NUI MAYNOOTH

Ollscoil na hÉireann Má Nuad

Coding in 802.11 WLANs

A dissertation submitted for the degree of

Doctor of Philosophy

by

Xiaomin Chen

Research Supervisor: Prof. Douglas Leith

Head of Department: Prof. Douglas Leith

Hamilton Institute

National University of Ireland Maynooth

Maynooth, Co. Kildare

Republic of Ireland

September 2012

Contents

List of Figures

List of Tables

Abstract

Forward error correction (FEC) coding is widely used in communication systems to correct transmission errors. In IEEE 802.11a/g transmitters, convolutional codes are used for FEC at the physical (PHY) layer. As is typical in wireless systems, only a limited choice of pre-specified coding rates is supported. These are implemented in hardware and thus difficult to change, and the coding rates are selected with point to point operation in mind.

This thesis is concerned with using FEC coding in 802.11 WLANs in more interesting ways that are better aligned with application requirements. For example, coding to support multicast traffic rather than simple point to point traffic; coding that is cognisant of the multiuser nature of the wireless channel; and coding which takes account of delay requirements as well as losses. We consider layering additional coding on top of the existing 802.11 PHY layer coding, and investigate the tradeoff between higher layer coding and PHY layer modulation and FEC coding as well as MAC layer scheduling.

Firstly we consider the joint multicast performance of higher-layer fountain coding concatenated with 802.11a/g OFDM PHY modulation/coding. A study on the optimal choice of PHY rates with and without fountain coding is carried out for standard 802.11 WLANs. We find that, in contrast to studies in cellular networks, in 802.11a/g WLANs the PHY rate that optimizes uncoded multicast performance is also close to optimal for fountain-coded multicast traffic. This indicates that in 802.11a/g WLANs cross-layer rate control for higher-layer fountain coding concatenated with physical layer modulation and FEC would bring few benefits.

Secondly, using experimental measurements taken in an outdoor environment, we model the channel provided by outdoor 802.11 links as a hybrid binary symmetric/packet erasure channel. This hybrid channel offers capacity increases of more than 100% compared to a conventional packet erasure channel (PEC) over a wide range of RSSIs. Based upon the established channel model, we further consider the potential performance gains of adopting a binary symmetric channel (BSC) paradigm for multi-destination aggregations in 802.11 WLANs. We consider two BSC-based higher-layer coding approaches, *i.e.* superposition coding and a simpler time-sharing coding, for multi-destination aggregated packets. The performance results for both unicast and multicast traffic, taking account of MAC layer overheads, demonstrate that increases in network throughput of more than 100% are possible over a wide range of channel conditions, and that the simpler time-sharing approach yields most of these gains and have minor loss of performance.

Finally, we consider the proportional fair allocation of high-layer coding rates and airtimes in 802.11 WLANs, taking link losses and delay constraints into account. We find that a layered approach of separating MAC scheduling and higher-layer coding rate selection is optimal. The proportional fair coding rate and airtime allocation (i) assigns equal total airtime (*i.e.* airtime including both successful and failed transmissions) to every station in a WLAN, (ii) the station airtimes sum to unity (ensuring operation at the rate region boundary), and (iii) the optimal coding rate is selected to maximise goodput (treating packets decoded after the delay deadline as losses).

Acknowledgements

I would like to express my deepest and sincere gratitude to my supervisor, Professor Douglas Leith. This thesis would not have been possible without his inspiring guidance, warm encouragement and continuous support. His wide knowledge, sharp judgement, enthusiasm and dedication to top-quality research have been of great value for me, and will continue to benefit me through my research journey.

I should also send my sincere thanks to Dr. Vijay Subramanian at Northwestern University, USA and Dr. Premkumar Karumbu at SSN College of Engineering, India for their patient instruction and kind help throughout my PhD study.

I am also grateful to Dr. David Malone, Dr. Ken Duffy, Dr. Tianji Li and everyone else in the Hamilton Network Group for their valuable suggestions and inspiring discussions.

A number of people have made my stay in Ireland an enjoyable experience, especially Rosemary Hunt and Kate Moriarty. I am so grateful for their willingness to help in all aspects of work and life.

Finally, I am forever indebted to my parents for their understanding, support and endless love, and also to my husband, Jiang, for always being supportive of me and helping me keep positive when I feel frustrated.

Chapter 1

Introduction

Forward error correction (FEC) coding [15, 41] is widely used in communication systems to correct transmission errors. The idea is that the sender protects the information message by adding redundancy. The redundancy allows the receiver to detect, and often correct, a limited number of errors.

In IEEE 802.11a/g transmitters, convolutional codes [41] are used for FEC at the physical layer. As is typical in wireless systems, only a limited choice of pre-specified coding rates is supported. These are implemented in hardware and thus difficult to change, and the coding rates are selected with point to point operation in mind.

However, in a <u>network</u> the potential exists to use coding in more interesting ways that are better aligned with application requirements. For example, coding to support multicast traffic rather than simple point to point traffic; coding that is cognisant of the multiuser nature of the wireless channel; and coding which takes account of delay requirements as well as losses. This is the focus of the present thesis. Rather than adopting a clean slate approach, we consider 802.11 WLANs, which are now ubiquitous, and investigate layering additional coding on top of the existing 802.11 physical layer coding. Of course this causes a performance cost compared to a clean slate design, but it has the compelling advantage of potentially being useful to the large number of users of the existing 802.11 devices.

1.1 IEEE 802.11 WLAN

A **Wireless Local Area Network** (WLAN) is a computer network that uses a wireless communication method to connect computers and devices in a limited geographical area, such as home, school, computer laboratory or office building. It typically extends an existing wired Local Area Network (LAN) by attaching a device, called the access point (AP), to the edge of the wired network. Users communicate with the AP using a wireless network adapter similar in function to a traditional Ethernet adapter. This gives users the mobility to move around within a local coverage area and still be connected to the network. Therefore, WLAN often provides the last mile wireless access to the wired network [19].

The most common and successful standard for WLAN thus far has been the 802.11 standard by IEEE.

802.11 is a member of IEEE 802 family. IEEE 802 is a series of specifications for LAN technologies. It focuses on the two lowest layers of the OSI model: the Media Access Control (MAC) layer and the Physical (PHY) layer. The MAC specifies rules on how to access the medium and send data. The details of transmission and reception are specified by the PHY.

The original version of the IEEE 802.11 standard was released in 1997 [25]. It specified two PHY data rates of 1Mbps and 2Mbps, and three PHY layer technologies *i.e.* Diffuse Infrared (DIR), Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS). The FHSS and DSSS PHY layers operate over the Industrial Scientific Medical (ISM) frequency band at 2.4GHz.

In 1999 IEEE released its second extension, 802.11b, to the basic 802.11 specification [27]. The 802.11b PHY layer is an extension of the DSSS PHY. It uses Complementary Code Keying (CCK) as its modulation technique and increases the maximum raw PHY rate to 11Mbps. 802.11b uses the same MAC as defined in the original standard and also operates in the 2.4GHz band.

The other amendment to the original standard released in 1999 is IEEE 802.11a [26]. 802.11a operates in the 5GHz band and uses a 52-subcarrier orthogonal frequency-division multiplexing (OFDM) technology at the PHY layer. The maximum raw PHY rate is increased up to 54Mbps. Since the 2.4GHz band is heavily used to the point of being crowded, using the relatively unused 5GHz band potentially offers 802.11a the significant advantage of less interference.

1n 2003, the fourth amendment 802.11g was ratified [28]. It operates in the 2.4GHz band as 802.11b,

but uses the same OFDM PHY layer as defined in 802.11a. The maximum PHY data rate remains 54Mbps.

802.11e [29] is an amendment that defines a set of Quality of Service (QoS) enhancements via modifications to the MAC layer. It is considered of importance for delay-sensitive applications. The amendment was incorporated into the published IEEE 802.11-2007 standard [30].

The latest IEEE 802.11 amendment 802.11n was released in October 2009 [31]. It supports multiple-input multiple-output (MIMO) antennas and increases the maximum raw PHY rate from 54Mbps to 600Mbps. 802.11n uses four spatial streams at a channel width of 40MHz and operates on both the 2.4GHz and the lesser used 5GHz bands.

1.1.1 The 802.11 MAC

The 802.11 MAC defines two different access mechanisms, the mandatory Distributed Coordination Function (DCF) which provides distributed channel access based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), and the optional Point Coordination Function (PCF) which provides centrally controlled channel access through polling.

Distributed Coordination Function

The DCF is the basic access mechanism of IEEE 802.11. In the DCF, all stations contend for access to the medium in a distributed manner, based on the CSMA/CA access mechanism. DCF is hence referred to as contention-based channel access.

CSMA works in a listen-before-talk fashion. Before transmitting, a station first listens (by carrier sensing) whether the radio link is clear. If the medium is sensed idle for at least a period of DCF Inter-Frame Space (DIFS), the station starts transmitting; meanwhile, all other stations which intend to transmit during this period have to wait until the medium becomes idle again for a DIFS period. If the destination station successfully receives a frame, it acknowledges by sending back an ACK frame after a period of Short Inter-Frame Space (SIFS). Fig. 1.1 illustrates the procedure.

802.11 defines three basic interframe spaces which are used to determine medium access. The relationship between them is shown in Fig. 1.2. The durations of these three interframe spaces are defined in
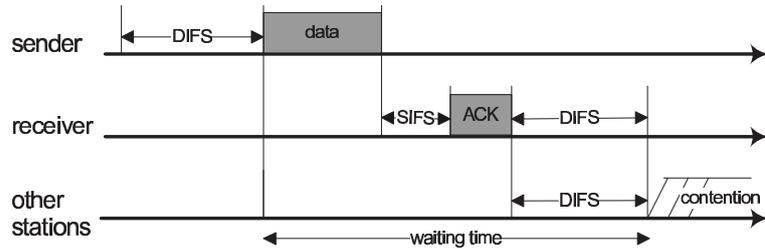
Figure 1.1: 802.11 DCF basic access mechanism.
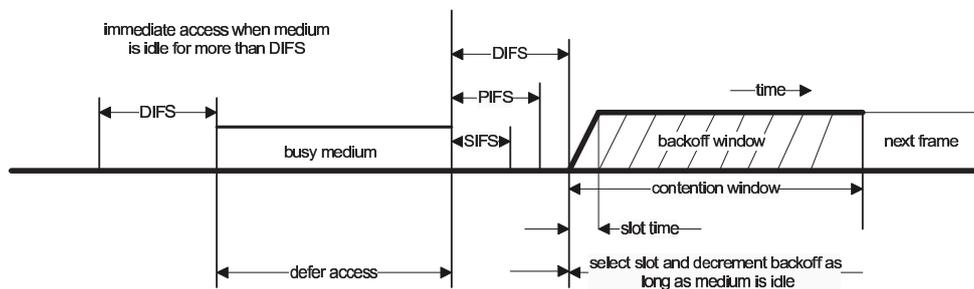
Figure 1.2: 802.11 interframe spacing relationships

relation to the PHY slot time, which is a fixed duration and independent of the PHY data rate. Varying the interframe spacing creates different priority levels for different types of transmissions. SIFS is the shortest interframe space for the highest-priority transmissions, such as ACK frames. The SIFS between the data and the ACK frame prevents other stations from transmissions during the period when the receiver transmits the ACK because other stations have to wait for the longer DIFS. The second shortest interframe space PIFS (PCF Inter-Frame Space) is used by the AP in the PCF. Since in PCF the AP controls the access to the medium by polling individual ordinary stations, the AP is given priority over ordinary stations by waiting PIFS instead of the longer DIFS prior to transmitting a frame.

