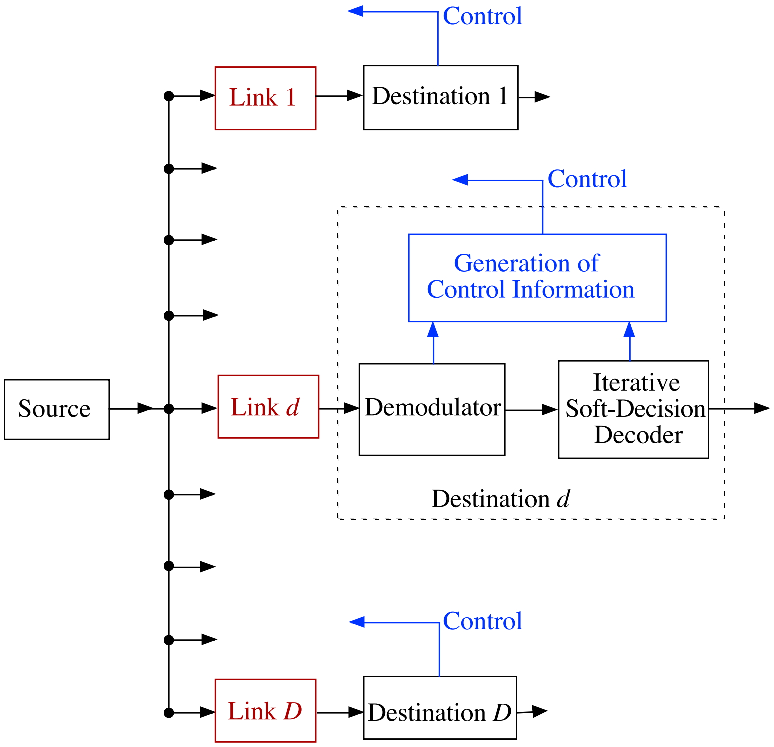


Figure 7.1: Model for multicast transmission.

a more challenging environment for an adaptive multicast transmission protocol than links with correlated fading. With independent fading on the links to each of the D destinations, the information obtained about one of the links reveals nothing to the adaptive protocol about the conditions of the links to the other destinations, and it does not help the protocol to select a good code-modulation combination for the other links.

In practice, the propagation losses on the D communication links may not all be the same, even in the absence of fading. Differences in communication ranges, varying terrain, or the presence of shadowing may cause variation in the ranges of propagation losses for each of the links. As a result, we introduce a *quality offset* parameter to model disparities in the link qualities of different destinations. The multicast channel model is illustrated in Fig. 7.2. The Markov chains modeling fading on each of the

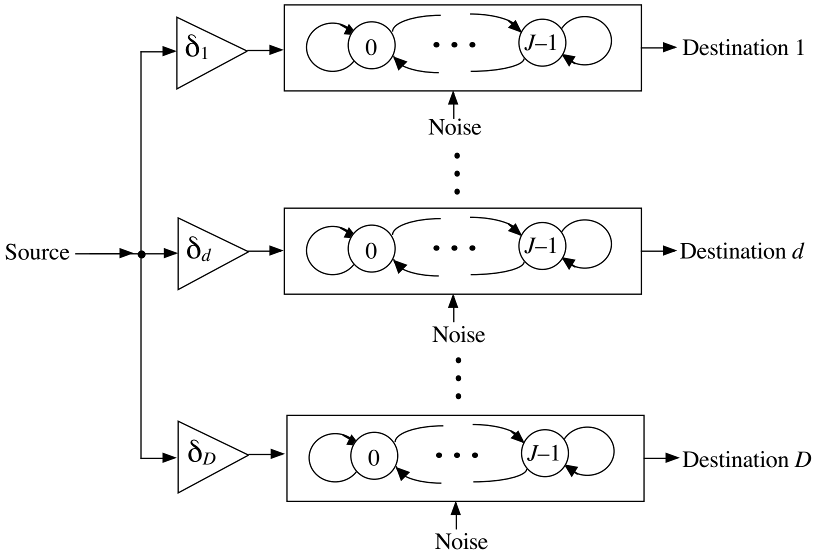


Figure 7.2: Multicast channel model with independent fading Markov chains and link quality offsets.

D communication links have identical parameters, and the jth state of the Markov chain for each link corresponds to an AWGN channel with channel gain G_j in dB. We let δ_d be the quality offset for the SNR of the dth link. If CENR* is the received SNR in the absence of fading with no quality offset, the the value of CENR corresponding to state j of the dth link is

$$\text{CENR}_{d,j} = \text{CENR}^* + G_j + \delta_d. \tag{7.1}$$

The quality offset parameter δ_d in (7.1) is one of two values for each link in the multicast channel. If link d is a *standard link*, then $\delta_d = 0$. If link d is a *superior link*, then $\delta_d = \delta > 0$.

The quality offset allows us to evaluate the performance of our protocols in multicast channels where some links have much better link qualities than others for the entire duration of a session. One of the potential causes for variations in the link qualities is the presence of shadowing. We can also use Markov chains to model shadowing, as described in Section 2.3. By introducing an independent Markov chain on each link to model shadowing in addition to the fading Markov chain, we can investigate the performance of our protocol in multicast channels where the link qualities vary over time. We adopt the same notation of Section 2.3 and consider a single J-state Markov chain, where $J = J_f J_s$, that represents both independent Markov chains for fading and shadowing. Thus, if j_1 is the state of the fading Markov chain and j_2 is the state of the shadowing Markov chain, then $\mathbf{j} = (j_1, j_2)$ is the state vector for the composite Markov chain. Each state of the Markov chain corresponds to an AWGN channel with channel gain $G_{\mathbf{j}}$ in dB, and the value of CENR corresponding to state $\mathbf{j}$ is

$$\mathrm{CENR}_{\mathbf{j}} = \mathrm{CENR}^* + G_{\mathbf{j}}. \tag{7.2}$$

Note that with no quality offset parameter there is no dependence on the link of the multicast channel, so all of the links are modeled by identical independent Markov chains.

CHAPTER 8
ADAPTIVE MULTICAST TRANSMISSION PROTOCOLS

In this chapter, we present and evaluate two practical adaptive protocols for reliable multicast transmission in packet radio networks. Our protocols provide efficient point-to-multipoint communication by adapting the modulation and channel coding used to transmit fountain-coded packets to the destinations in the multicast system. The destinations must allocate some feedback information to the source to serve as the control information for the adaptive protocols; however, it is not feasible for all destinations to report feedback after each packet transmission. Our protocols employ single-destination reporting, which means after each packet transmission, a single destination is designated to be the *reporting destination.* The reporting destination for each packet is identified by a field in the packet header. Our protocols use a very simple reply schedule to determine the reporting destination for each packet that we refer to as *round-robin reporting.* Each destination is selected in a round-robin fashion to be the reporting destination, so each destination reports feedback to the source once every Dth packet. Round-robin reporting minimizes the overhead time required to collect feedback messages, but it also increases the response time of the adaptive protocols. Our performance results show that for a moderate number of destinations, our protocols suffer very little degradation from the use of round-robin reporting instead of full reporting.

The operation of our adaptive multicast transmission protocols is governed by the same simple receiver statistics used for our unicast protocols. Adaptive modulation and channel coding with fixed-length data packets can be achieved using demodulator statistics or the error count. Suppose that destination d is designated to be the reporting destination for a packet. We let E_d denote the receiver statistic computed by destination d during reception of the packet. An interval test is applied to E_d using the procedure described for the AMCFL protocol in Appendix A to determine the index I_d of the code-modulation combination that destination d suggests should be used for the next packet transmission. If modulation $\mathcal{M}_j$ is the modulation for the previous packet (or in the case of the error count, if combination $B_3(j)$ is the code-modulation combination for the previous packet), and $E_d \in \mathcal{I}_j(\ell)$, then combination $B_3(\ell)$ is the combination that destination d suggests for the next packet. The *suggested index* is $I_d = \ell$. If the error count is the employed statistic, and the channel packet is not successfully received by destination d, then the suggested index is $I_d = \max\{1, j-1\}$.

The source stores the suggested indexes for each of the D destinations in an index table. Each time the source receives a suggested index from destination d, it updates the index table with the most current value for I_d. Once destination d has received enough data packets for the fountain decoder to decode the frame, the next time it is selected as the reporting destination, its feedback message includes a notification that it has decoded the frame to go along with its suggested index for the previous packet. When the source is notified of its successful decoding of the frame, destination d is removed from the reporting schedule and its suggested index is no longer considered in the selection of the code-modulation combination for the remaining channel packets in the frame. The transmission of a frame concludes when all D destinations have reported to the source that they have successfully decoded the frame. In the following sections, we will describe the selection criteria used by our two protocols to determine the code-modulation combination for the impending channel packet.

8.1 Maximum Data Recovery Rate

The criteria used by our adaptive protocols to select the code-modulation combination for each packet are based on performance measures for a single packet transmission. As with unicast transmission, because the protocols must rely on feedback from the destinations on a packet-by-packet basis and has no knowledge of future states of the multicast channel or the channel parameters, they can only attempt to maximize a given measure of performance for the next packet transmission, not the entire session.

The first selection criterion is a measure of the total amount of data delivered to the destinations in the multicast system. We say that a data bit is *recovered* by a destination if the channel packet that contains the data bit is successfully decoded by the destination's channel decoder. Thus, a single data bit is recovered if and only if all of the data bits in a packet are recovered. The total amount of data delivered to the destinations in the multicast system is simply the number of data bits recovered by destination d summed over all D destinations in the system. We let D_t denote the number of destinations that are able to successfully decode the tth channel packet. Each data packet contains η data bits, so the *number of data bits recovered in the multicast network* from the tth packet is ηD_t. We define the *single-transmission data recovery rate* to be the number of data bits recovered in the multicast network for a single packet transmission divided by the number of modulation symbols required to transmit the packet. If combination $B_3(h)$ is used to transmit the tth channel packet and D_t is the number of destinations whose channel decoders are able to decode the packet, then the single-transmission data recovery rate is $\eta D_t/\tau(h)$, where $\tau(h)$ is the number of modulation chips required to transmit a packet with combination $B_3(h)$.

The *maximum data recovery rate* (max-DRR) protocol attempts to maximize the single-transmission data recovery rate for each packet transmission. The practical max-DRR protocol uses the index table generated from round-robin feedback from each destination to select the code-modulation combination that is expected to provide the maximum single-transmission data recovery rate for the next packet transmission. If I_d is the most recent value in the index table for destination d, then the protocol assumes that destination d will be able to decode a channel packet transmitted with any combination whose index is less than or equal to I_d. Thus, the number of destinations expected to decode a packet transmitted with combination $B_3(h)$ is $W(h) = |\{d : I_d \geq h\}|$, where $|\mathcal{U}|$ denotes the cardinality of the set $\mathcal{U}$, and the expected single-transmission data recovery rate for combination $B_3(h)$ is $\mathcal{R}(h) = \eta W(h)/\tau(h)$. The max-DRR protocol chooses combination $B_3(m)$ for the next packet transmission if

$$\mathcal{R}(m) = \max\{\mathcal{R}(h) : 1 \leq h \leq H_3\}. \tag{8.1}$$

If there is a tie when determining $\mathcal{R}(m)$, then the protocol selects the combination with the highest information rate among those that achieve the maximum in (8.1). For example, consider a multicast session with two destinations. Suppose that the current entries in the index table form the suggested index pair $(I_1, I_2) = (4, 6)$. Using the parameters for the combinations in the set $\mathcal{B}_3$ given in Table A.1 of Appendix A, it is easy to show that evaluating the maximum expected data recovery rate in (8.1) results in a tie, with $R(4) = R(6) \approx 0.521$. In this example, the max-DRR protocol chooses combination $B_3(6)$ to transmit the next packet.