If two or more stations sense the medium idle and try to transmit at the same time, a collision will

4

happen. To mitigate such collisions, stations have to wait for an additional time prior to transmitting if the medium is sensed busy during the DIFS period, or, if the medium was busy before a station started waiting for the DIFS period. In these situations, the station defers access by choosing a random backoff value, which specifies the time period (measured in PHY time slots) that the station has to wait in addition to the DIFS after the medium becomes idle. This additional random delay in form of backoff helps to mitigate collisions, otherwise all stations would try to transmit as soon as medium becomes idle for a DIFS period. This mechanism is called Collision Avoidance (CA) and the whole access mechanism is thus referred to as CSMA/CA.

After choosing the backoff value, once the medium is sensed idle for at least a DIFS, the station starts decrementing its backoff value by one for each PHY time slot. If the medium becomes busy during this backoff process, the station pauses its backoff timer. The backoff timer is resumed as soon as the medium is sensed idle for a DIFS period again. The station is allowed to transmit once the backoff timer reaches zero. The random backoff value is uniformly chosen from the interval $[0, CW]$, called the Contention Window. At the first transmission attempt, CW is set to be the minimum Contention Window size CW_{min}. An unsuccessful transmission is determined if the transmitter does not receive an ACK frame after a specified ACK timeout period. After each unsuccessful transmission, CW is doubled, *i.e.* increased exponentially until it reaches the maximum Contention Window size CW_{max}. With the doubled CW size, the probability of choosing a larger random backoff value is higher, and hence the probability of the stations colliding again is reduced. The CW size is reset to CW_{min} after each successful transmission. The values of CW_{min} and CW_{max} are dependent on the underlying PHY layer.

DCF specifies a retransmit limit (also referred to as the retry limit), *i.e.* the number of times a frame is allowed to be retransmitted. If the transmitter has not received an ACK for a frame after reaching the retransmit limit, the frame is dropped.

The backoff mechanism is also used after a successful transmission before sending the next frame, that is, if the transmitter has another frame to send just after receiving an ACK for the previous one, it waits the medium to be idle for a DIFS and chooses a new backoff value. This is referred to as post backoff, as it is done after the transmission rather than before. The post backoff ensures that the medium is not occupied by only one station for a long time. It allows other stations to decrement their backoff timers as well and thereby to have a chance to get access to the medium.

The Collision Avoidance mechanism does not completely eliminate the risk of collisions. Collisions
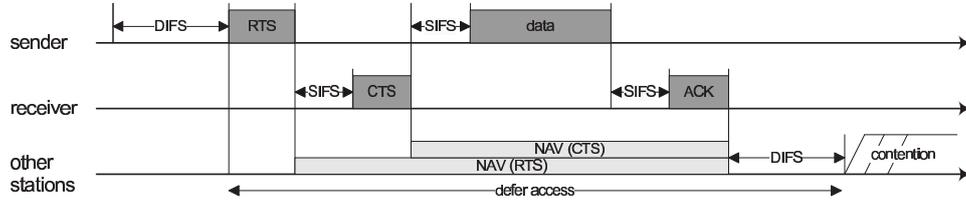
Figure 1.3: 802.11 RTS/CTS frame exchange mechanism

may still occur if the backoff timers for two or more stations reach zero at the same time. Although a larger contention window size reduces the probability of collisions, it results in higher delays and less efficient bandwidth utilization.

An additional mechanism, RTS/CTS, is defined in 802.11 to mitigate hidden node problems. With RTS/CTS, the transmitter and receiver perform a handshake by exchanging Request to Send (RTS) and Clear To Send (CTS) control frames. The procedure is shown in Fig. 1.3. After waiting a DIFS, prior to transmitting the data frame, the transmitter sends a RTS frame to the receiver. The receiver responds with a CTS frame after waiting for a SIFS. The CTS frame indicates that the handshake is successful and ensures that the medium has been reserved for the particular pair of transmitter and receiver.

1.1.2   The 802.11 Physical Layer

IEEE 802.11 standard specifies five physical layers in the 2.4GHz ISM band and the 5GHz band. Table 1.1 lists the characteristics of the five PHY layers.

OFDM PHY in 802.11a/g

We will focus on the 802.11a/g OFDM PHY in this thesis. Table 1.2 summarizes the encoding details for each of the PHY rates available in IEEE 802.11a/g OFDM PHY.

OFDM encodes digital data onto multiple orthogonal carrier frequencies, *i.e.* a binary data stream is

PHY layer	Maximum PHY rate (Mbps)	Frequency band (GHz)	Standard version
FH PHY	2	2.4	original 802.11
DS PHY	2	2.4	original 802.11
HD/DSSS PHY	11	2.4	802.11b
OFDM PHY	54	2.4/5	802.11a/g
MIMO PHY	600	2.4/5	802.11n

Table 1.1: IEEE 802.11 PHY layers

Data rate (Mbps)	Modulation	Convolutionol coding rate	Coded bits per subcarrier N_{BPSC}	Coded bits per OFDM symbol N_{CBPS}	Data bits per OFDM symbol N_{DBPS}
6	BPSK	1/2	1	48	24
9	BPSK	3/4	1	48	36
12	QPSK	1/2	2	96	48
18	QPSK	3/4	2	96	72
24	16QAM	1/2	4	192	96
36	16QAM	3/4	4	192	144
48	64QAM	2/3	6	288	192
54	64QAM	3/4	6	288	216

Table 1.2: IEEE 802.11a/g OFDM PHY encoding parameters
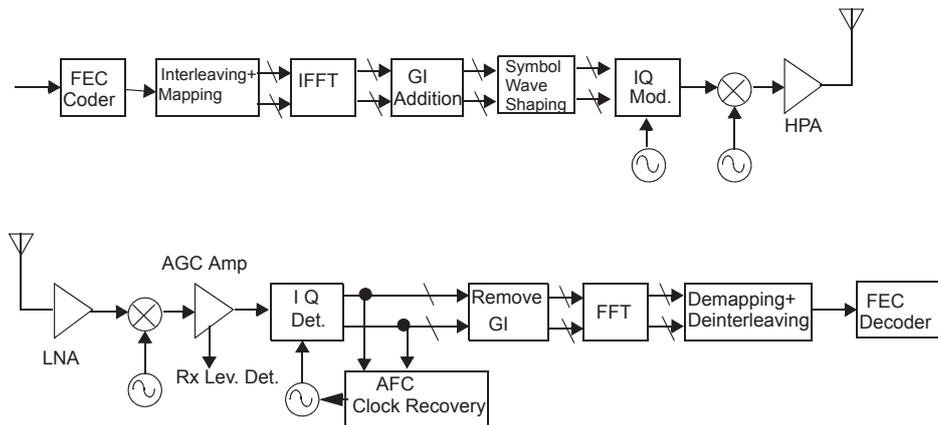
Figure 1.4: Transmitter and receiver block diagram for 802.11 OFDM PHY

partitioned into lower-speed substreams, with each substream then being modulated onto an orthogonal subcarrier [5]. A block diagram of the transmitter and receiver in the 802.11a/g OFDM PHY is shown in Fig. 1.4.

At the transmitter the input bit stream is first encoded using a FEC coder. In the 802.11a/g OFDM PHY convolutional coding is used, as illustrated in Fig. 1.5. The encoder uses a 6-stage shift register. The input bit stream is shifted along the register 1 bit at a time. The output is alternately generated according to two generator polynomials $g_0 = 1011011$ and $g_1 = 1111001$ with coding rate 1/2. Higher coding rates are derived by employing "puncturing" *i.e.* some of the encoded bits are omitted. At the receiver side, a dummy "zero" is inserted into the convolutional decoder in place of the omitted bits. The higher coding rate 2/3 and 3/4 in the 802.11a/g OFDM PHY are obtained by puncturing the low-rate 1/2 code using the specific patterns indicated in Fig. 1.6.

The encoded bit stream is then interleaved by a block interleaver with a block size equal to the number of coded bits in a single OFDM symbol N_{CBPS} (see Table. 1.2). The interleaver is defined by a two-step permutation. The first permutation ensures that adjacent coded bits are mapped onto nonadjacent subcarriers. The second ensures that adjacent coded bits are mapped alternately onto less and more significant bits of the constellation and, thereby, long runs of low reliability bits are avoided.

After interleaving the encoded and interleaved bit stream is partitioned into groups of N_{BPSC} bits and converted into complex numbers representing constellation points. The conversion is performed

8
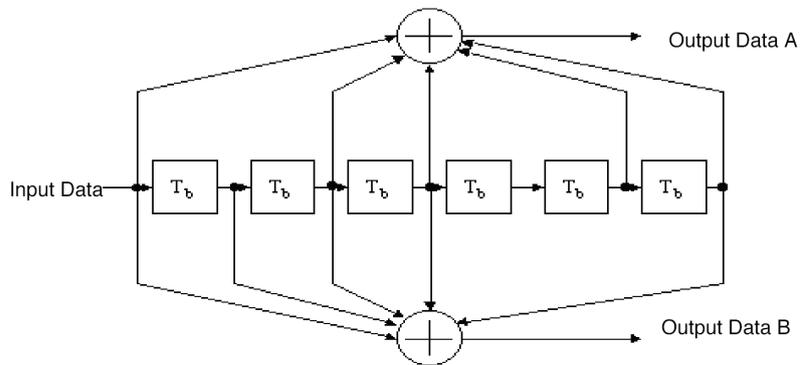
Figure 1.5: 802.11a/g OFDM PHY convolutional encoder, coding rate 1/2 [30]
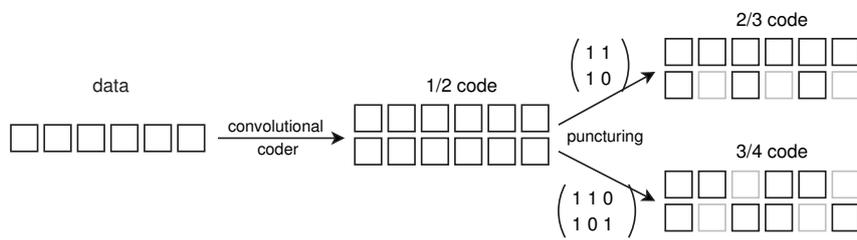
Figure 1.6: Rate 2/3 and 3/4 convolutional codes are obtained by puncturing a low-rate 1/2 code

according to Gray-coded constellation mappings, as illustrated in Fig. 1.7, with the input bit b_0 being the earliest in the bit stream.

The 802.11 OFDM PHY uses 52 subcarriers, of which 4 are dedicated to pilot signals in order to assist the receiver compensate against frequency offsets and phase noise. These pilot signals are modulated onto subcarriers -21, -7, 7 and 21. The stream of data complex numbers is partitioned into groups of 48 complex numbers, with each group modulated onto one of the remaining 48 OFDM subcarriers. OFDM symbols are transformed from frequency domain to time domain by applying the Inverse Fast Fourier Transform (IFFT). Guard Interval (GI) is inserted between symbols to make the system robust to multipath propagation. Windowing is applied after to bring the signal of a new symbol gradually up to full strength while allowing the old symbol to fade away. Using an IQ modulator, the signal is converted to analog, then upconverted on the 5GHz band, amplified, and transmitted through the antenna. The receiver performs the reverse operations of the transmitter in the reverse order.

1.2   Contributions

This thesis is concerned with the application of coding in 802.11 WLANs. We consider the joint performance of higher-layer coding concatenated with 802.11 PHY layer modulation/coding in two channel paradigms *i.e.* packet erasure channel (PEC) and binary symmetric channel (BSC). In the BSC paradigm, we further derive the proportional fair allocation of higher-layer coding rates and airtimes in an 802.11 WLAN.

In Chapter 2 we consider the joint multicast performance of higher-layer fountain coding concatenated with 802.11a/g OFDM PHY modulation/coding. We are interested in the cross-layer trade-offs between fountain coding and PHY layer modulation and coding rate selection. A detailed study on the optimal choice of PHY modulation/coding rates with and without higher layer fountain coding is carried out for standard 802.11 WLANs. Optimality is considered both in terms of maximising goodput and minimising energy. We find that, in contrast to studies in cellular networks, in 802.11a/g WLANs the PHY rate that optimizes uncoded multicast performance is also close to optimal for fountain-coded multicast traffic. This indicates that in 802.11a/g WLANs cross-layer rate control for higher-layer fountain coding concatenated with physical layer modulation and FEC would bring few benefits and PHY layer rate control can be carried out without regard to the use of fountain coding at higher layers. Fountain codes are a class of block codes for erasure channels. The analysis of fountain
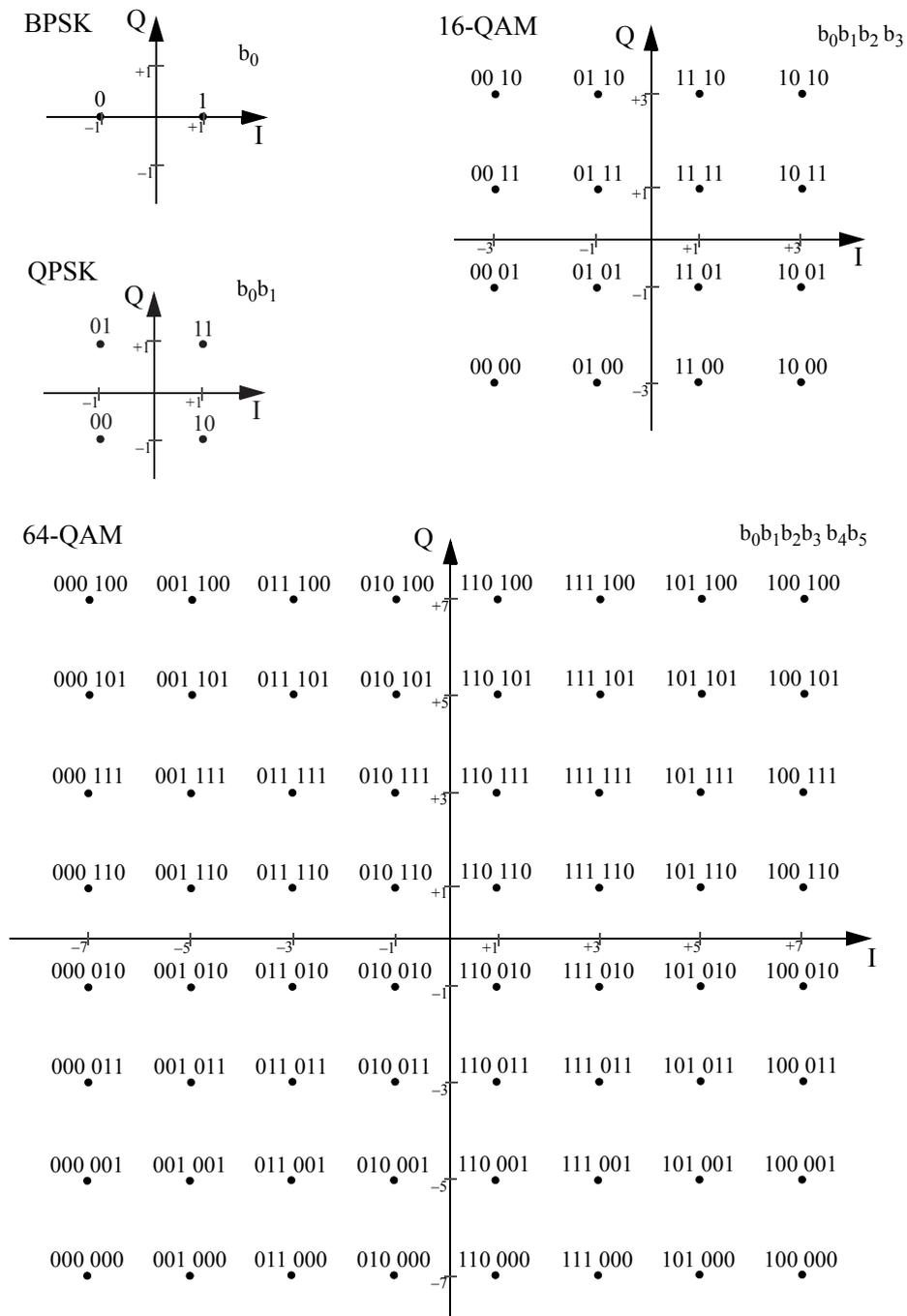
Figure 1.7: BPSK, QPSK, 16-QAM and 64-QAM Gray-coded constellation mappings

coding in 802.11 WLANs is thus based on the PEC paradigm.

Frame reception in 802.11 WLANs is in an "all or nothing" fashion with frames having PHY or CRC errors being discarded and only frames received without error retained, and hence the channel provided by 802.11 frames is conventionally modelled as a PEC. However, the fraction of incorrect bits in frames with CRC errors is observed to be small in experimental measurements. In Chapter 3 we model the channel provided by outdoor 802.11 links as a hybrid binary symmetric/packet erasure channel. Using experimental measurements taken in an outdoor environment, we demonstrate that the channel provided by corrupted frames alone (i.e. ignoring frames with PHY errors and frames received without error) can be accurately modelled as a BSC provided appropriate pre- and post-processing is carried out. The channel provided by corrupted frames and other frames combined can be accurately modelled as a hybrid binary symmetric/packet erasure channel. Importantly, we find that this hybrid channel offers capacity increases of more than 100% compared to a conventional PEC over a wide range of RSSIs. This is a striking observation as it indicates that the potential exists for significant network throughput gains if the information contained in 802.11 corrupted packets is exploited.

In Chapter 4 we consider the potential performance gains of adopting a BSC paradigm for multi-destination aggregations in 802.11 WLANs. We consider two BSC-based higher-layer coding approaches, i.e. superposition coding and a simpler time-sharing coding, for multi-destination aggregated packets. The performance results for both unicast and multicast traffic, taking account of important MAC layer overheads such as contention time and collision losses, demonstrate that increases in network throughput of more than 100% are possible over a wide range of channel conditions, and that the simpler time-sharing approach yields most of these gains and has minimal loss of optimality. Importantly, these performance gains involve software rather than hardware changes, and thus essentially come for "free".

In Chapter 5 we further consider the proportional fair allocation of high-layer coding rates and airtimes in 802.11 WLANs. We consider BSC lossy links and take delay constraints into account. We show that the joint optimisation of coding rate and airtime decomposes into decoupled allocation tasks, i.e. a layered approach of separating MAC scheduling and higher-layer coding rate selection is optimal. This property of 802.11 differs from and contrasts with TDMA wireless networks. Further, we establish that the proportional fair coding rate and airtime allocation (i) assigns equal total airtime (i.e. airtime including both successful and failed transmissions) to every station in a WLAN, (ii) the station airtimes

sum to unity (ensuring operation at the rate region boundary), and (iii) the optimal coding rate is selected to maximise goodput (treating packets decoded after the delay deadline as losses).

1.3 Publications

Journals:

- **Xiaomin Chen** and D. J. Leith, "Proportional Fair Coding for 802.11 WLANs ," to appear in IEEE Wireless Communications Letters, 2012.

- **Xiaomin Chen**, V. G. Subramanian and D. J. Leith, "PHY Modulation/Rate Control for Fountain Codes in 802.11a/g WLANs," accepted by Physical Communication, 2012.

- P. Karumbu, **Xiaomin Chen** and D. J. Leith, "Proportional Fair Coding for Wireless Mesh Networks," submitted to IEEE/ACM Transactions on Networking, 2011.

- **Xiaomin Chen** and D. J. Leith, "Frames in Outdoor 802.11 WLANs Provide a Hybrid Binary Symmetric/Packet Erasure Channel," submitted to IEEE Communications Letters, 2012.

Conferences:

- **Xiaomin Chen**, V. G. Subramanian and D. J. Leith, "Binary Symmetric Channel Based Aggregation with Coding for 802.11n WLANs," IEEE Broadnets 2010, Athens, Greece, Oct 2010.

- P. Karumbu, **Xiaomin Chen** and D. J. Leith, "Utility Optimal Coding for Packet Transmission over Wireless Networks - Part I: Networks of Binary Symmetric Channels," 49th Annual Allerton Conference on Communication, Control, and Computing, Monticello, Illinois, USA, Sep 2011.

- P. Karumbu, **Xiaomin Chen** and D. J. Leith, "Utility Optimal Coding for Packet Transmission over Wireless Networks - Part II: Networks of Packet Erasure Channels," 49th Annual Allerton Conference on Communication, Control, and Computing, Monticello, Illinois, USA, Sep 2011.

- **Xiaomin Chen**, V. G. Subramanian and D. J. Leith, "An Upper Bound on the Packet Error Rate of 802.11a/g Viterbi Soft Decision Decoding in the AWGN Channel," IEEE/IFIP Wireless Days 2012, Dublin, Ireland, Nov. 2012.

Chapter 2

PHY Rate Control for Fountain Codes in 802.11a/g WLANs

2.1 Introduction

In this chapter we consider the joint performance of fountain codes concatenated with 802.11a/g OFDM PHY modulation/coding. Fountain codes have been the subject of much interest in recent years, both in the context of wireless video multicast (e.g. see [11, 39]) and of network coding and joint coding/routing (e.g. see [4]). However, although there is a wealth of literature on the subject of fountain codes, there have been relatively few studies of fountain code performance in 802.11 WLANs and even fewer on the cross-layer trade-offs between higher-layer fountain coding and PHY layer modulation and FEC coding selection. The use of fountain codes in 802.11 WLANs is nevertheless of considerable interest in view of their ubiquitous deployment and the trend towards their use for multimedia distribution within the home and elsewhere.

Fountain coding is a higher layer technology. In a packet-switch network, the fountain code symbols are typically packets and a cyclic redundancy check (CRC) checksum is used to mark each packet as either erased or error-free. This abstraction essentially provides an interface to the PHY layer which in turn interfaces with the actual wireless channel. In the erasure channel model the probability that a packet is erased is strongly coupled to the choice of modulation and FEC coding scheme used at the

PHY layer. To understand system performance, it is therefore necessary to consider the joint fountain coding and PHY modulation/coding performance.

Prior work on the joint performance of a concatenated higher layer fountain code and PHY modulation and FEC code includes: [7] which looks at the tradeoff in a general single-user wireless setting, while in our setting we consider multicast rather than single-user operation and include the specific constraints imposed by 802.11 technology; [55] which considers multicast and the use of Raptor codes in 802.11-like WLANs, but their analysis is based on an access mechanism different from the IEEE 802.11 standards and does not consider the rate-selection problem in comparison to the uncoded setting; [14] which considers a fundamentally different multi-receiver scenario from the standard multicast setting with the objective for a group of M client stations to cooperatively receive a block of N packets rather than for each individual client to receive all N packets successfully.