The max-DRR protocol attempts to deliver as many data bits as possible to the destinations in the shortest possible time, without regard to the distribution of the data bits among the destinations. The protocol is designed for multicast systems that have fading on the communication links to the destinations and employ fountain coding for packet erasure correction. The protocol's strategy will at times result in paying less attention to the destinations that have poor links. Destinations experiencing poor channel conditions may fall behind when collecting data, but the hope is that the fading on the links of the multicast channel will be such that all destinations have good channel conditions for a significant period of time during the session, and the fountain code is able to correct any packet erasures that occur when the links are poor. We demonstrate in Chapter 9 that with fountain coding, the max-DRR protocol is able to achieve good session throughput performance in multicast systems; however, it has poor performance in multicast systems that do not employ fountain coding because all of the data packets that are erased by the slighted destinations must be retransmitted.

We compare the performance of our practical max-DRR protocol with that of hypothetical protocols that are given different levels of perfect channel state information. The first protocol is the max-DRR-N protocol, and it is told the state of each link of the multicast channel for the *next* packet transmission. Equivalently, the max-DRR-N protocol has full reporting of the perfect channel state information for the next packet transmission; however, we do not penalize the hypothetical protocols in our performance results for having full reporting. The second hypothetical protocol is the max-DRR-P protocol, and it is told what state each link was in when the *previous* packet was transmitted. These hypothetical protocols emulate the practical max-DRR protocol and use their knowledge of the state of the multicast channel to form an index table of suggested indexes for each destination. The suggested index for destination d, I_d, is chosen deterministically by the hypothetical protocol and is the index corresponding to the code-modulation combination that provides the maximum throughput for the next state of link d or the previous state, depending on the protocol being used. The protocols then use their index tables to select the code-modulation combination for the next packet transmission by applying (8.1). The max-DRR-N and max-DRR-P protocols have a decided advantage over the practical max-DRR protocol because the practical protocol is given no channel state information, and it must maintain its index table based on imperfect, outdated information. The performance results for the hypothetical max-DRR protocols presented in Chapter 9 require simulation of the Markov chain models for the fading on each link to the D destinations, but they do not require simulation of the D iterative decoders required to decode the channel packets. The performance can be computed by combining the Markov chain simulations with the packet error probabilities for each code-modulation combination for a static AWGN channel corresponding to each state of the Markov chain model. These packet error probabilities can be obtained through offline simulations.

8.2 Minimum Suggested Index

The second selection criterion for designing an adaptive multicast transmission protocol is the *single-transmission D-node throughput.* It is a measure of the amount of *common* data that is delivered to *all* destinations in the multicast system. If a channel packet is successfully decoded by the channel decoders at every destination, then the single-transmission D-node throughput for the packet is equal to the number of data bits in the packet divided by the number of modulation symbols required to transmit the packet. If a channel packet fails to decode at any of the D destinations, then the single-transmission D-node throughput for the packet is zero. Thus, if combination $B_3(h)$ is used to transmit the tth channel packet, and D_t is the number of destinations that are able to decode the packet, then the single-transmission D-node throughput for the tth packet is $\eta/\tau(h)$ if $D_t = D$, and it is zero if $D_t < D$.

The *minimum suggested index* (min-index) protocol attempts to maximize the single-transmission D-node throughput for each packet transmission. As described in Section 8.1 for the max-DRR protocol, the practical min-index protocol relies on its index table maintained through round-robin feedback from each destination and assumes that destination d is able to decode a channel packet transmitted with a combination whose index is less than or equal to I_d. The min-index protocol selects combination $B_3(m)$ for the next packet transmission if

$$m = \min\{I_d : 1 \le d \le D\}. \tag{8.2}$$

The min-index protocol is designed to satisfy all destinations, so it focuses on the most disadvantaged destination for each packet transmission. The idea is that if the destination whose link is experiencing the worst channel conditions is able to decode a packet, then the other destinations will be able to decode the packet as well. This strategy works well for multicast systems that do not employ fountain coding because packets that fail to decode at any destination must be retransmitted [9]. However, with fountain-coded transmissions, there is no need for retransmissions, and the min-index protocol is at a disadvantage because it frequently elects to transmit at low rates to accommodate destinations with poor links.

We compare the performance of our practical min-index protocol that relies on round-robin reporting of receiver statistics for its operation, with hypothetical protocols that are given perfect channel state information for all links in the multicast network for each packet transmission. The min-index-N protocol knows the state that each link will be in when the next packet is transmitted, and the min-index-P protocol knows the state that each link was in when the previous packet was transmitted. The hypothetical protocols use their knowledge of the multicast channel to formulate an index table and select the code-modulation corresponding to the minimum suggested index to transmit the next packet. We will demonstrate with performance results in Chapter 9 that our practical min-index protocol performs nearly as well as these hypothetical protocols in fountain-coded multicast systems.

CHAPTER 9
MULTICAST PERFORMANCE RESULTS

9.1 Adaptive Multicast Transmission Without Fountain Coding

We first consider the performance of our adaptive multicast transmission protocols in systems with no fountain coding. For such a system, we can compute the session throughput performance analytically for our hypothetical adaptive multicast transmission protocols. We define session throughput for a reliable multicast system to be the number of information bits delivered to *all* destinations divided by the number of modulation chips required for the source to transmit enough channel packets for all destinations to recover the fountain-coded frame containing the information bits. Let the state of the Markov chain for link d be denoted by v_d. We define $\mathbf{v} = (v_1, v_2, \ldots, v_D)$ to be a *multicast channel state vector*, and the set of possible multicast channel state vectors is $\{\mathbf{v}_j : 0 \leq j \leq J^D - 1\}$. The steady-state probability for multicast channel state $\mathbf{v}_j$ is denoted by π_j. If the hypothetical protocol is told that the next multicast channel state is $\mathbf{v}_j$, then it selects combination $B_3(m_j)$ if it satisfies (8.1) for the max-DRR-N protocol or (8.2) for the min-index-N protocol. The average session throughput for the hypothetical protocol given perfect information about the next multicast channel state is

$$\mathcal{S}_N = \frac{\eta \sum_{j=0}^{J^D-1} \pi_j P_C(m_j|j)}{\sum_{j=0}^{J^D-1} \pi_j \tau(m_j)}, \tag{9.1}$$

where $P_C(m_j|j)$ is the probability that all D destinations decode the packet correctly if combination $B_3(m_j)$ is used to transmit the packet and the multicast channel state is $\mathbf{v}_j$. The value of $P_C(m_j|j)$ can be easily determined through offline simulations of single links for a static AWGN channel.

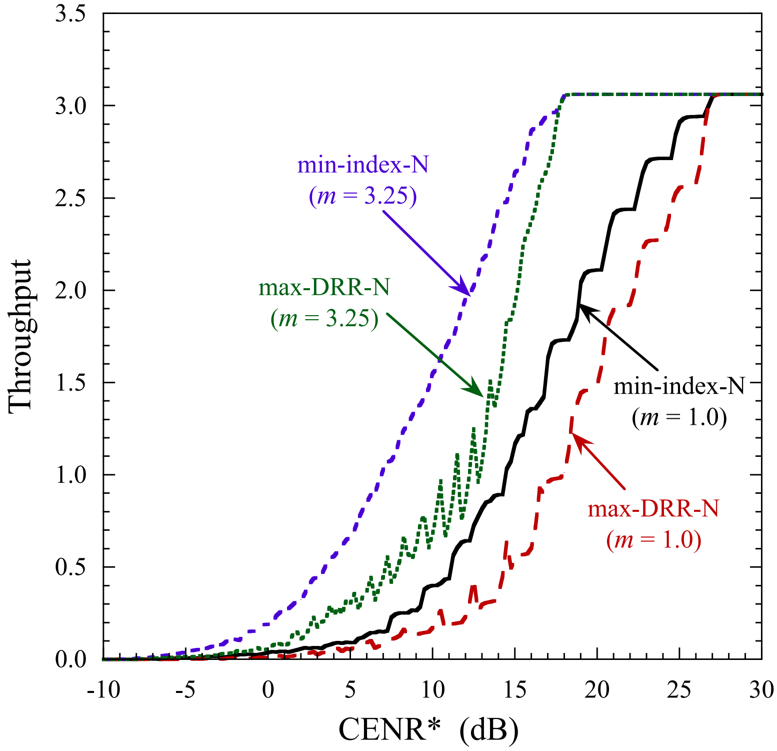


Figure 9.1: Multicast system with six standard links and no fountain coding.

For a reliable multicast system, if fountain coding is not employed, then all packets that fail to decode at *any* destination must be retransmitted. As discussed in Chapter 8, an adaptive protocol that attempts to maximize the single-transmission D-node throughput and chooses the minimum suggested index provides better performance in such a system than a protocol that attempts to maximize the single-transmission data recovery rate. This fact is illustrated in Fig. 9.1 for a multicast system with six standard links. Each of the links experiences Nakagami-m fading and has the same CENR*. Analytical results are shown for $m = 1$, which corresponds to Rayleigh fading, and for $m = 3.25$, which is moderate fading. In both instances, the min-index-N protocol outperforms the max-DRR-N protocol. It should be noted that the performance of the max-DRR-N protocol is non-monotonic. This is due to the fact that in many cases an increase in CENR* will result in the max-DRR-N protocol choosing a more aggressive code-modulation combination, but the use of that combination causes more packet failures, which degrades the performance in a multicast system with no fountain coding. Because the min-index protocol focuses on the most disadvantaged destination, it does not make such decisions.

9.2 Adaptive Fountain-Coded Multicast Transmission

For all of our numerical results, we employ the 3GPP raptor code for fountain coding [11]. Our simulations for our practical protocols include simulation of the Markov chain models for the Nakagami fading process as well as simulation of the iterative decoders required by the turbo product codes for each link of the multicast channel; however, to reduce the simulation time, we use a probabilistic model of the failure probability of the raptor code in order to simulate the raptor decoding process. Details of the probabilistic model are presented in Appendix B.

Recall that the 3GPP raptor code is a systematic code, which means that the first K packets transmitted by the source consist of the K information packets that make up the frame or a decodable set of these information packets, so that no excess packet transmissions are required to recover the frame of information packets. Let $N(d, t)$ be the number of channel packets that have been successfully received by destination d after t channel packets have been transmitted by the source. For the systematic 3GPP raptor code, destination d is able to decode the frame after the Kth channel packet has been transmitted if $N(d, K) = K$. Otherwise, if $N(d, K) < K$, then destination d is not able to decode the frame after the Kth packet transmission, and the source must continue to transmit fountain-coded packets until the destination is able to decode the frame. If $N(d, K) < K$, then our simulation follows the probabilistic model described in Appendix B to simulate the attempts by destination d to decode the frame. We let $P_f(d,t)$ denote the probability that destination d fails to decode the frame after t channel packets have been transmitted. This failure probability depends only on the number of excess packets received by the destination and not on the size of the frame. We use a counter, $\ell(d) = N(d,t) - K$, to represent the number of excess packets that have been received by destination d. In the simulation, this counter begins at zero when $N(d,t) = K$ and is incremented each time that destination d correctly decodes a channel packet. Each time destination d successfully receives a channel packet, the simulation draws a binary random variable $X_{\ell(d)}$ with a distribution that is given by $P(X_{\ell(d)} = 1) = 1 - P_f(d,t)$. Once a value of $\ell(d)$ is reached such that $X_{\ell(d)} = 1$, then destination has decoded the frame, and it reports its successful decoding of the frame to the source the next time it is designated as the reporting destination.

Our practical protocols rely on the round-robin reporting schedule to send the control information for the adaptive protocol to the source as well as send a notification of successful decoding of a frame. Our hypothetical protocols have no need for any control information from the destinations since the adaptive protocols are already given perfect channel state information for the links to *all* destinations; however, they do rely on the round-robin reporting schedule to notify the source when each destination has successfully decoded the frame.

The throughput results for our two adaptive protocols in a fountain-coded multicast system with six standard links is shown in Fig. 9.2. Each of the links experiences independent Nakagami-m fading with $m = 1$ and a normalized Doppler frequency of $f_d T_s = 0.02$. When comparing the results in this figure with those of Fig. 9.1, it is clear that the max-DRR protocol is a better choice for multicast systems that employ fountain coding. In Fig. 9.1 for $m = 1$ and no fountain coding, the min-index protocol outperformed the max-DRR protocol; however, when fountain coding is included the system in Fig. 9.2, the max-DRR protocol achieves much better performance

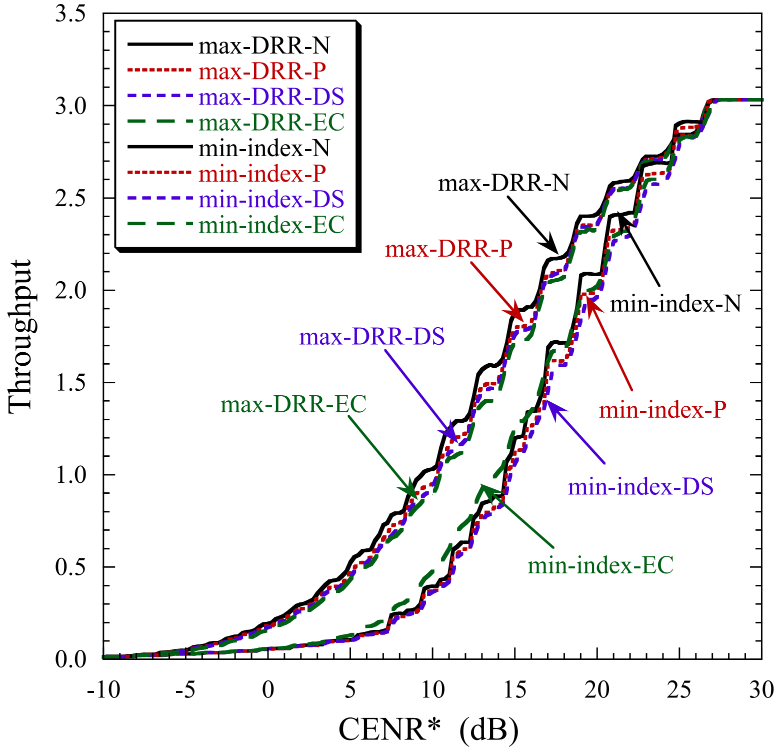


Figure 9.2: Fountain-coded multicast system with six standard links ($D = 6$, $m = 1$, $f_d T_s = 0.02$).

than the min-index protocol. This is not due to the min-index protocol performing worse with fountain coding. In fact, the performance of the min-index protocol is nearly unchanged when fountain coding is employed; however, the performance of the max-DRR protocol increases substantially when fountain coding is employed. For example, at CENR$^* = 15\,\text{dB}$, the hypothetical min-index-N protocol achieves a throughput of approximately 1.17 bits per chip when there is no fountain coding and a throughput of approximately 1.20 bits per chip when fountain coding is employed. In contrast, the max-DRR-N protocol achieves a throughput of approximately 0.55 bits per chip with no fountain coding and 1.89 bits per chip with fountain coding.

The performance for our practical protocols is shown in Fig. 9.2 as well as the performance for the hypothetical protocols given perfect channel state information. The practical protocol that uses the max-DRR selection criterion and demodulator statistics for its control information is the max-DRR-DS protocol. The max-DRR-DS protocol performs nearly as well as the hypothetical protocol given perfect channel state information for the previous packet transmission (max-DRR-P) and even has comparable performance to that of the protocol given perfect channel state information for the next packet transmission (max-DRR-N). Our practical protocol achieves this level of performance, even though it relies on round-robin reporting of imperfect information and the hypothetical protocols are given perfect information for all links for each packet transmission. The max-DRR-DS protocol has slightly better performance than the protocol that uses the max-DRR criterion and the error count for its control information (max-DRR-EC).

The min-index-DS protocol uses demodulator statistics for its control information, and it performs nearly as well as the hypothetical min-index-P and min-index-N protocols. The min-index protocol that relies on the error count (min-index-EC) actually has better overall performance than the min-index-P protocol and even outperforms the min-index-N protocol for some values of CENR*. This is due to the fact that the min-index criterion is not designed to provide good performance in fountain-coded multicast systems. The EC is less accurate than demodulator statistics, and it makes more mistakes when being used to suggest the combination index for the next packet transmission at each destination. Because the EC can only be determined if a packet is received correctly, the EC values are biased toward smaller values, and the resulting suggested indexes are biased toward combinations with higher indexes. This bias causes the min-index criterion to select code-modulation combinations with higher information rates, which actually improves the performance in fountain-coded multicast systems. As a result, the min-index-EC protocol provides slightly improved performance than even the min-index-N protocol in some cases, especially at low average SNR where the min-index criterion is more likely to select low-rate combinations.

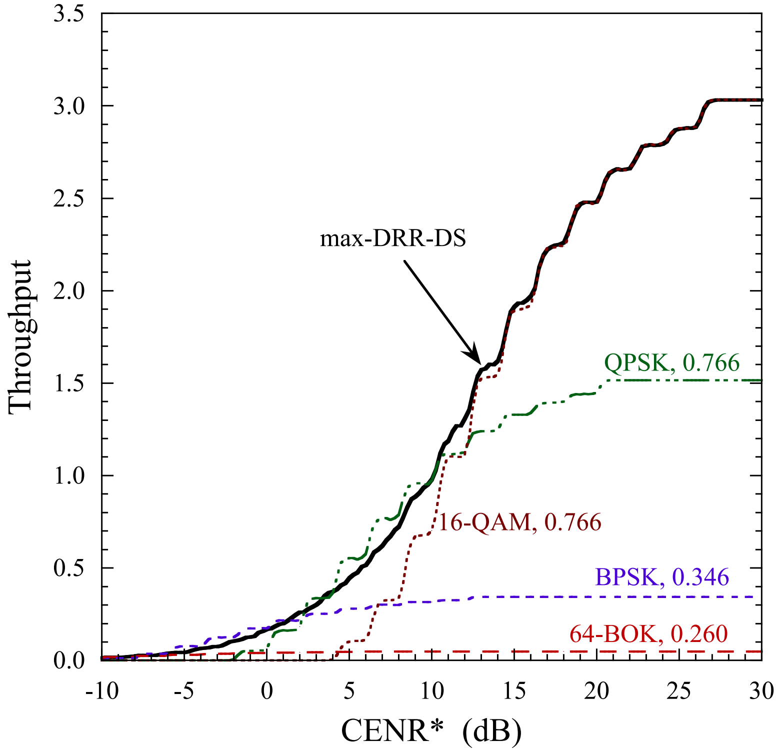


Figure 9.3: Fountain-coded multicast system with three standard links and three superior links ($D = 6$, $m = 1$, $f_d T_s = 0.02$, $\delta = 12\,\text{dB}$).

The performance results for a fountain-coded multicast system with three standard links and three superior links is shown in Fig. 9.3. The links of the multicast channel have independent Nakagami-m fading with $m = 1$ and a normalized Doppler frequency of $f_d T_s = 0.02$, and the superior links have a quality offset of 12 dB. In other words, the superior links have a value of CENR that is 12 dB larger than that of standard links in the absence of fading, so the channel conditions of the superior links are better than the standard links for the majority of the session (e.g., the destinations may be closer to the source). With the severe fading in this example, there is some overlap in the possible channel states for superior and standard links. For $m = 1$, the channel gain is $G_0 = -16\,\text{dB}$ in the worst channel state and the channel gain is $G_{11} = +6\,\text{dB}$ in the best state; therefore, a superior link in the worst channel state has a SNR of $\text{CENR}^* - 4\,\text{dB}$ and a standard link in the best channel state has a SNR of CENR* + 6 dB, which is 10 dB better than the superior link in the worst state.

The throughput results for the max-DRR-DS protocol are shown in Fig. 9.3, along with the throughput for a fixed-rate system with four of the code-modulation combinations from the set B_3. The max-DRR-DS protocol outperforms each of the fixed-rate combinations for the majority of the range of CENR* and nearly achieves the upper envelope of the four curves. For example, the max-DRR-DS protocol outperforms combination $B_3(13)$ (16-QAM with the 0.766 rate TPC) by at least 30% for all values of CENR$^* \leq 10.5$ dB, and combination $B_3(13)$ provides zero throughput for all values of CENR$^* < 4$ dB. The adaptive protocol outperforms combination $B_3(3)$ (64-BOK, 0.766) by a factor of two or more for all values of CENR$^* > 1.5$ dB. The adaptive protocol exceeds the performance of combination $B_3(5)$ (BPSK, 0.346) for all values of CENR$^* > 1.5$ dB, and it can exceed the performance of $B_3(5)$ by more than a factor of eight. The protocol outperforms combination $B_3(10)$ (QPSK, 0.766) for all values of CENR$^* \geq 7.5$ dB. Similar comparisons can be made for all possible fixed code-modulation combinations available to the adaptive protocol that are not shown in Fig. 9.3, and it is clear that fountain-coded multicast systems with time-varying channel conditions require adaptive transmission to achieve good performance.