In this chapter we carry out a detailed study on the optimal choices of PHY modulation/coding with and without higher layer fountain coding in standard 802.11 WLANs. We consider optimality both in terms of maximising goodput and minimising energy, and results are presented for a wide range of channel conditions. In contrast to studies in cellular networks [40, 39], we find that in 802.11a/g WLANs the PHY rate that optimizes uncoded multicast performance is also close to optimal for fountain-coded multicast traffic. This is potentially an important observation as it indicates that in 802.11a/g WLANs cross-layer rate control for higher-layer fountain coding concatenated with physical layer modulation and FEC coding would bring few benefits and PHY layer rate control can be carried out without regard to the use of fountain coding at higher layers.

2.2   Fountain Codes

Fountain codes are a class of block codes for channels with erasures. The erasure channel [15] is a communication channel over which the transmitted symbols are either received error-free or not received (erased). A fountain-encoded symbol is a random linear combination of source symbols within a code block. The original source symbols can ideally be recovered from any subset of encoded symbols of size equal to or slightly larger than the number of source symbols. Thus, regardless of the underlying channel erasure rate, the encoder keeps producing as many encoded symbols as needed until a sufficient number of encoded symbols are received to recover the source block. The coding rate is determined on the fly, and fountain codes are therefore also known as rateless codes. In our

analysis, we consider using the equiprobable random linear fountain code [42] and a class of systematic fountain codes based on it [59].

Assume that the original block has K source symbols, $\boldsymbol{s} = \{s_1, s_2, \cdots, s_K\}$. Each symbol is a bit vector. For coded symbol t_n, $n = 1, 2, \cdots$, the equiprobable random linear fountain encoder generates a K-bit binary vector $\boldsymbol{G}_n = \{g_{1n}, g_{2n}, \cdots, g_{Kn}\}$, in which each element is selected to be 0 or 1 uniformly at random. The coded symbol t_n is the bitwise sum, modulo 2, of the source symbols for which $g_{kn} = 1, k = 1, 2, \cdots, K$,

$$t_n = \sum_{k=1}^{K} s_k g_{kn} = \boldsymbol{s} \boldsymbol{G}_n^T. \tag{2.1}$$

This sum can be carried out by successively exclusive-or-ing the source symbols together. Fig. 2.1 illustrates an example of the associated generator matrix $\mathbf{G}$. Coded symbol t_n corresponds to a column $\boldsymbol{G}_n^T$ of size K, and the matrix grows as more coded symbols are generated. When coded symbols are transmitted over an erasure channel, some of them are erased. The received coded symbol vector is

$$\boldsymbol{t} = \boldsymbol{s} \mathbf{G}' \tag{2.2}$$

where $\mathbf{G}'_{\mathbf{K} \times \mathbf{N}}$ is the generator matrix with columns corresponding to the N received symbols. Provided that $\mathbf{G}'$ is full column rank we can use Gaussian elimination to recover source symbols $\boldsymbol{s}$. $\mathbf{G}'$ will be full rank with high probability for N slightly larger than K [59].

A class of systematic fountain codes are proposed in [59]. The block of source symbols are first transmitted uncoded, and the subsequent transmissions are coded symbols constructed using an equiprobable random linear fountain code. This class of fountain codes stochastically minimizes the number of received symbols necessary for recovering source symbols over a large class of fountain codes, including LT and Raptor codes [38, 57], and has reasonable decoding complexity for small block sizes.

2.3   802.11a/g Packet Erasure Model

In this section we consider the performance of the packet erasure channel provided by 802.11a/g links, over which an 802.11 packet is either erased (due to collisions or noise errors) or correctly received. Given a physical channel model, for each combination of modulation and FEC coding rate provided by 802.11a/g OFDM PHY, we derive the relationship between the packet error rate (PER) and signal to noise ratio (SNR).
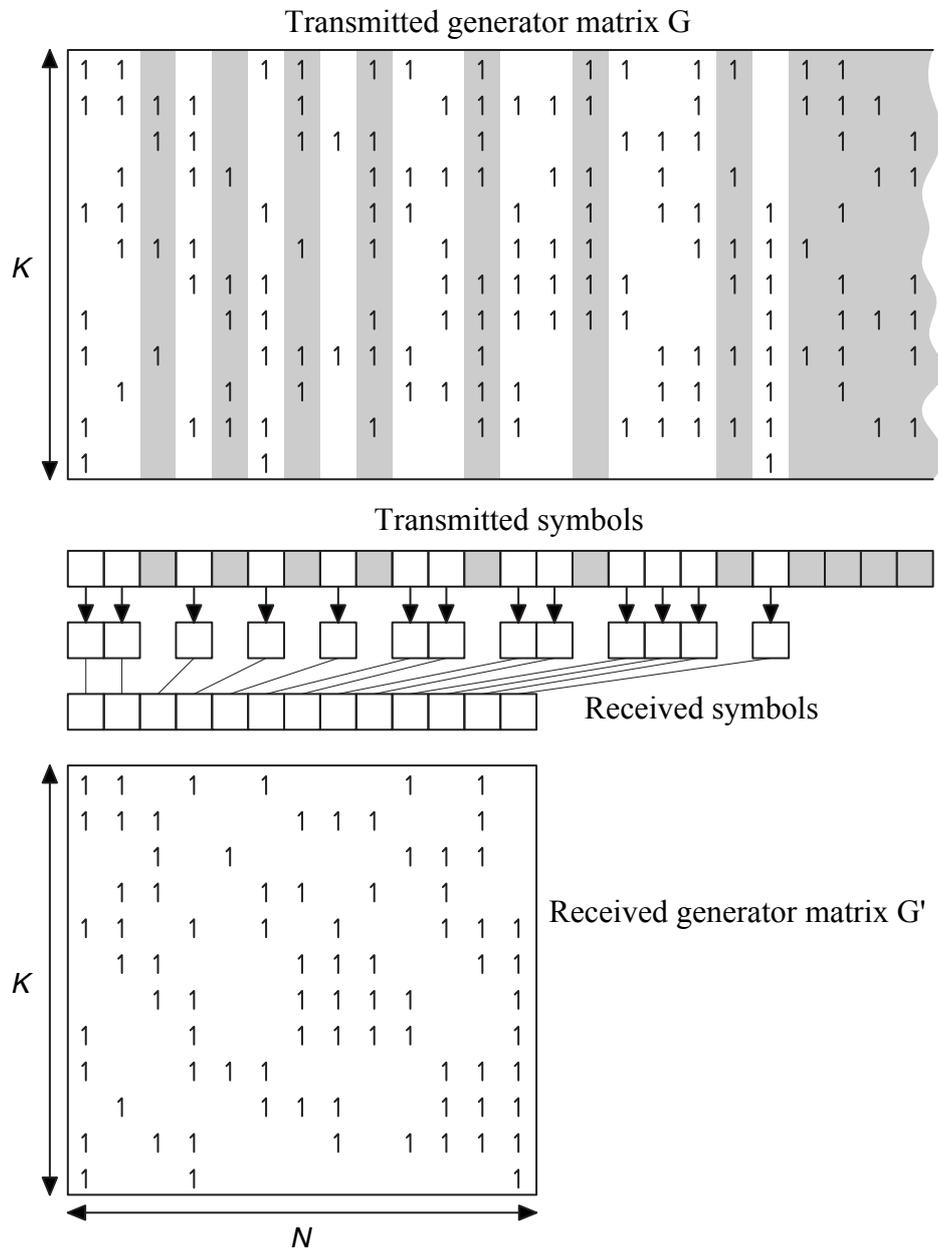
Figure 2.1: Schematic illustration of the transmitted and received generator matrices $\mathbf{G}$ and $\mathbf{G}'$ of equiprobable random linear fountain code. Gray shading indicates symbols which are erased in the channel.

The FEC decoder at the receiver follows the demodulator in the 802.11a/g OFDM PHY, as illustrated in Fig. 1.4. The PER thus depends on the performance of both the demodulation and the decoding. The 802.11a/g OFDM PHY uses the Viterbi algorithm to decode convolutional codes. In terms of the output of the demodulator, *i.e.* a hard decision demodulator outputs only 1's or 0's, while a soft decision demodulator may output any of the values in-between which indicates the distance to the decision boundary, the Viterbi decoder performs hard or soft decision decoding. We derive the PER expressions for both hard and soft decision decoding in an additive white Gaussian noise (AWGN) channel, and for the hard decision decoding in a Rayleigh channel.

2.3.1 AWGN Channel Model

The Viterbi decoding algorithm searches through the trellis of a convolutional code to determine a path that maximises the probability that the received signals represent the coded bit sequence expressed by this path [62]. For this purpose, the decoder associates a metric with each path. The hard and soft decision decoders respectively use the Hamming distance and the Euclidean distance as a path metric. An upper bound on the PER in the Viterbi decoding is given by [53]

$$P_p < 1 - (1 - P_e)^L \tag{2.3}$$

where L is the length of packet in bits, and P_e is the union bound on the first-event error probability, which is calculated as follows:

Viterbi Hard Decision Decoding (HDD)

We begin by reviewing the derivation of the bit error rate (BER) for hard decision demodulation in an AWGN channel when using each of the modulation schemes provided by the 802.11a/g OFDM PHY [21]. The quality of a received signal is indicated by the SNR, which is given by

$$\gamma = \frac{E_b}{N_0 B T_b} \tag{2.4}$$

where E_b is the energy per bit; N_0 is the noise density in W/Hz; T_b is the transmission duration per bit; B is the channel bandwidth, which is 20MHz in the 802.11a/g OFDM PHY [30]. The SNR per bit is

$$\gamma_b = \frac{E_b}{N_0} = \gamma \cdot B T_b = \gamma \cdot \frac{B}{R} \tag{2.5}$$

where $R = 1/T_b$ is the bit transmission rate in Mbps. The SNR per symbol is then

$$\gamma_s = \gamma_b \cdot k \tag{2.6}$$

where k is the number of bits per modulated symbol, which is determined by the modulation scheme used in the communication system.

For the hard decision demodulation in an AWGN channel, the symbol error rate (SER) P_s of all modulation schemes used in the 802.11a/g OFDM PHY can be derived from the SER function of N-ary pulse-amplitude modulation (N-PAM),

$$P_{sN-PAM} = 2\left(1 - \frac{1}{N}\right) Q\left(\sqrt{\frac{6}{N^2 - 1} \cdot \gamma_s}\right), \tag{2.7}$$

where the function $Q(\cdot)$ is the tail probability of the standard normal distribution [21],

$$Q(x) = \frac{1}{2}\mathrm{erfc}\left(\frac{x}{\sqrt{2}}\right). \tag{2.8}$$

Since BPSK can be viewed as 2-ary PAM and each BPSK symbol modulates only one bit, the BER of BPSK is,

$$P_{bBPSK} = P_{sBPSK} = P_{s2-PAM}. \tag{2.9}$$

The SER of N-ary quadrature-amplitude modulation (N-QAM) can be expressed in form of PAM,

$$P_{sN-QAM} = 1 - \left(1 - P_{s\sqrt{N}-PAM}\left(\frac{1}{2}\gamma_s\right)\right)^2. \tag{2.10}$$

Since QPSK can be viewed as 4-ary QAM,

$$P_{sQPSK} = P_{s4-QAM}. \tag{2.11}$$

As the constellation mapping in 802.11a/g OFDM PHY is based on Gray code, the BER of N-ary QAM is [47],

$$P_{bN-QAM} \approx \frac{1}{k}P_{sN-QAM}, \tag{2.12}$$

where k is the number of bits per modulated symbol, $k = \log_2(N)$.