The max-DRR criterion is best suited for multicast channels with high variability, where the fading on the links is such that no link suffers poor channel conditions for a significant period of time during the transmission of a frame. It works best for multicast channels with faster fading, where links are less likely to remain in a poor state for a long time. This is demonstrated by the performance results in Fig. 9.4 for a multicast system with six standard links. Each of the links experiences Rayleigh fading ($m = 1$). One curve shows the throughput for a system with a

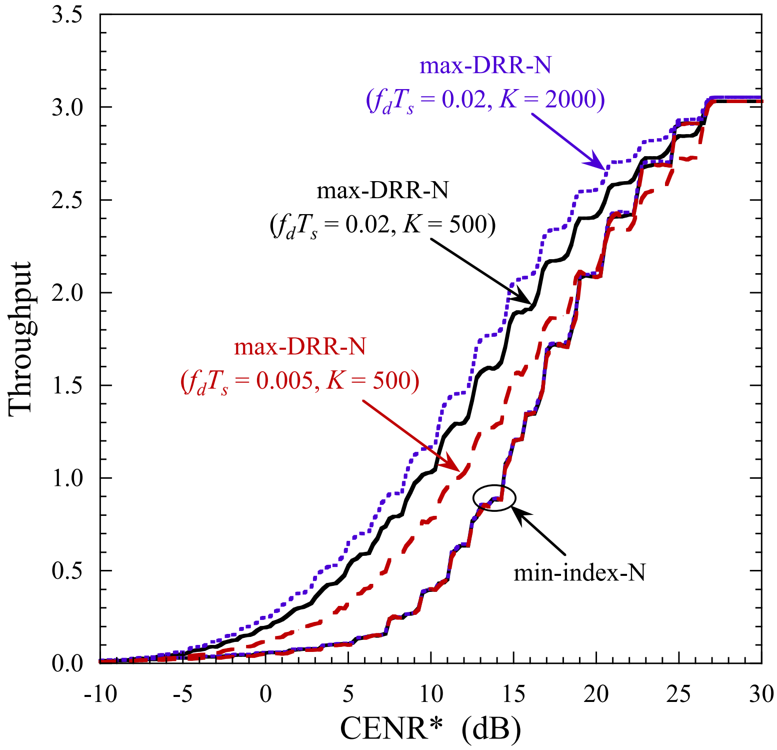


Figure 9.4: Fountain-coded multicast system with six standard links ($D = 6$, $m = 1$).

normalized Doppler frequency of $f_dT_s = 0.02$, which is relatively fast fading on each link, and a frame size of $K = 500$ information packets. Another curve shows the throughput results for a system with the same frame size but slower fading with $f_dT_s = 0.005$. The max-DRR-N protocol attains much better performance for the multicast system with faster fading. This is due to the fact that with slower fading, a given link is much more likely to stay in a deep fade for a long period of time. For our 12-state Markov chain model of Nakagami-m fading with $m = 1$, the state probability for the state with the deepest fade (state 0) is $\pi_0 \approx 0.0311$, and the channel gain for this state is $G_0 = -16\,\text{dB}$. These values are the same for any value of f_dT_s; however, the probability of staying in state 0 for the next packet given that a link was in state 0 for the previous packet is much greater for $f_dT_s = 0.005$ than for $f_dT_s = 0.02$. This probability is $q(0|0) \approx 0.727$ for $f_dT_s = 0.02$, but it is $q(0|0) \approx 0.931$

for $f_d T_s = 0.005$. In fact, the probability of staying in state 0 for the next four packets for $f_d T_s = 0.005$ is greater than the probability of staying in state 0 for just the next packet for $f_d T_s = 0.02$. The more dynamic channel conditions benefits the max-DRR protocol because a given destination is less likely to lag behind in collecting enough data packets to decode a frame due to a persistent deep fade.

A third max-DRR-N curve is included in Fig. 9.4 for $f_d T_s = 0.02$ and a frame size of $K = 2000$ information packets. This system has better performance than the system with a frame size of $K = 500$ and the same normalized Doppler frequency. The larger frame size allows for more variability in the link qualities for the duration of a frame. The increase in packet transmissions required to deliver a frame to all destinations allows more time for all of the links to experience good channel conditions for a significant amount of time. The extended duration of a frame allows the probability of a link being in state j over the course of a single frame to be closer to the state probability π_j, so all of the destinations are more likely to experience the same relative link qualities over the course of a frame. As explained above, the max-DRR protocol works best under these conditions.

All of the three scenarios shown for the max-DRR-N protocol in Fig. 9.4 are also shown for the min-index protocol; however, there is almost no difference between the three cases. The min-index protocol always focuses on the most disadvantaged destination, so its performance is not affected by the variability of the link qualities. Its performance is more dependent on the probability that any link is in a poor state. The probability that at least one of the six links in the multicast system of Fig. 9.4 is in state 0 is $1 - (1 - \pi_0)^6 \approx 0.173$, and this value is does not depend on the frame size or the normalized Doppler frequency.

The min-index criterion is designed to accommodate the needs of the most disadvantaged destination, so it can provide good performance in a multicast system

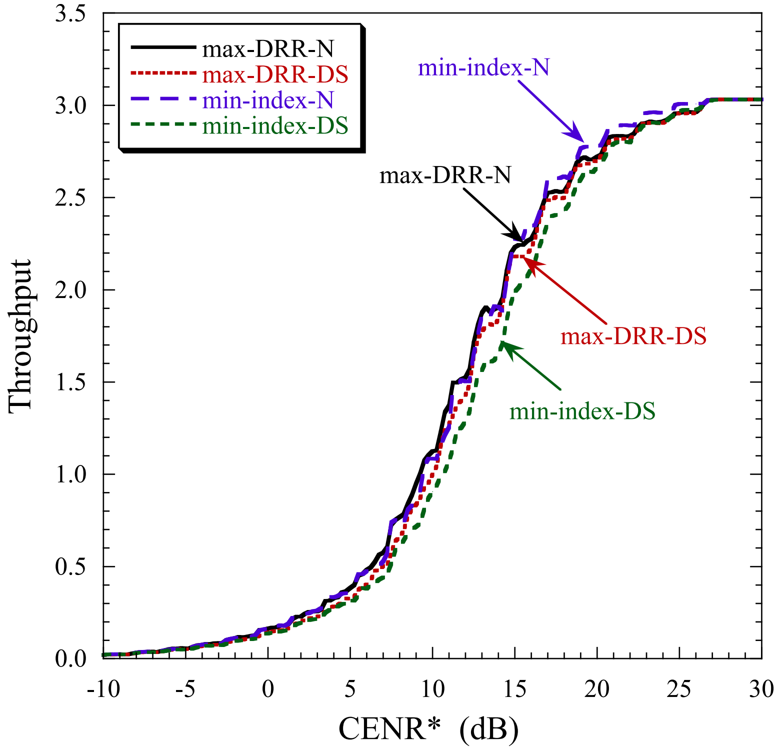
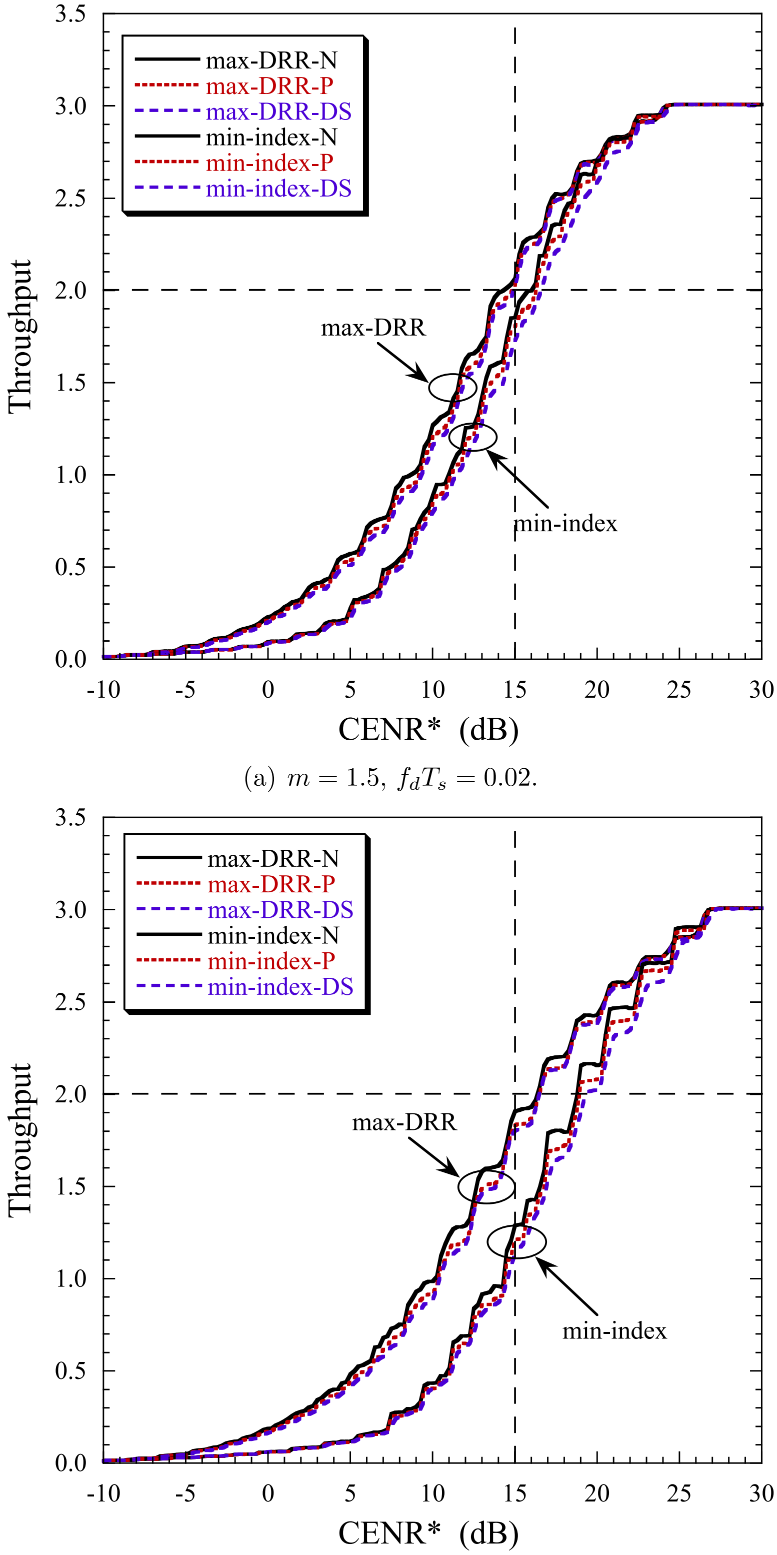


Figure 9.5: Fountain-coded multicast system with one standard link and five superior links ($D = 6$, $m = 1$, $f_d T_s = 0.02$, $\delta = 12\,\text{dB}$).

where one link has much poorer channel conditions than the other links in the multicast channel. The poor link is often referred to as the *disadvantaged node*, and it is often a concern that a single disadvantaged node will receive poor service during multicast transmission, even if adaptation is employed. In Fig. 9.5, we demonstrate that our protocols can achieve good performance in a system with a single disadvantaged node. The throughput results for a fountain-coded multicast system with one standard link and five superior links is shown in Fig. 9.5, where the SNR for the single disadvantaged node is 12 dB lower than the other five links. Each of the links experiences Nakagami-m fading with $m = 1$ and $f_d T_s = 0.02$. Results are shown for the practical protocols with demodulator statistics and the hypothetical protocols given perfect information about the state of the channel for the next packet transmission. The min-index-N protocol has the best overall performance, but the max-DRR-N protocol is very competitive and achieves nearly the same performance. The practical max-DRR-DS protocol actually outperforms the practical min-index-DS protocol, even though the single disadvantaged node is not conducive to the strategy of the max-DRR criterion. The fountain code allows the max-DRR protocol to target the destinations with better channel conditions without paying a severe penalty for packet erasures at the disadvantaged node. In fact, the difference between Fig. 9.2 and Fig. 9.5 is not due to the max-DRR protocol performing worse, but it is due to the min-index protocol performing much better. For example, with six standard links (Fig. 9.2) the max-DRR-DS protocol achieves a throughput of approximately 1.76 bits per chip at $\text{CENR}^* = 15\,\text{dB}$, and with one standard link (Fig. 9.5) the max-DRR-DS protocol achieves a throughput of approximately 2.18 bits per chip at $\text{CENR}^* = 15\,\text{dB}$. The max-DRR protocol actually performs slightly better in the system with a single disadvantaged node. The min-index-DS protocol by comparison achieves a throughput of approximately 1.063 bits per chip with six standard links at $\text{CENR}^* = 15\,\text{dB}$ and approximately 1.99 bits per chip with only one standard link at $\text{CENR}^* = 15\,\text{dB}$. The performance of the practical min-index protocol increases by 87% when going from six standard links to one. The max-DRR protocol only increases by 24%.

The severity of the fading on the communication links also plays a role in the performance of the adaptive protocols. The throughput for our adaptive protocols is shown in Fig. 9.6 for multicast systems with ten total links, five standard and five superior. The links for the system in Fig. 9.6(a) experience Nakagami-m fading with $m = 1.5$, and the links for the system in Fig. 9.6(b) experience more severe fading with $m = 1$. The normalized Doppler frequency is $f_d T_s = 0.02$ for both systems. The max-DRR protocol shows modest improvement with less severe fading, but the min-index protocol shows significant improvement with less severe fading. For example, at $\text{CENR}^* = 15\,\text{dB}$, the practical max-DRR-DS protocol achieves a throughput of ap-



(a) $m = 1.5,\ f_dT_s = 0.02$.

(b) $m = 1,\ f_dT_s = 0.02$.

Figure 9.6: Fountain-coded multicast system with five standard links and five superior links ($D = 10,\ \delta = 12\,\text{dB}$).

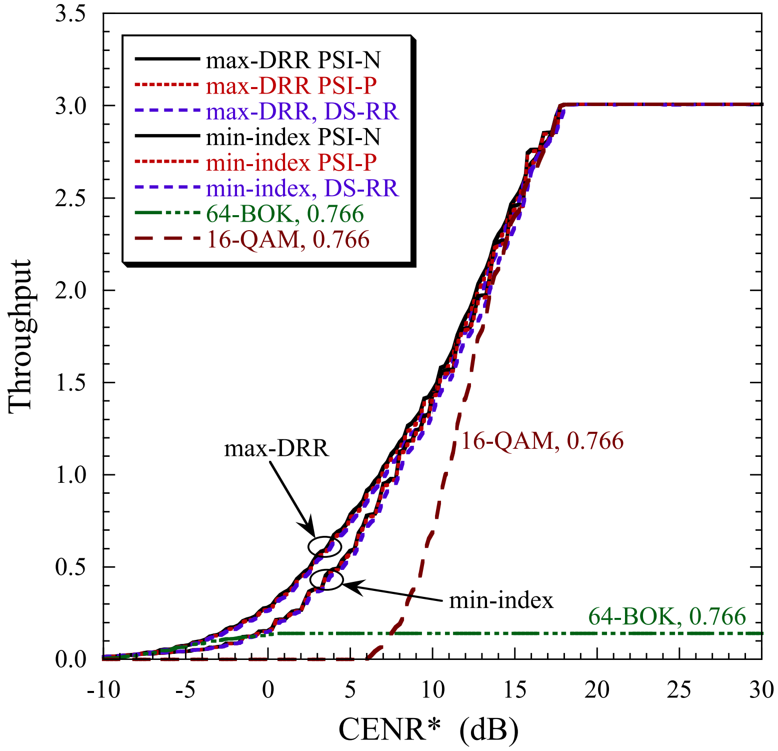


Figure 9.7: Fountain-coded multicast system with ten standard links ($D = 10$, $m = 3.25$, $f_d T_s = 0.02$).

proximately 1.8 bits per chip with Rayleigh fading (Fig. 9.6(b)) and approximately 2.03 bits per chip when $m = 1.5$ (Fig. 9.6(a)), which is a 12% increase in throughput with less severe fading. The min-index-DS protocol achieves a throughput of approximately 1.14 bits per chip with Rayleigh fading and approximately 1.73 bits per chip when $m = 1.5$, which is a 52% increase in throughput. With less severe fading, the min-index protocol shows substantial improvement; however, the max-DRR protocol is still the better option for the fountain-coded multicast system with $m = 1.5$. It provides a significant throughput advantage for all values of CENR* between -7 dB and 20 dB in Fig. 9.6(a). The protocols have similar performance outside this range.

The advantage of the max-DRR criterion becomes less significant as the severity of the fading decreases. This is demonstrated in Fig. 9.7 for a fountain-coded multicast system with ten standard links that experience relatively moderate Nakagami-m

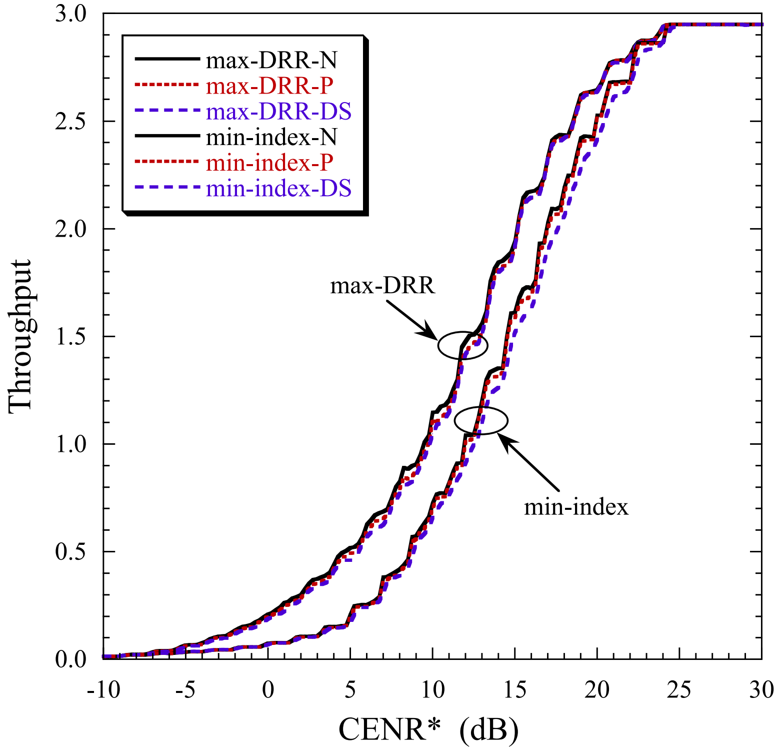


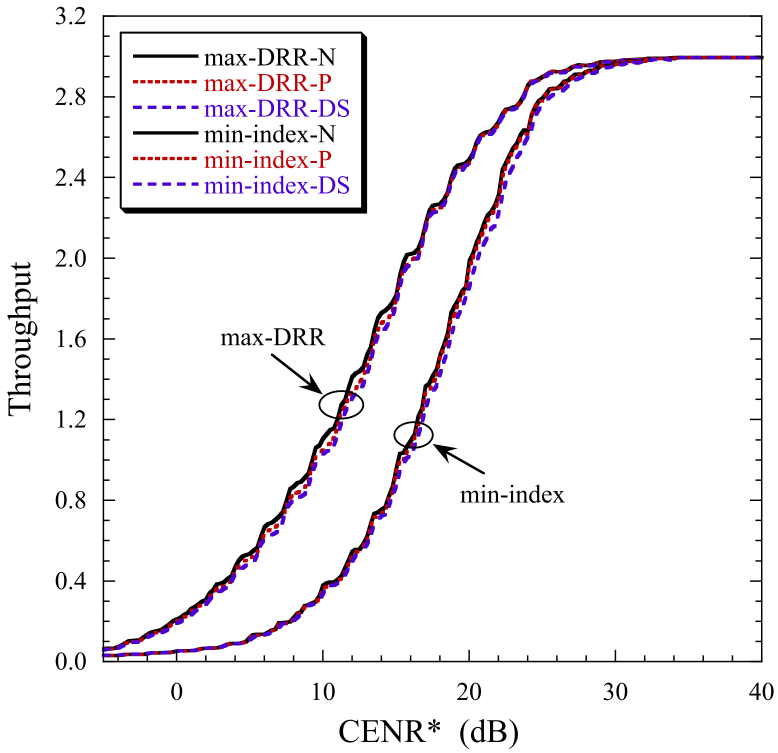
Figure 9.8: Fountain-coded multicast system with eight standard links and twelve superior links ($D = 20$, $m = 1.5$, $f_dT_s = 0.02$, $\delta = 12\,\text{dB}$).

fading with $m = 3.25$. The two protocols have similar performance for the majority of the range of CENR* considered in Fig. 9.7. The max-DRR still provides a throughput advantage for CENR* between $-7.5\,\text{dB}$ and $8\,\text{dB}$, but the performance is about the same outside of this range. Both protocols outperform the fixed-rate schemes that are also shown in Fig. 9.7. The adaptive protocols have an even larger advantage over fixed-rate schemes when the fading on the links is less severe. For example, combination $B_3(13)$ provides zero throughput for all valules of CENR$^* < 6\,\text{dB}$, and the max-DRR-DS protocol outperforms combination $B_3(3)$ by more than two-fold for all values of $CENR^* > 0\,\text{dB}$.