As the demodulation is followed by the Viterbi decoder, we need to adjust the demodulation BER to take account of the error correction provided by convolutional coding. With the Viterbi HDD, the union bound on the first event error probability P_e is given by [62]

$$P_{eHDD} \leq \sum_{d=d_{free}}^{\infty} \alpha_d \cdot P_2(d) \tag{2.13}$$

19

where d_{free} is the minimum free distance of the convolutional code; α_d is the total number of paths with degree d; $P_2(d)$ is the probability that an incorrect path with degree d is selected (assuming that the all-zero path is the correct path transmitted), which is given by

$$P_2(d) = \begin{cases} \sum\limits_{k=(d+1)/2}^{d} \binom{d}{k} \cdot P_b^k \cdot (1 - P_b)^{d-k} & \text{if d is odd} \\ \frac{1}{2} \cdot \binom{d}{d/2} \cdot P_b^{d/2} \cdot (1 - P_b)^{d/2} + \sum\limits_{k=d/2}^{d} \binom{d}{k} \cdot P_b^k \cdot (1 - P_b)^{d-k} & \text{if d is even} \end{cases} \qquad (2.14)$$

with P_b being the HDD demodulation BER. Note that for a packet payload size of 992 bytes used in the following simulations, the upper bound on the HDD first-event error probability P_e is always tight for both the AWGN and Rayleigh channels [33].

The union bound P_e is practically approximated by the summation of the first few dominant terms. In our simulations, we consider the first 10 terms. The values of α_d and d_{free} for the three convolutional coding rates used in 802.11a/g OFDM PHY are listed in Table 2.1 [13, 22]:

FEC coding rate	PHY rate (Mb/s)	d_{free}	α_d
1/2	6, 12, 24	10	11, 0, 38, 0, 193, 0, 1331, 0, 7275, 0
3/4	9, 18, 36, 54	5	8, 31, 160, 892, 4512, 23307, 121077, 625059, 3234886, 16753077
2/3	48	6	1, 16, 48, 158, 642, 2435, 9174, 34705, 131585, 499608

Table 2.1: d_{free} and α_d values for the three convolutional coding rates used in 802.11a/g OFDM PHY

Viterbi Soft Decision Decoding (SDD)

In this section, we derive the probability $P_2(d)$ that a path with degree d is incorrectly selected in the Viterbi soft decision decoding algorithm for each of the modulation schemes used in the 802.11a/g OFDM PHY over an AWGN channel. For BPSK and QPSK, we give an exact analytical expression. But for 16-QAM and 64-QAM because of the difficulty in exact evaluation of this error probability, we derive an upper bound and a lower bound on it. A bound on the first event error probability can then be calculated based on this error probability and the convolutional code parameters. Finally, an upper bound on the packet error rate is then obtained using Eqn. (2.3).

In the Viterbi SDD the path metric is the Euclidean distance *i.e.* the distance between two symbols in the modulation constellation. Hence the probability of choosing a wrong path in a pairwise comparison with the all-zero path depends not only on the degree d, but also on the modulation scheme used in

the communication system, *i.e.* the series of modulation symbols constructed along that path. The union bound on the first-event error probability in the SDD is then expressed as

$$P_{eSDD} \leq \sum_{d=d_{free}}^{\infty} \sum_{i=1}^{\alpha_d} P_2(d_s^i(d)) \tag{2.15}$$

where $P_2(d_s^i(d))$ is the probability that the ith d-degree path with $d_s^i(d)$ symbols different from the all-zero path is wrongly chosen.

The derivation of $P_2(d_s^i(d))$ for a coherent M-ary QAM demodulator in an AWGN channel is described in [48], given by,

$$P_2(d_s^i(d)) = Q\left(\sqrt{\frac{\sum_{l=1}^{d_s^i(d)} \| C_l - C_0 \|^2}{2N_0}}\right) = Q\left(\sqrt{\frac{\sum_{l=1}^{d_s^i(d)} D_l^2}{2N_0}}\right) \tag{2.16}$$

where C_0 is the all-zero symbol, *i.e.* the symbol modulating only bit '0'; C_l is the lth symbol that differs from C_0 along the path; N_0 is the noise spectral density; $D_l = \| C_l - C_0 \|$ is the Euclidean distance between a non-zero symbol C_l and the all-zero symbol C_0 in the constellation.

The convolutional coded bit sequence is converted into constellation points using Grey-coded constellation mapping rules in the 802.11a/g OFDM PHY. According to the mapping rules illustrated in Fig. 1.7, $P_2(d_s^i)$ can be evaluated for each of the modulation schemes used in the 802.11a/g OFDM PHY as follows:

- BPSK:

  Since each symbol modulates only one bit in BPSK, for every path with degree d we have $d_s^i = d$, and

  $$P_2(d_s^i)_{BPSK} = Q\left(\sqrt{\frac{\sum_{l=1}^{d} \| C_l - C_0 \|^2}{2N_0}}\right) = Q\left(\sqrt{\frac{d(2A)^2}{2N_0}}\right) \tag{2.17}$$

  where A is one unit of the grid in the constellation diagram. Suppose that γ_b is the received SNR per coded bit, we have

  $$\gamma_b = \frac{E_s}{N_0 \cdot \log_2 M} \tag{2.18}$$

  where E_s is the average energy per coded symbol, and M is the number of bits modulated on each symbol. The SNR per coded bit for BPSK is given by

  $$\gamma_{bBPSK} = \frac{2 \cdot \frac{1}{2} \cdot (A^2)}{N_0 \cdot \log_2 2} = \frac{A^2}{N_0} \tag{2.19}$$

21

Substituting (2.19) into (2.17) yields

$$P_2(d_s^i)_{BPSK} = Q\left(\sqrt{2d\gamma_b}\right) \tag{2.20}$$

The union bound on the first-event error probability is then given by

$$P_{eSDD-BPSK}(\gamma_b) \leq \sum_{d=d_{free}}^{\infty} \alpha_d Q\left(\sqrt{2d\gamma_b}\right) \tag{2.21}$$

- QPSK:

In the rectangular Gray-coded QPSK constellation, the squared Euclidean distance of symbol '11' is twice that of symbol "01" or '10'. Hence for each path with degree d, no matter how the d '1's are distributed, the path value $\sum_{l=1}^{d_s^i} \| C_l - C_0 \|^2$ is the same. We can calculate $P_2(d_s^i)$ by simply assuming that in the ith path every non-zero symbol contains only one '1', which is either '01' or '10',

$$P_2(d_s^i)_{QPSK} = Q\left(\sqrt{\frac{\sum_{l=1}^{d} \| C_l - C_0 \|^2}{2N_0}}\right) = Q\left(\sqrt{\frac{d(2A)^2}{2N_0}}\right) \tag{2.22}$$

The SNR per coded bit for QPSK is

$$\gamma_{bQPSK} = \frac{4 \cdot \frac{1}{4} \cdot (\sqrt{2}A)^2}{N_0 \cdot \log_2 4} = \frac{A^2}{N_0} \tag{2.23}$$

Then, P_{eSDD} for QPSK is bounded by

$$P_{eSDD-QPSK}(\gamma_b) \leq \sum_{d=d_{free}}^{\infty} \alpha_d Q\left(\sqrt{2d\gamma_b}\right) \tag{2.24}$$

- 16-QAM:

For 16-QAM and 64-QAM, the value of $\sum_{l=1}^{d_s^i} \| C_l - C_0 \|^2$ is not equal for every d-degree path. But we can establish a range for $P_2(d_s^i)$. An upper bound on $P_2(d_s^i)$, *i.e.* the highest probability that a path with degree d is selected, can be established by determining a combination of d_s^U symbols that minimises $\sum_{l=1}^{d_s^i} \| C_l - C_0 \|^2$. Also a lower bound, *i.e.* the lowest probability that a path with degree d is selected, can be established by determining a combination of d_s^L symbols that maximises this value.

Table 2.2 lists the squared minimum and maximum Euclidean distances for symbols modulating i, $i = 1, \cdots, 4$, 1's in the 16-QAM rectangular Grey-coded constellation. To obtain an upper bound, we need to minimise $\sum_{i=1}^{d_s^i} D_{min}^2$. By observing that $4A^2 + 4A^2 = 8A^2$, $4A^2 + 4A^2 + 4A^2 < 20A^2$ and $4A^2 + 4A^2 + 4A^2 + 4A^2 < 32A^2$, the optimal symbol combination for an upper bound

22

	D^2_{min}	D^2_{max}
1 bit 1 symbol	$4A^2$	$36A^2$
2 bit 1's symbol	$8A^2$	$72A^2$
3 bit 1's symbol	$20A^2$	$52A^2$
4 bit 1's symbol	$32A^2$	$32A^2$

Table 2.2: D^2_{min} and D^2_{max} in the 16-QAM rectangular Grey-coded constellation of 802.11a/g

should include only the 1 bit '1' symbol, the 2 bit '1's symbol and the all-zero symbol. Similar to the upper bound, the optimum symbol combination for a lower bound should also only include the 1 bit '1' symbol, the 2 bit '1's and the all-zero symbol so as to maximise $\sum_{i=1}^{d^i_s} D^2_{max}$.

We calculate the upper bound on $P_2(d^i_s)$ for 16-QAM by simply assuming that the non-zero symbols in the ith d-degree path are either '0001' or '0100' which contains only one bit '1' and has D^2_{min}, and thus

$$P_2(d^U_s)_{16-QAM} = Q\left(\sqrt{\frac{\sum_{l=1}^d \| C_l - C_0 \|^2}{2N_0}}\right) = Q\left(\sqrt{\frac{d(2A)^2}{2N_0}}\right) \tag{2.25}$$

Similarly, the lower bound on $P_2(d^i_s)$ for 16-QAM is

$$P_2(d^L_s)_{16-QAM} = Q\left(\sqrt{\frac{\sum_{l=1}^d \| C_l - C_0 \|^2}{2N_0}}\right) = Q\left(\sqrt{\frac{d(6A)^2}{2N_0}}\right) \tag{2.26}$$

The SNR per coded bit for 16-QAM is

$$\gamma_{b16-QAM} = \frac{4 \cdot \frac{1}{16} \cdot \left((\sqrt{2}A)^2 + (\sqrt{10}A)^2 + (\sqrt{10}A)^2 + (3\sqrt{2}A)^2\right)}{N_0 \cdot \log_2 16} = \frac{5A^2}{2N_0} \tag{2.27}$$

Substituting (2.27) into (2.25) and (2.26) yields,

$$Q\left(\sqrt{\frac{36}{5}d\gamma_b}\right) \leq P_2(d^i_s)_{16-QAM} \leq Q\left(\sqrt{\frac{4}{5}d\gamma_b}\right) \tag{2.28}$$

P_{eSDD} is thus bounded by

$$\sum_{d=d_{free}}^{\infty} \alpha_d Q\left(\sqrt{\frac{36}{5}d\gamma_b}\right) \leq P_{eSDD-16QAM}(\gamma_b) \leq \sum_{d=d_{free}}^{\infty} \alpha_d Q\left(\sqrt{\frac{4}{5}d\gamma_b}\right) \tag{2.29}$$

- 64-QAM:

  Similar to the analysis for 16-QAM, Table 2.3 lists D^2_{min} and D^2_{max} for symbols modulating i, $i = 1, \cdots, 6$, bit 1's in the 64-QAM rectangular Grey-coded constellation. The upper and

	D_{min}^2	D_{max}^2
1 bit 1 symbol	$4A^2$	$196A^2$
2 bit 1's symbol	$8A^2$	$392A^2$
3 bit 1's symbol	$20A^2$	$340A^2$
4 bit 1's symbol	$32A^2$	$296A^2$
5 bit 1's symbol	$116A^2$	$244A^2$
6 bit 1's symbol	$200A^2$	$200A^2$

Table 2.3: D_{min}^2 and D_{max}^2 in the 64-QAM rectangular Grey-coded constellation of 802.11a/g

lower bounds on $P_2(d_s^i)$ are respectively established by a symbol combination of the all-zero symbol, the 1 bit '1' symbol and the 2 bit '1's symbol so as to achieve D_{min}^2 or D_{max}^2. P_{eSDD} is therefore bounded by

$$\sum_{d=d_{free}}^{\infty} \alpha_d Q\left(\sqrt{14d\gamma_{cb}}\right) \leq P_{eSDD-64QAM}(\gamma_b) \leq \sum_{d=d_{free}}^{\infty} \alpha_d Q\left(\sqrt{\frac{2}{7}d\gamma_b}\right) \qquad (2.30)$$

Fig. 2.2(a) shows the resulting HDD PER and the SDD PER based on the derived upper bound on P_{eSoft} versus SNR for each of the 802.11a/g OFDM PHY modulation/coding schemes with a packet length of 1024 bytes. It can be seen that the Viterbi SDD provides a lower PER than the HDD for any given SNR.

2.3.2 Rayleigh Channel Model

For Nakagami-m fast-fading channels the hard decision demodulation BER for each of the modulation schemes provided by the 802.11a/g OFDM PHY can be calculated following the analysis in [58]. We note that a Rayleigh channel corresponds to a Nakagami channel with $m = 1$. The average BER of BPSK hard decision demodulation in a Nakagami-m fast fading channel with integer $m = 1$ is given by

$$P_{bBPSK} = \frac{1}{2}\left(1 - \mu_{BPSK}\sum_{k=0}^{m-1}\binom{2k}{k}\left(\frac{1-\mu_{BPSK}^2}{4}\right)^k\right)\Bigg|_{m=1} \qquad (2.31)$$
$$= \frac{1}{2}\left(1 - \mu_{BPSK}\right)$$

where

$$\mu_{BPSK} = \sqrt{\frac{\overline{\gamma}_s}{m + \overline{\gamma}_s}} \qquad (2.32)$$

24

with $\overline{\gamma}_s$ being the average SNR per symbol.

The average BER of N-QAM ($N \geq 4$, and QPSK corresponds to an equivalent 4-QAM) hard decision demodulation in Nakagami-m fast fading channels with integer $m = 1$ is given by

$$P_{bN-QAM} = 4\left(\frac{\sqrt{N}-1}{\sqrt{N}}\right)\left(\frac{1}{\log_2 N}\right)\sum_{i=1}^{\sqrt{N}/2}\frac{1}{2}\left(1 - \mu_{iN-QAM}\sum_{k=0}^{m-1}\binom{2k}{k}\left(\frac{1-\mu_{iN-QAM}^2}{4}\right)^k\right)\Bigg|_{m=1}$$

$$= 4\left(\frac{\sqrt{N}-1}{\sqrt{N}}\right)\left(\frac{1}{\log_2 N}\right)\sum_{i=1}^{\sqrt{N}/2}\frac{1}{2}\left(1 - \mu_{iN-QAM}\right)$$

(2.33)

where

$$\mu_{iN-QAM} = \sqrt{\frac{1.5(2i-1)^2\overline{\gamma}_s}{m(N-1)+1.5(2i-1)^2\overline{\gamma}_s}}$$

(2.34)

With the average demodulation BER, the HDD PER can be calculated using Eqns. (2.3), (2.13) and (2.14). Fig. 2.2(b) shows the corresponding HDD PER vs SNR curves for a Rayleigh channel.

2.4 Performance Modelling

We consider an 802.11a/g single-hop downlink multicast network with one access point (AP) and M client stations. Without higher-layer coding, to achieve reliable multicast it is necessary for each client station to transmit higher-layer acknowledgement packets to inform the multicast sender of which packets were noise-corrupted and thus need to be retransmitted. As the number M of client stations increases, the probability of a given packet being successfully received over a noisy channel by all stations becomes small and we may quickly end up in a situation where almost every packet requires to be retransmitted at least once and perhaps multiple times. In contrast, fountain coding allows a block of N packets to be recovered, on average, from reception of *any* $N + \delta$ coded packets, where δ is the decoding overhead counted in terms of number of extra packets that need to be transmitted for decoding with high probability. Fountain coding therefore fundamentally changes the scaling behaviour of network performance with the number M of client stations. In this section we derive analytic expressions for the mean number of transmissions and acknowledgements required for all M clients to successfully receive a block of N multicast packets over a noisy channel with and without fountain coding.
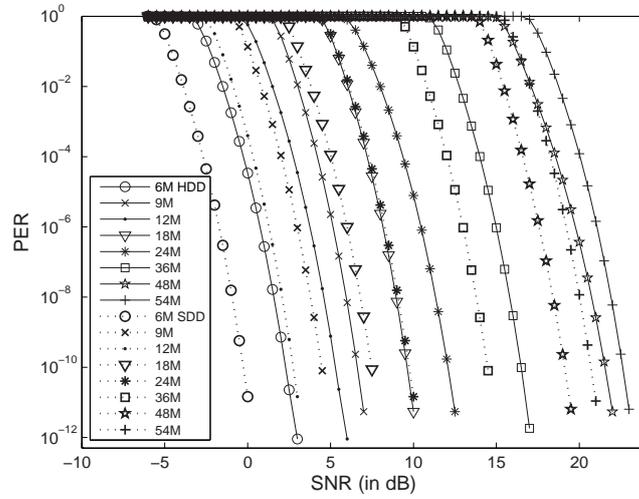
(a) AWGN channel
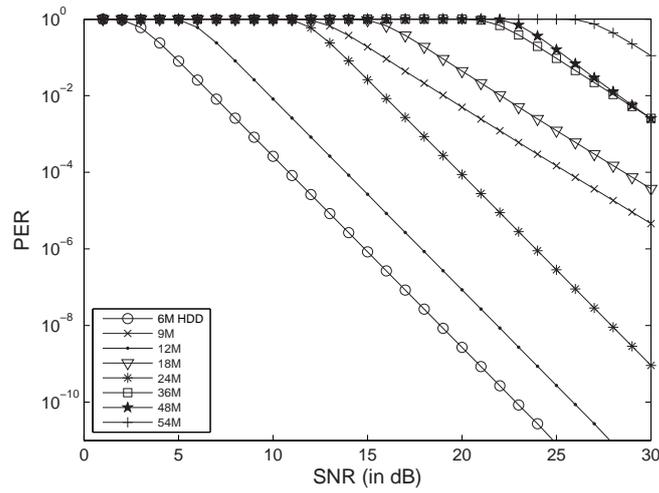
(b) Rayleigh channel

Figure 2.2: 802.11a/g packet error rate (PER) vs channel SNR for the range of modulation and coding schemes available in 802.11a/g. HDD and SDD indicate hard and soft decision decoding, respectively.

2.4.1 Fountain Encoding

We consider both systematic and non-systematic fountain codes and hence the analysis is relevant to both Raptor and LT codes. The non-systematic fountain code we use in our numerical results is the equiprobable random linear fountain code. The systematic fountain code we consider is the optimal rateless code proposed in [59], which, as noted previously, stochastically minimizes the number of received packets necessary for recovery of coded packets over a large class of fountain codes

Fountain coding is carried out at the application layer. An application file is first partitioned into multiple equal-length blocks. Each block is further partitioned into equal-length packets. Packets in each block are then encoded by a fountain encoder. A block sequence number is added in the header of each coded packet to indicate which block it belongs to. In addition, the header contains the pseudo-random seed used to generate the Bernoulli(1/2) random vector associated with the coded packet, which is required to reconstruct the generator matrix for decoding. We account for these encoding overheads in our analysis, see Section 2.4.6.

2.4.2 Higher-layer ACK/NACK Modelling

For reliable multicast without fountain coding it is necessary for clients to use higher-layer ACK/NACK transmissions to inform the multicast sender of which packets have been successfully received. To derive the corresponding goodput expression we assume the use of the following signaling scheme. Namely, after transmission of a block of N packets, each client checks whether it has successfully received the whole block or not. If it has, an application layer ACK is transmitted to inform the multicast sender. Otherwise, an application layer NACK is transmitted which identifies the missing packets. Since our interest is in obtaining an upper bound on performance, we assume that higher-layer ACK/NACK transmissions are scheduled in such a way that they never collide and that their transmissions are error-free. At the next round, the multicast sender retransmits the union of any packets that were not received, and this is repeated until all clients have received the block of N packets. When fountain coding is used, we assume that each client transmits an application layer ACK on successfully decoding a block of N packets. No further signalling is required, and again it is assumed that the higher-layer ACKs never collide and that their transmissions are error-free.

2.4.3   Mean #Transmissions without Higher-layer Fountain Coding

Let p denote the PER– given a physical channel model and the channel SNR this can be obtained from Fig. 2.2. We assume that all clients have the equal error probability p and client packet receptions are independent of each other. Without higher-layer fountain coding, the probability that client $j \in \{1, 2, \cdots, M\}$ receives packet $i \in \{1, 2, \cdots, N\}$ after $k \geq 1$ transmissions is

$$\mathbb{P}\left\{r_{i,j} = k\right\} = p^{k-1}(1-p) \tag{2.35}$$

in which $r_{i,j}$ is the number of transmissions, given a pair i and j, and is a Geometric$(1-p)$ random variable. The number of transmissions required before the ith packet is received by all M clients is the random variable

$$t_i = \max_{j \in \{1,2,\cdots,M\}} r_{i,j} \tag{2.36}$$

and the total number of transmissions for M clients to receive the full block of N packets is then

$$T_{ucd} = \sum_{i=1}^{N} \max_{j \in \{1,2,\cdots,M\}} r_{i,j} \tag{2.37}$$

The mean of the total number of transmissions is given by

$$
\begin{aligned}
E[T_{ucd}] &= \sum_{i=1}^{N} E[\max_{j \in \{1,2,\cdots,M\}} r_{i,j}] \\
&\overset{(a)}{=} N\left(1 + \sum_{t=1}^{\infty}\left(1 - \left(\sum_{x=1}^{t}(1-p)p^{x-1}\right)^M\right)\right) \\
&= N\left(1 + \sum_{t=1}^{\infty}\left(1 - (1 - p^t)^M\right)\right) \\
&= N\left(1 + \sum_{t=1}^{\infty}\sum_{i=1}^{M}\binom{M}{i}(-1)^{i+1}(p^t)^i\right) \\
&= N\left(1 + \sum_{i=1}^{M}\binom{M}{i}(-1)^{i+1}\sum_{t=1}^{\infty}(p^i)^t\right) \\
&= N\left(1 + \sum_{i=1}^{M}\binom{M}{i}(-1)^{i+1}\frac{p^i}{1-p^i}\right)
\end{aligned}
\tag{2.38}
$$

in which the equality (a) follows from the derivation of the mean of the maximum of M independent negative binomial random variables of order n (see Appendix B) and the fact that the geometric distribution is a special case of negative binomial distribution with $n = 1$.