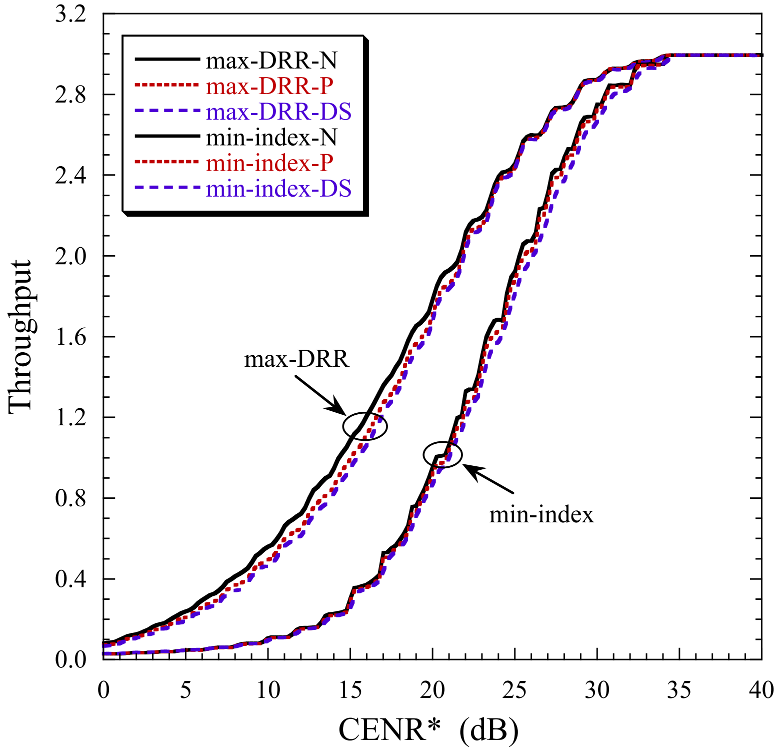
Our practical protocols perform nearly as well as their hypothetical counterparts for all of the performance results shown in this chapter, even though our protocols rely on round-robin reporting of imperfect information. Even if the size of the multicast group is increased to twenty destinations, our protocols are still able to perform nearly as well with round-robin reporting as the hypothetical protocols given full reporting of perfect channel state information. The multicast system in Fig. 9.8 is for a system with twenty destinations, with eight standard links and twelve superior links. Each of the links experiences Nakagami-m fading with $m = 1.5$ and a normalized Doppler of $f_d T_s = 0.02$. The results for the practical protocols are for systems that use demodulator statistics as the control information, but the error count may also be employed.

For all of our previous results, the quality offset parameter is used to model variations in the link qualities. The average channel conditions of superior links is better than the conditions of standard links for the entire duration of the session. In practice, the links qualities may vary over time. Phenomena such as shadowing or changes in terrain for mobile communications that may cause variations in link qualities will also vary over time, although slower than fading. For the performance results in Fig. 9.9, we replace the deterministic quality offset parameter with a Markov chain to model shadowing that is independent of the fading Markov chain. Each of the twelve links in the multicast system experiences Nakagami-m fading with $m = 1.5$ as well as shadowing. In Fig. 9.9(a), shadowing is modeled by a two-state Markov chain, where each link experiences a 10 dB shadow loss 4% of the time. In Fig. 9.9(b), shadowing is modeled by a three-state Markov chain, where all three states are equally likely, and the difference in SNR between states is 5 dB. Thus the maximum shadow loss is 10 dB.

The increased variation in the link qualities causes poor performance in the min-index protocol; however, the max-DRR protocol is designed for highly variable channel conditions. The packet erasure correction provided by the fountain coding allows the max-DRR criterion to select higher rate code-modulation combinations, even if



(a) Two-state Markov chain for shadowing ($\Delta = 10\,\text{dB}$).



(b) Three-state Markov chain for shadowing ($\Delta = 5\,\text{dB}$).

Figure 9.9: Fountain-coded multicast system with twelve links that experience fading and shadowing ($D = 12$, $m = 1.5$, $f_dT_s = 0.02$).

there is shadowing on some of the links. The min-index criterion elects to transmit at a low rate too often, and the protocol has low throughput as a result. Both adaptive protocols still perform nearly as well as the hypothetical protocols given perfect channel state information.

CHAPTER 10
CONCLUSION

Fountain coding is useful for its erasure correction capabilities and its potential to produce as much redundant information as needed to recover the original data. However, fountain coding alone does not guarantee good performance in wireless communication systems, and adaptive modulation and channel coding is necessary to compensate for fading, shadowing, and other time-varying propagation losses in order to achieve high throughput in such a system. Fixed-rate coding schemes are ineffective for these dynamic channel environments, but adaptation of the modulation and channel coding provides efficient communication in poor channel conditions and high throughput in good channel conditions.

We have introduced three low-complexity adaptive protocols for unicast transmission in systems that employ fountain coding for packet erasure correction. Two of our protocols, the AMPCF and AMCP protocols, are designed for systems with fixed-length channel packets, and the other, the AMCFL protocol, uses fixed-length data packets and variable-length channel packets. Of our three protocols, the AMCP protocol provides the highest throughput, but it is also the most complex of the three protocols, requiring the simultaneous transmission of multiple frames. The additional complexity may be worth the use of the AMCP protocol over the AMPCF protocol; however, if fixed-length channel packets are not important to the application, then the AMCFL protocol can provide performance that is competitive with that of the AMCP protocol with much lower complexity. The AMPCF protocol requires demodulator statistics for its operation, but the AMCP and AMCFL protocols have approximately the same performance using either the error count or demodulator statistics.

We have demonstrated that our adaptive protocols outperform fixed-rate coding schemes for fountain-coded unicast transmission and perform nearly as well as hypothetical protocols that are given perfect channel state information and employ ideal fountain codes. In most cases, the simulation results for a system with the raptor code and one of our practical protocols achieves performance that is within 5% of the analytical results for the benchmark protocol that has an ideal fountain code and is given perfect information about the state of the channel when the previous packet was received by the destination. Much of the 5% difference is due to the number of excess packet transmissions required by the raptor code. We have also demonstrated that permitting our adaptive protocols to send a pseudorandom sequence of bits at low signal-to-noise ratios can improve the session throughput performance, and our protocols show graceful degradation when operating with intermittent or delayed feedback from the destination.

We have also introduced two low-complexity adaptive protocols for multicast transmission. The max-DRR protocol is designed for systems that employ fountain coding to correct erased packets, and it attempts to maximize the rate of aggregate data recovered by the destinations in the multicast network. The min-index protocol is designed primarily for multicast systems that do not employ fountain coding, and it attempts to deliver each packet to all destinations. We showed that the max-DRR protocol provides much better performance for fountain-coded multicast transmissions if there is severe fading on the links of the multicast channel, and the two protocols have similar performance for multicast channels with more moderate fading. Even in unfavorable situations, the max-DRR protocol is able to provide high throughput in fountain-coded multicast systems. We also demonstrated that using only single-destination reporting of imperfect channel state information provided by the receiver statistics, our adaptive multicast transmission protocols perform nearly as well as hypothetical protocols given perfect channel state information for all links of the multicast channel for each packet transmission.

APPENDICES

APPENDIX A
ADAPTATION INTERVAL TESTS AND ENDPOINTS

For the numerical results for all of our adaptive protocols, each of the channel codes employed by the protocols is a turbo product code (TPC). The encoder and decoder hardware is available on a single chip for the TPCs employed by our protocols [17], and software encoders and decoders are also available. There are $L = 5$ TPCs available to each of our adaptive protocols. We let (C_i, n_i, k_i) denote that code C_i has block length n_i with k_i information bits per codeword. Two of our adaptive protocols for unicast transmission are designed for systems with fixed-length channel packets. The TPCs employed by the AMPCF and AMCP protocols for our numerical results are $(C_1, 2048, 484)$, $(C_2, 4096, 1331)$, $(C_3, 4096, 2028)$, $(C_4, 1024, 676)$, and $(C_5, 4096, 3249)$. The approximate rates for the channel codes are $r_1 \approx 0.236$, $r_2 \approx 0.325$, $r_3 \approx 0.495$, $r_4 \approx 0.660$, and $r_5 \approx 0.793$. Each of the fixed-length channel packets have length $N_i = 4096$ binary code symbols, so there are two codewords per channel packet for code C_1, four codewords per packet for C_4, and one codeword per packet for the other three channel codes. Our third adaptive protocol for unicast transmission is designed for systems with fixed-length data packets and variable-length channel packets. The channel codes employed by the AMCFL protocol for our numerical results are $(C_1, 4608, 1200)$, $(C_2, 6930, 2400)$, $(C_3, 5082, 2400)$, $(C_4, 3872, 2400)$, and $(C_5, 3135, 2400)$, and the approximate rates for the channel codes are $r_1 \approx 0.260$, $r_2 \approx 0.346$, $r_3 \approx 0.472$, $r_4 \approx 0.620$, and $r_5 \approx 0.766$. These channel codes are also used for the numerical results for our adaptive protocols for multicast transmission. Each of the variable-length channel packets have $\eta = 2400$ information bits, which corresponds to the number of data bits in each of the fixed-length data packets. There are two codewords per channel packet for code C_1, and one codeword per packet for the other four channel codes. The TPCs are convenient for evaluating the performance of our adaptive protocols and making comparisons between the practical protocols, our benchmarks, and fixed-rate schemes; however, other types of binary codes would work just as well with our protocols.

The set $\mathcal{B}_1$ of code-modulation combinations for the AMPCF protocol consists of all 25 possible combinations of the five TPCs with fixed-length channel packets and the five modulation formats. The AMCP and AMCFL protocols have the freedom to select any code-modulation combination for each packet transmission, and they are able to operate with a smaller set of combinations that lowers the complexity without affecting the performance of the protocol. If combination $B_n(h)$ uses a channel code of rate $\rho_n(h)$ and a modulation format that has an alphabet size of $M(h)$ with $L(h)$ modulation chips per modulation symbols, then the *information rate* of combination $B_n(h)$ is given by $R_n(h) = \rho_n(h) \log_2[M(h)/L(h)]$. The sets $\mathcal{B}_n = \{B_n(h) : 1 \leq h \leq 13\}$ of combinations employed by the AMCP ($n = 2$) and AMCFL ($n = 3$) protocols are indexed in order of increasing information rate and have the parameters shown in Table A.1. The five TPCs for fixed-length channel packets are used for set $\mathcal{B}_2$ for the AMCP protocol, and the five TPCs for the variable-length channel packets and fixed-length data packets are used for set $\mathcal{B}_3$ for the AMCFL protocol.

The endpoints for the decision intervals for the inter-frame adaptive channel coding stage of the AMPCF protocol are shown in Table A.2. If the modulation format for the last packet of a frame is $\mathcal{M}_j$ and the demodulator statistic falls in the interval $\mathcal{I}_j(\ell)$, then the channel code used for the next frame is C_ℓ. The ratio statistic used for 64-BOK ($\mathcal{M}_1$) and 16-BOK ($\mathcal{M}_2$) increases as channel conditions improve, and the decision intervals are of the form $\mathcal{I}_j(\ell) = [\gamma_{j,\ell-1}, \gamma_{j,\ell})$ for $j \in \{1, 2\}$, where $\mathcal{I}_j(1) = [0, \gamma_{j,1})$. The distance statistic used for BPSK ($\mathcal{M}_3$), QPSK ($\mathcal{M}_4$), and 16-QAM ($\mathcal{M}_5$) decreases as channel conditions improve, and the decision intervals are of the form $\mathcal{I}_j(\ell) = [\gamma_{j,\ell}, \gamma_{j,\ell-1})$ for $j \in \{3, 4, 5\}$, where $\mathcal{I}_j(1) = [\gamma_{j,1}, \infty)$. For simplification, only the interval endpoint $\gamma_{j,\ell}$ is shown for each entry in Table A.2.