The number of higher-layer acknowledgement packets (including ACKs and NACKs) is

$$A_{ucd} = \sum_{j=1}^{M} \max_{i \in \{1,2,\cdots,N\}} r_{i,j} \tag{2.39}$$

and the mean of it is then given by

$$
\begin{aligned}
E[A_{ucd}] &= M\left(1 + \sum_{t=1}^{\infty}\left(1 - \left(\sum_{x=1}^{t}(1-p)p^{x-1}\right)^{N}\right)\right) \\
&= M\left(1 + \sum_{t=1}^{\infty}\left(1 - (1-p^t)^N\right)\right) \\
&= M\left(1 + \sum_{i=1}^{N}\binom{N}{i}(-1)^{i+1}\frac{p^i}{1-p^i}\right)
\end{aligned}
\tag{2.40}
$$

.

2.4.4 Mean #Transmissions with Non-systematic Fountain Code

With fountain coding, to receive a block of N packets the requirement becomes that every client has on average to successfully receive $N + \delta$ packets, where δ is the coding overhead (usually a fixed sublinear function of N depending on the code). The probability that client j receives $N + \delta$ packets after $k \geq N + \delta$ transmissions is

$$\mathbb{P}\{y_j = k\} = \binom{k-1}{N+\delta-1}p^{k-N-\delta}(1-p)^{N+\delta-1}(1-p) \tag{2.41}$$

where y_j is the number of transmissions before client j receives $N + \delta$ packets, which is a Negative Binomial $(N + \delta, 1 - p)$ random variable. The number of transmissions required for all M clients to successfully decode a block of N packets is the random variable

$$T_{nonsys} = \max_{j \in \{1,2,\cdots,M\}} y_j \tag{2.42}$$

The mean number of transmissions is then given by

$$
\begin{aligned}
E[T_{nonsys}] &= N + \delta + \sum_{t=N+\delta}^{\infty}\left(1 - (1-p)^{(N+\delta)M}\left(\sum_{x=N+\delta}^{t}\binom{x-1}{N+\delta-1}p^{x-N-\delta}\right)^{M}\right) \\
&= N + \delta + \sum_{t=N+\delta}^{t'}\left(1 - (1-p)^{(N+\delta)M}\left(\sum_{x=N+\delta}^{t}\binom{x-1}{N+\delta-1}p^{x-N-\delta}\right)^{M}\right) + \mathcal{O}(1)
\end{aligned}
\tag{2.43}
$$

for some large t' and where $\mathcal{O}(1) \to 0$ as $t' \to \infty$. Unfortunately, we cannot analytically derive closed-form expressions for $E[T_{nonsys}]$ due to the series not being any standard known series. It is, however,

easy to see that the series is summable and hence we use the summation of a finite number of terms to t' as an approximation. Unless otherwise stated, $t' = 20000$ for all our numerical results.

The corresponding number of higher-layer ACK transmissions is $A_{nonsys} = M$.

2.4.5 Mean #Transmissions with Systematic Fountain Code

When a systematic fountain code is used, the first N transmissions are uncoded packets while subsequent transmissions are coded. Let r_j denote the number of uncoded packets successfully received by client j, out of the N uncoded packets transmitted. If r_j is less than N, an additional $N + \delta - r_j$ coded packets must be received for the client to decode the block. Let y_j denote the number of transmissions required for client j to successfully receive r_j uncoded packets and $N + \delta - r_j$ coded packets. We notice that y_j is a Negative Binomial$(N + \delta, 1 - p)$ random variable. It follows that the total number of transmissions required for all clients to recover a block of N packets is the random variable

$$T_{sys} = \begin{cases} N, & (1 - p)^{MN} \\ \max_{j \in \{1, \cdots, M\}} y_j, & 1 - (1 - p)^{MN} \end{cases} \tag{2.44}$$

The mean number of transmissions is thus given by

$$E[T_{sys}] = (1 - p)^{MN} \cdot N + \left(1 - (1 - p)^{MN}\right) \cdot E[\max_{j \in \{1, \cdots, M\}} y_j] \tag{2.45}$$

where $E[\max_{j \in \{1, \cdots, M\}} y_j]$ is the mean number of transmissions when using the non-systematic fountain code, given by expression (2.43).

The number of higher-layer ACK transmissions is $A_{sys} = M$.

2.4.6 Goodput

Goodput here is defined as the mean number of application layer information bits correctly received by each client per second. For 802.11 WLANs goodput can be be expressed as

$$G = \frac{N \cdot L}{E[T] \cdot (D_{data} + D_{cw} + D_{DIFS}) + E[A] \cdot (D_{ack} + D_{SIFS})} \tag{2.46}$$

where $E[T]$ is the mean number of transmissions required for all clients to receive a block of N packets, given by (2.38), (2.43) and (2.45); $E[A]$ is the corresponding mean number of higher-layer

D_{SIFS} (μs)	16	hdr_{APP} (bytes)	3
D_{DIFS} (μs)	34	hdr_{UDP} (bytes)	8
σ (μs)	9	hdr_{IP} (bytes)	20
CW_{min}	15	hdr_{MAC} (bytes)	24
D_{PHYhdr} (μs)	20	D_{symbol}(μs)	4

Table 2.4: 802.11a/g protocol parameters used in the simulations

acknowledgement packets; L is the application layer payload size, which is taken to be 992 bytes in the following numerical results; D_{DIFS} and D_{SIFS} are respectively the durations of DIFS and SIFS. D_{cw} is the mean of the 802.11 contention window countdown duration. When there is a single multicast sender and no collisions between higher-layer acknowledgements and data packets, the random backoff number is uniformly selected between 0 and CW_{min}. The value of D_{cw} is hence $\sigma CW_{min}/2$ where σ is the duration of a PHY idle slot. If desired, a more sophisticated calculation of D_{cw} could also be carried out using modelling approaches such as that in [46]. D_{data} is the transmission duration of a data packet. For a UDP packet over 802.11a/g this is given by

$$D_{data} = D_{PHYhdr} + D_{symbol} \cdot \lceil \frac{(L + hdr_{APP} + hdr_{UDP} + hdr_{IP} + hdr_{MAC}) \times 8 + 22}{DBPS} \rceil \quad (2.47)$$

where hdr_{APP}, hdr_{UDP}, hdr_{IP} and hdr_{MAC} are respectively the length of the APP, UDP, IP and MAC headers in bytes; N_{DBPS} is the number of data bits per symbol, which is dependent of the PHY transmission rate (see Table 1.2); D_{symbol} is the OFDM symbol duration; D_{PHYhdr} is the duration of the PHY header, including the PLCP preamble and PLCP header but excluding the 16bit SERVICE field [30]. Similarly, D_{ack} is the transmission duration of a higher-layer ACK/NACK packet. In the following we assume that the acknowledgement packet has a 8 byte payload, and is transmitted at the basic PHY rate of 6Mbps. The values of protocol parameters in 802.11a/g are listed in Table 2.4.

Fig. 2.3 shows typical behaviors of goodput vs PER for a fixed 802.11a/g PHY rate $R = 6$Mbps and $M = 10$ clients. As we use an upper bound on the packet error rate given by Eqn. 2.3, the goodput curves shown in Fig. 2.3 are hence lower bounds. We use a fountain code block of size $N = 50$ and assume an overhead $\delta = 2$ for both systematic and non-systematic fountain codes (this is a lower bound on the decoding overhead, see [59]) - unless otherwise stated we use the same values for N and δ in the remainder of numerical results. At low PERs, uncoded and systematic fountain coded traffic have similar goodputs and this is higher than the goodput with a non-systematic fountain code owing to the decoding overhead δ. At higher PERs, both the systematic and non-systematic fountain codes exhibit similar performances, with the achieved goodput substantially higher than the uncoded case.

31
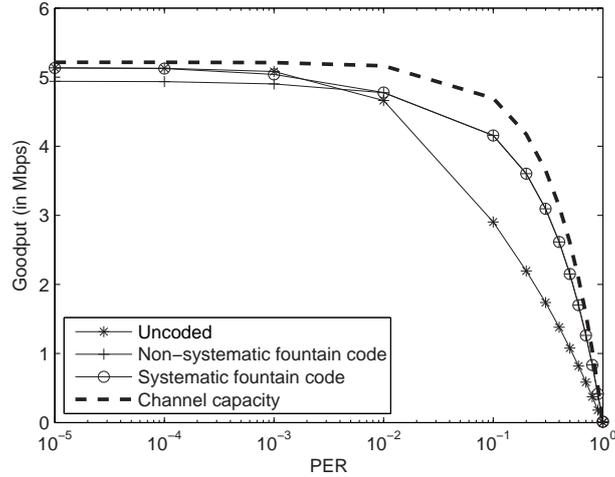
Figure 2.3: Goodput vs packet error rate (PER) for the 802.11a/g PHY rate 6Mbps. $M = 10$ client stations, block size $N = 50$ packets, coding overhead $\delta = 2$ packets.

The channel capacity curve is obtained by plotting $R(1 - PER)$. The gap between the capacity curve and the goodput curves is due to the protocol overheads included in each transmitted packet.

2.5 Maximising Goodput: Optimal PHY Modulation/FEC Rate

By combining the analysis in Section 2.4 (which gives goodput as a function of PER) with the packet erasure model described in Section 2.3 (which yields PER as a function of SNR and PHY rate) we can obtain goodput as a function of SNR and PHY rate. We can then determine (by numerical search) the optimal choice of PHY rate that maximises goodput for a range of channel SNRs and for both AWGN and Rayleigh channels. Fig. 2.4 shows a plot, obtained using this approach, of maximum goodput and optimal choice of PHY rate versus channel SNR for $M = 10$ client stations and a block of $N = 50$ packets. A first observation is that fountain coding yields uniformly higher goodputs at all SNRs. At some SNRs the increase is considerable, while at other SNRs it is minor. We observe this consistently in all of our results, even with Viterbi SDD. We also observe that at those SNRs where the difference in goodput is significant, the PER at optimal PHY rates with fountain coding is around

32

$10 - 30\%$ and higher than that without fountain coding. At other SNRs however, the PER value is very low (close to 0), and therefore there is only a small difference in goodput.

Our primary interest here, however, is in the impact of fountain coding on the optimal choice of PHY modulation/coding rate. It can be seen from Fig. 2.4 that the use of fountain coding allows the PHY rate to be stepped up slightly sooner as SNR rises. However, the SNR interval over which fountain coding yields an advantage is rather narrow (less than 1 dB wide), and hence the performance loss of operating fountain coding with uncoded optimal PHY rate choice is rather minor. This is perhaps unsurprising with only 10 client stations, since there may be insufficient diversity. Fig. 2.5 show the corresponding results for $M = 100$ client stations. This is, of course, a very large number of clients in the context of 802.11 WLANs. It can be seen that even with this larger number of clients the difference in goodput is still quite small, although the difference in the optimal PHY rate between fountain coded and uncoded traffic is more pronounced, *e.g.* the SNR range where fountain coding allows a higher PHY rate used is around 4dB wide in a Rayleigh channel. However, this is quite an extreme regime and for smaller numbers of client stations (up to around 50 clients) the SNR interval continues to be only about 1dB even over a Rayleigh channel.

In summary, we find that fountain coding uniformly increases network goodput which is as expected. The optimal PHY rate that maximises goodput with fountain coding generally differs from the optimal PHY rate for uncoded traffic, which is again as expected. Interestingly, however, the difference in goodput is extremely small over a wide range of network conditions, and hence the goodput loss associated with operating fountain coding at uncoded optimal PHY rates is minor. Therefore the optimal PHY rate for uncoded traffic is also close to optimal for fountain-coded traffic.

2.6 Minimising Energy: Joint PHY Power and Modulation/FEC Rate Optimization

We now extend consideration to the optimal choice of PHY power and rate to minimise energy expenditure for a given goodput. This is a key issue in the context of battery powered mobile devices, and in sensor networks.