The interval endpoints for the intra-frame adaptive modulation stage of the AM-PCF protocol are shown in Table A.3. Let C_ℓ be the channel code for the frame, and let $\mathcal{M}_j$ be the modulation for the previous packet. If the demodulator statistic falls in the interval $\mathcal{J}_{j,\ell}(k)$, then $\mathcal{M}_k$ is chosen as the modulation format for the next packet. The decision intervals for adaptive modulation are given by $\mathcal{J}_{j,\ell}(k) = [\alpha_{j,k,\ell-1}, \alpha_{j,k,\ell})$ for $j \in \{1, 2\}$, and the decision intervals are given by $\mathcal{J}_{j,\ell}(k) = [\alpha_{j,k,\ell}, \alpha_{j,k,\ell-1})$ for $j \in \{3, 4, 5\}$.

The interval endpoints for the AMCP-DS protocol are shown in Table A.4. If $\mathcal{M}_j$ is the modulation format for the previous packet and the demodulator statistic falls in the interval $\mathcal{I}_j(\ell)$, then the protocol selects $B_2(\ell)$ as the code-modulation combination for the next packet transmission. The demodulator statistics, and consequently the decision intervals, are dependent upon the modulation format for the previous packet as mentioned above. There are no combinations in the reduced set in Table A.1 that use 16-BOK, so there are no interval endpoints for $\mathcal{M}_2$ in Tables A.4–A.7.

The interval endpoints for the AMCP-EC are shown in Table A.5. The EC is determined at the output of the demodulator and depends on the modulation format used to transmit the channel packet. With fixed-length channel packets, the EC does not depend on the channel code. If $\mathcal{M}_j$ is the modulation for the previous packet and the EC falls in the interval $\mathcal{I}_j(\ell) = [\gamma_{j,\ell}, \gamma_{j,\ell-1})$, then the code-modulation combination for the next packet is $B_2(\ell)$. The EC has a larger variance than our demodulator statistics, and it cannot be used to span the entire set of code-modulation combinations. For example, if BPSK is the modulation for the previous packet, the lowest-rate combination that may be selected by the protocol for the next packet is combination $B_2(3)$, and the decision interval for $B_2(3)$ is $\mathcal{I}_3(3) = [900, 4096]$. The highest-rate combination that may be selected is $B_2(10)$, and the decision interval is $\mathcal{I}_3(10) = [0, 30)$. Recall that the EC can only be determined if the channel packet is successfully decoded. If the packet fails to decode, then the protocol selects the combination of next-lowest rate, if there is one available.

The interval endpoints for the AMCFL-DS are shown in Table A.6. Similar to the AMCP protocol, if $\mathcal{M}_j$ is the modulation for the previous packet, and the demodulator statistic falls in interval $\mathcal{I}_j(\ell)$, then the code-modulation combination for the next packet is $B_3(\ell)$. The interval endpoints for the AMCFL-EC protocol are shown in Table A.7. Because this protocol uses variable-length channel packets, the EC is dependent upon the modulation and channel code of the previous packet. Thus, if $B_3(j)$ is the code-modulation combination for the previous packet, and the EC is in the interval $\mathcal{I}_j(\ell)$, then the combination for the next packet is $B_3(\ell)$. The intervals are given by $\mathcal{I}_j(\ell) = [\gamma_{j,\ell}, \gamma_{j,\ell-1})$ for $j = 1, \ldots, 13$. If $B_3(j)$ was used for the previous packet, and ℓ is the index of the combination with the lowest information rate that may be chosen for the next packet, then the decision interval for $B_3(\ell)$ is $\mathcal{I}_j(\ell) = [\gamma_{j,\ell}, N_i]$, where i is the index of the channel code corresponding to combination $B_3(j)$.

Table A.1: $B_2(h)$ (AMCP) and $B_3(h)$ (AMCFL), $1 \leq h \leq 13$.

h	Modulation	$\rho_2(h)$	$R_2(h)$	$\rho_3(h)$	$R_3(h)$
1	64-BOK	0.236	0.044	0.260	0.049
2	64-BOK	0.495	0.093	0.472	0.089
3	64-BOK	0.793	0.149	0.766	0.144
4	BPSK	0.236	0.236	0.260	0.260
5	BPSK	0.325	0.325	0.346	0.346
6	QPSK	0.236	0.473	0.260	0.520
7	QPSK	0.325	0.650	0.346	0.693
8	QPSK	0.495	0.990	0.472	0.945
9	QPSK	0.660	1.320	0.620	1.240
10	QPSK	0.793	1.586	0.766	1.531
11	16-QAM	0.495	1.980	0.472	1.889
12	16-QAM	0.660	2.641	0.620	2.479
13	16-QAM	0.793	3.173	0.766	3.062

Table A.2: Interval endpoints for AMPCF channel code selection.

ℓ	$\gamma_{1,\ell}$	$\gamma_{2,\ell}$	$\gamma_{3,\ell}$	$\gamma_{4,\ell}$	$\gamma_{5,\ell}$
1	0.665	0.492	0.558	1.10	1.619
2	0.744	0.608	0.463	0.922	1.289
3	0.804	0.703	0.371	0.771	1.081
4	0.830	0.745	0.325	0.691	0.999
5	1.0	1.0	0	0	0

Table A.3: Interval endpoints for AMPCF adaptive modulation.

		C_1	C_2	C_3	C_4	C_5
j	k	$\alpha_{j,k,1}$	$\alpha_{j,k,2}$	$\alpha_{j,k,3}$	$\alpha_{j,k,4}$	$\alpha_{j,k,5}$
1	1	0.352	0.412	0.504	0.634	0.634
	2	0.458	0.519	0.634	0.712	0.752
	3	0.611	0.665	0.744	0.798	0.830
	4	0.779	0.815	0.865	0.899	0.915
	5	1.0	1.0	1.0	1.0	1.0
2	1	0.288	0.306	0.348	0.454	0.454
	2	0.325	0.356	0.454	0.558	0.620
	3	0.429	0.492	0.608	0.694	0.745
	4	0.664	0.721	0.799	0.850	0.874
	5	1.0	1.0	1.0	1.0	1.0
3	1	0.889	0.809	0.713	0.590	0.590
	2	0.758	0.699	0.590	0.504	0.452
	3	0.613	0.558	0.463	0.381	0.325
	4	0.411	0.352	0.259	0.195	0.164
	5	0	0	0	0	0
4	1	2.0	1.771	1.494	1.172	1.172
	2	1.623	1.455	1.172	0.995	0.904
	3	1.226	1.10	0.922	0.787	0.691
	4	0.836	0.739	0.567	0.431	0.364
	5	0	0	0	0	0
5	1	3.582	3.075	2.465	1.768	1.768
	2	2.749	2.379	1.768	1.417	1.261
	3	1.881	1.619	1.289	1.10	0.999
	4	1.162	1.046	0.90	0.794	0.724
	5	0	0	0	0	0

Table A.4: Interval endpoints for the AMCP-DS protocol.

ℓ	$\gamma_{1,\ell}$	$\gamma_{3,\ell}$	$\gamma_{4,\ell}$	$\gamma_{5,\ell}$
1	0.276	1.041	2.423	4.521
2	0.367	0.868	1.939	3.447
3	0.443	0.775	1.670	2.853
4	0.519	0.699	1.455	2.379
5	0.611	0.613	1.226	1.881
6	0.665	0.558	1.10	1.619
7	0.744	0.463	0.922	1.289
8	0.804	0.371	0.771	1.081
9	0.830	0.325	0.691	0.999
10	0.865	0.259	0.567	0.90
11	0.899	0.195	0.431	0.794
12	0.918	0.159	0.353	0.712
13	1.0	0	0	0

Table A.5: Interval endpoints for the AMCP-EC protocol.

ℓ	$\gamma_{1,\ell}$	$\gamma_{3,\ell}$	$\gamma_{4,\ell}$	$\gamma_{5,\ell}$
1	491			
2	189			
3	67	900		
4	15	762		
5	0	563	901	
6		425	763	
7		210	508	
8		70	276	
9		30	172	
10		0	61	511
11			0	299
12				173
13				0

Table A.6: Interval endpoints for the AMCFL-DS protocol.

ℓ	$\gamma_{1,\ell}$	$\gamma_{3,\ell}$	$\gamma_{4,\ell}$	$\gamma_{5,\ell}$
1	0.265	1.071	2.503	4.700
2	0.367	0.868	1.939	3.448
3	0.474	0.743	1.578	2.650
4	0.547	0.673	1.382	2.219
5	0.645	0.579	1.147	1.716
6	0.694	0.526	1.035	1.492
7	0.737	0.473	0.939	1.319
8	0.792	0.391	0.803	1.119
9	0.825	0.334	0.707	1.014
10	0.861	0.267	0.582	0.724
11	0.893	0.206	0.456	0.816
12	0.915	0.164	0.364	0.724
13	1.0	0	0	0

Table A.7: Interval endpoints for the AMCFL-EC protocol.

ℓ	$\gamma_{1,\ell}$	$\gamma_{2,\ell}$	$\gamma_{3,\ell}$
1	1221	673	
2	425	235	145
3	88	49	30
4	0	0	0

ℓ	$\gamma_{4,\ell}$	$\gamma_{5,\ell}$	$\gamma_{6,\ell}$	$\gamma_{7,\ell}$	$\gamma_{8,\ell}$
3	1904				
4	1588	1194			
5	1079	811	1844		
6	781	1588	1526	1148	
7	519	390	429	907	665
8	211	159	727	547	401
9	81	61	429	323	237
10	0	0	159	120	88
11			28	21	0
12			0	0	

ℓ	$\gamma_{9,\ell}$	$\gamma_{10,\ell}$	$\gamma_{11,\ell}$	$\gamma_{12,\ell}$	$\gamma_{13,\ell}$
8	305				
9	180	147			
10	67	55	663		
11	0	0	422	321	
12			236	179	146
13			0	0	0

APPENDIX B
RAPTOR CODE FAILURE PROBABILITY

For all of our numerical results, the fountain code that we employ for erasure correction is the 3GPP standard raptor code [11]. For our unicast performance results, the raptor encoding process and the raptor decoding process is included in the simulation. For our fountain-coded multicast performance results in Section 9.2, we use a probabilistic model for the raptor decoding process to reduce the simulation time required to apply the fountain decoder to multiple destinations for each frame. In this appendix we provide the details of our probabilistic model and the behavior of the decoding process of the 3GPP raptor code.

We let $N(d,t)$ denote the number of channel packets that were successfully decoded by the channel decoder at destination d after t packet transmissions. For the remainder of this appendix, we assume that $N(d,K) < K$, so that the systematic raptor code is not able to decode the frame after K packet transmissions, and $N(d,t) = K$ requires that $t > K$. In this case, the systematic 3GPP raptor code functions in the same manner as a nonsystematic raptor code. The source must continue to transmit fountain-coded packets until the destination has received enough packets to decode the frame.

Once the first channel packet is received such that $N(d,t) = K$, the destination applies the raptor decoder and attempts to decode the frame. If the decoding attempt fails, then the destination collects another data packet so that $N(d,t) = K + 1$ and attempts to decode the frame again. This process is repeated until the destination can successfully recover the frame of K information packets. The number of excess packets required to decode a frame is often referred to as the *overhead* of the fountain code. The overhead is dependent upon the fountain decoding algorithm, and for some fountain codes, it depends upon the frame size K, so that the number of total packets required to decode a frame is given by $(1+\varepsilon)K$ for some $\varepsilon > 0$ [4]. The 3GPP raptor code provides greater efficiency, in that the overhead of the raptor code depends only on the number of excess packets received by the destination, and not on the size of the frame. In other words, the total number of packets required to decode the 3GPP raptor code is of the form $K + \varepsilon$, rather than $(1 + \varepsilon)K$.

The probability that the raptor decoder fails to decode a frame after t packet transmissions is denoted by $P_f(d,t)$. For the 3GPP raptor code, this probability is only dependent upon the number of excess packets that are received by the destination. In [7], the authors present an approximation for the failure probability of the 3GPP raptor code. The approximation is given by

$$P_f(d,t) = \beta[N(d,t) - K], \tag{B.1}$$

where $\beta(\ell) = 0$ for $\ell < 0$ and

$$\beta(\ell) = 0.85(0.567)^{\ell}, \ \ \ell = 0, 1, 2, \ldots. \tag{B.2}$$

In [7], the authors claim that the approximation in (B.1) is valid for $K > 200$. We verified this approximation empirically, and found it to be reasonably accurate for $K > 200$ and less accurate for values of $K \leq 200$. The failure probability of the raptor decoder as determined by (B.1) is shown in Fig. B.1 as a function of the number of excess packets. The empirical values for the failure probability are also shown for four values of K. The approximation is relatively accurate when the number of excess packets is less than five, and it becomes less accurate as the number of excess packets increases. For $K = 300$ and $K = 500$, the failure probability determined by (B.2) is

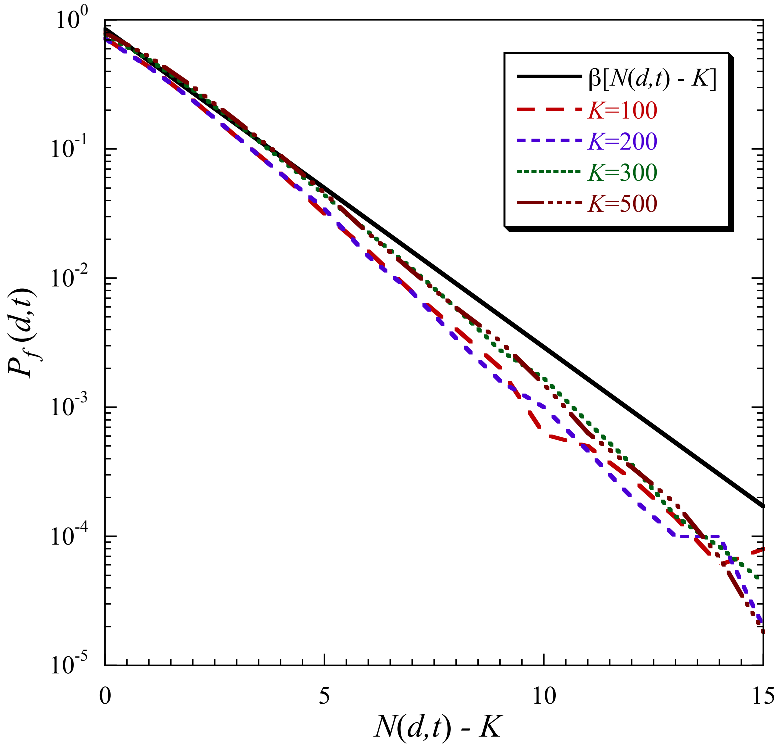


Figure B.1: Failure probability of 3GPP raptor decoder.

within 10% of the empirical values when the number of excess packets is less than five. For $K = 100$ and $K = 200$, the approximation is within 35% of the empirical values when the number of excess packets is less than five.

For our simulations, a counter, $\ell(d) = N(d,t) - K$, represents the number of excess packets that have been received by destination d. Once $N(d,t) = K$, the counter is set to zero and is incremented each time a channel packet is successfully decoded by destination d. Each time $\ell(d)$ is incremented, the simulation draws a binary random variable $X_{\ell(d)}$, whose distribution is given by $P(X_{\ell(d)} = 1) = 1 - P_f(d,t) = 1 - P_f[\ell(d)]$, where we have removed the dependence of P_f on t. Once a value of $\ell(d)$ is reached such that $X_{\ell(d)} = 1$, then the frame has been decoded by destination d, and the destination reports its successful decoding of the frame to the source the next time it is designated as the reporting destination. We obtained reliable empirical values for the failure probability $P_f[\ell(d)]$ for $\ell(d) \leq 15$. For our performance results presented

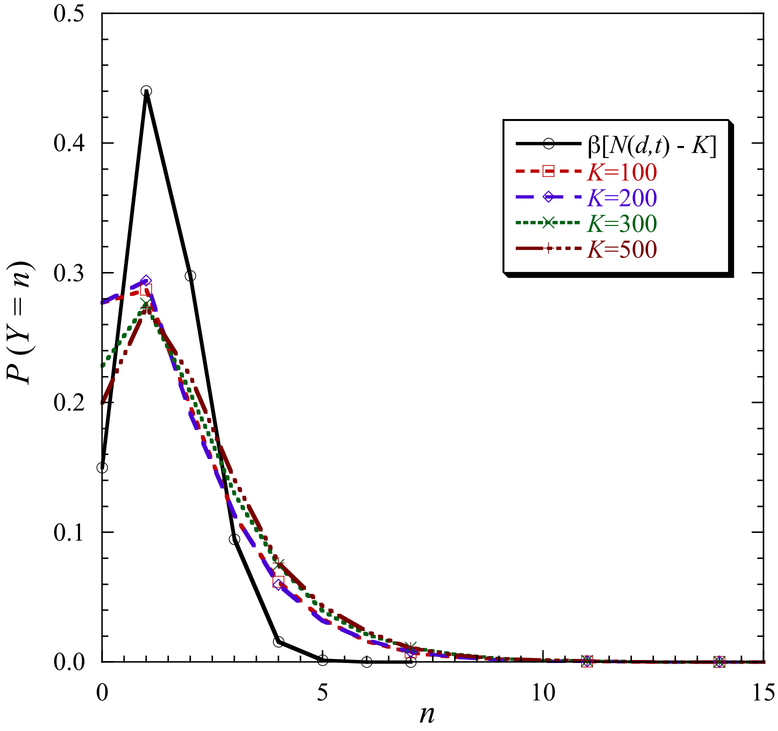


Figure B.2: Raptor decoder failure probability mass function.

in Chapter 9, if $\ell(d) \leq 15$, we use the empirical values of $P_f(\ell(d))$ for $K = 500$ when drawing the random variable $X_{\ell(d)}$. If $\ell(d)$ grows larger than fifteen before the frame has been decoded, then we use $P_f[\ell(d)] = \beta[\ell(d)]$ when drawing the random variable $X_{\ell(d)}$.

We let Y be a random variable that denotes the number of excess packets that have been received by the destination when the raptor decoder is first able to decode a frame. The distribution of the random variable Y can be determined from the failure probability of the raptor code. If the $P_f(i)$ is the probability that the raptor decoder fails to decode a frame after i excess packets have been successfully received by the destination, then the probability that the raptor decoder is able to recover the frame after i excess packets have been received is $1 - P_f(i)$. The probability mass function of Y is given by

$$P(Y = n) = (1 - P_f(n)) \prod_{i=0}^{n-1} P_f(i), \quad n = 0, 1, 2, \ldots . \tag{B.3}$$

Using (B.3), we can compute the probability that at least n excess packets must be received before the raptor decoder can recover a frame. Either the empirical values for P_f or the approximation can be used to compute the probability mass function. The discrete probability mass function, $P(Y = n)$, is shown in Fig. B.2, where the failure probability has been computed by the approximation in (B.2) or determined empirically for four values of K. The approximation gives a smaller average value for the required number of excess packets. We let $P(Y = n) = g(n)$, and the expected value of Y is given by

$$E\{Y\} = \sum_{n=0}^{\infty} n g(n). \tag{B.4}$$

Using (B.2) to compute P_f gives $E\{Y\} \approx 1.39$. Using the empirical value of P_f for $K = 500$ gives $E\{Y\} \approx 1.98$. Thus, on average, the raptor decoder is able to decode a frame after only two excess packets have been received by the destination.

REFERENCES

[1] M. Luby,"LT codes," *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–280, 2002.

[2] M. Mitzenmacher, "Digital fountains: A survey and look forward," *Proceedings of the IEEE Information Theory Workshop*, 2004, pp. 271–276, October 2004.

[3] D. J. C. MacKay, "Fountain codes," *IEE Proceedings—Communications*, vol. 152, no. 6, pp 1062–1068, December 2005.

[4] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[5] T. Courtade and R. D. Wesel, "Optimal allocation of redundancy between packet-level erasure coding and physical-layer channel coding in fading channels," *IEEE Transactions on Communications*, vol. 59, no. 8, pp. 2101–2109, August 2011.

[6] C. R. Berger, S. Zhou, Y. Wen, P. Willett, and K. Pattipati, "Optimizing joint erasure- and error-correction coding for wireless packet transmissions," *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4586–4595, November 2008.

[7] M. Luby, T. Gasiba, T. Stockhammer, and M. Watson, "Reliable multimedia download delivery in cellular broadcast networks," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 235–246, March 2007.

[8] Y. Cao and S. D. Blostein,"Cross-layer optimization of rateless coding over wireless fading channels," *Proceedings of the 25th Biennial Symposium on Communicaitons*, pp. 144–149, May 2010.

[9] S. W. Boyd, J. D. Ellis, and M. B. Pursley, "Adaptive coding and modulation for multicast transmission in packet radio networks," *International Journal of Wireless Information Networks*, vol. 20, no. 2, pp. 103–119.

[10] B. Rengarajan, G. H. Sim, and J. Widmer, "Adaptive modulation for finite horizon multicasting of erasure-coded data," *Proceedings of the Fifth International Conference on Communication Systems and Networks*, January 2013.

[11] *Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs*, 3GPP TS 26.346, v. 10.4.0, Release 10, ETSI, 2012.

[12] C. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

[13] M. K. Simon, S. M. Hinedi, and W. C. Lindsey, *Digital Communication Techniques*. Upper Saddle River, NJ: Prentice Hall, 1995.

[14] M. Nakagami, "The m-distribution–A general formula of intensity distribution of rapid fading," in *Statistical Methods in Radio Wave Propagation*, W. C. Hoffman (ed.), pp. 3–36, Pergamon Press, London, 1960.

[15] M. A. Juang and M. B. Pursley, "Finite-state Markov models for the intensity of Nakagami fading," *International Journal of Wireless Information Networks*, vol. 20, no. 2, pp. 95–102, May 2013.

[16] M. B. Pursley and T. C. Royster IV, "Low-complexity adaptive transmission for cognitive radios in dynamic spectrum access networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 83–94, January 2008.

[17] Advanced Hardware Architectures, Inc., Product Specification for AHA4501Astro 36 Mbits/sec Turbo Product Code Encoder/Decoder. Available: http://www.aha.com