Let E_b/N_0 denote the SNR per bit. We are interested in the choice of PHY rate which minimises E_b/N_0 while achieving a target goodput. For a given choice of PHY rate, the analysis in Section 2.4
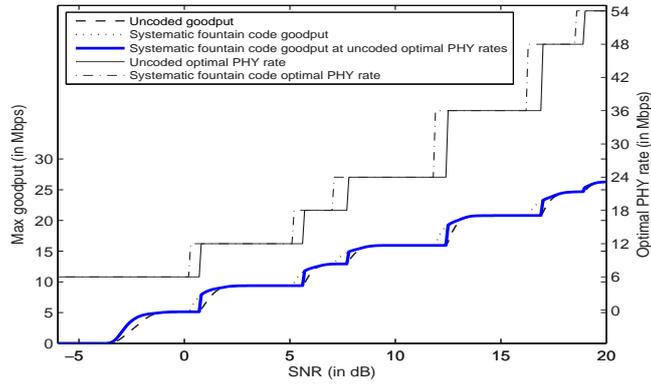
(a) AWGN, HDD
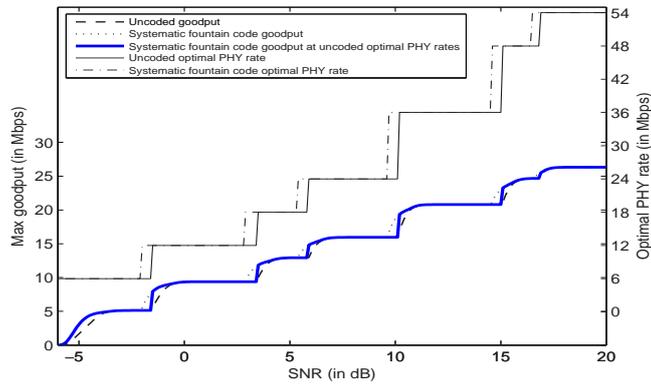
(b) AWGN, SDD
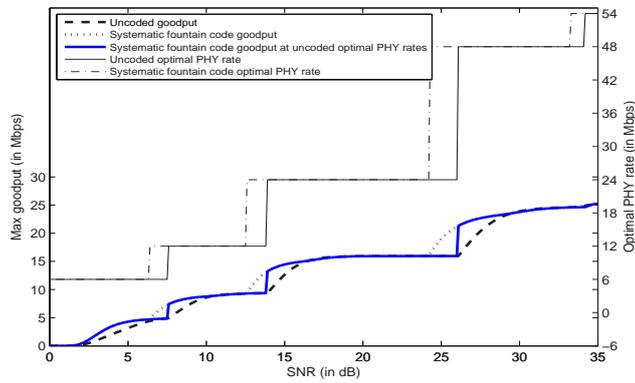
(c) Rayleigh channel, HDD

Figure 2.4: Optimal PHY rate maximising goodput vs SNR, $M = 10$ client stations, block size $N = 50$ packets. The three goodput curves at the bottom correspond to the left y-axis, and the two PHY rate curves at the top correspond to the right y-axis.

34
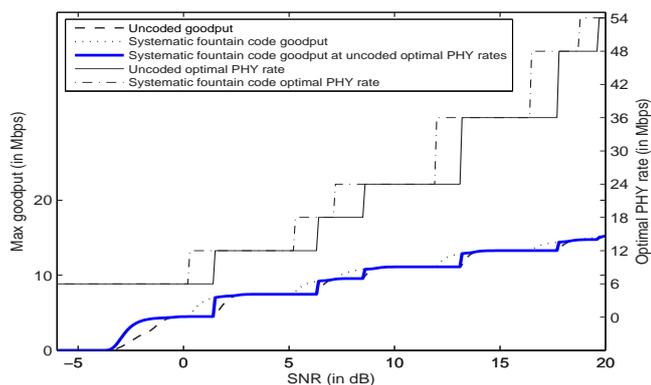
(a) AWGN, HDD
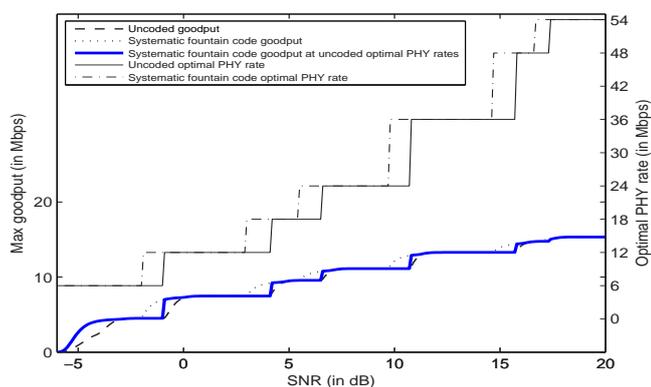
(b) AWGN, SDD
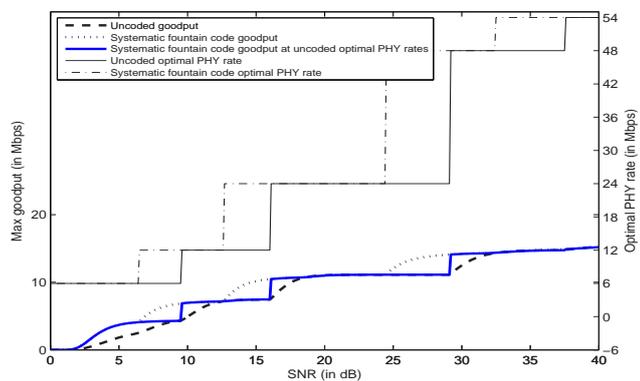
(c) Rayleigh channel, HDD

Figure 2.5: Optimal PHY rate maximising goodput vs SNR, $M = 100$ client stations, block size $N = 50$ packets. The three goodput curves at the bottom correspond to the left y-axis, and the two PHY rate curves at the top correspond to the right y-axis.

relates PER to goodput. We can invert this relationship to obtain the maximum permissible PER for a given PHY rate and target goodput, and then use the channel models described in Section 2.3 relating PER to SNR to determine the corresponding minimum SNR per bit E_b/N_0. Given a target goodput value, the minimum E_b/N_0 and the corresponding optimal PHY rate at which the minimum E_b/N_0 is achieved can be obtained by numerical search amongst the PHY rate set supported by 802.11 a/g.

This yields Fig. 2.6. It can be seen that fountain coding uniformly reduces the energy required to achieve a target goodput, however, even with such a large number $M = 100$ of clients, the optimal choice of PHY rate is almost identical for coded and uncoded traffic. Therefore, we find that the optimal PHY rates minimising energy usage for coded/uncoded traffic are almost the same, so that the energy loss of operating fountain coding using uncoded PHY rate choices is extremely minor.

2.7   Discussion

The joint performance of a fountain code concatenated with a PHY layer code has previously been considered in [40, 39] in the context of 3G cellular networks, and more recently in [7, 55, 14] in a general wireless context. In both cases it is concluded that when fountain codes are used at higher-layers then the overall system performance is improved if the PHY layer rate is selected to operate at much higher packet error rate (a $20-30\%$ loss rate in [39]) than would normally be used for uncoded traffic. That is, a cross-layer approach, whereby the PHY layer rate control exploits knowledge of the use of fountain coding at higher layers, can yield significant performance gains. We find in the context of the PHY rate choices available in 802.11a/g WLANs that a higher PHY rate associated with a higher PER $(10-30\%)$ is selected at some conditions, *i.e.* at some SNR values, as can be seen in Fig. 2.4 and Fig. 2.5, or at some target goodput values, as can be seen in Fig. 2.6. However, we also find that it holds only at a small number of SNRs and target goodputs, and thus offers limited benefits. The main reason for this is that the PHY rate-set choice in 802.11a/g is not rich enough that one can always operate close to a $10-30\%$ packet error rate. Therefore over a wide range of network conditions the optimal choice of PHY rate is similar for both fountain coded and uncoded traffic. Moreover this is true not only when the objective is to maximise goodput but also when we seek to minimise energy expenditure.

The paper [39] considers a similar setup to that considered here, but focuses specifically on Raptor
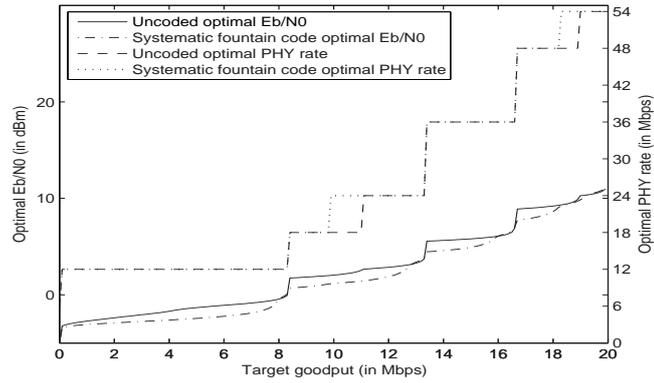
(a) AWGN, HDD
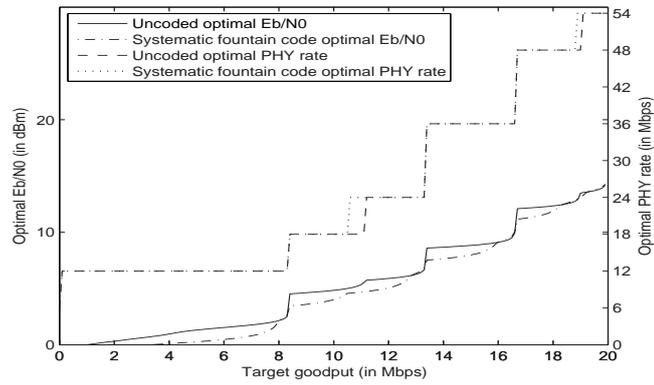
(b) AWGN, SDD
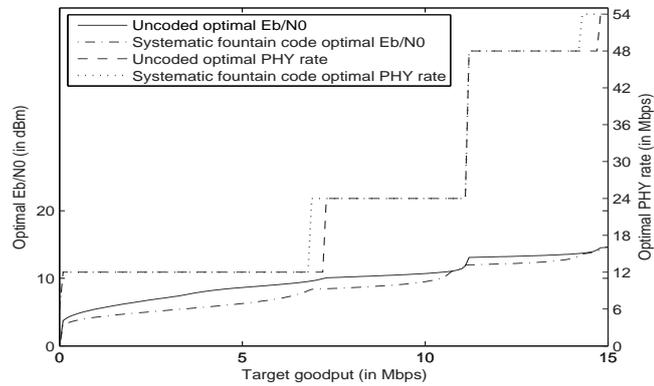
(c) Rayleigh channel, HDD

Figure 2.6: Optimal PHY rate/modulation minimising energy given target goodputs, $M = 100$ clients, block size $N = 50$. The three E_b/N_0 curves at the bottom correspond to the left y-axis, and the two PHY rate curves at the top correspond to the right y-axis.

codes (a particular systematic fountain code) over a UMTS MBMS service that uses QPSK modulation and turbo-coded FEC concatenated with a physical layer FEC. The performance objective is to minimise energy rather than maximise goodput. The authors in [39] observe that an optimum level of PHY level FEC exists, however this is intimately related to the FEC options supported by the PHY layer. Additionally, they only consider scenarios with 500 client stations. This is far higher than the number of clients likely to be present in an 802.11 WLAN. We have already noted in our own analysis that the impact of coding becomes more pronounced as the number of client stations increases. This suggests that it might be interesting to repeat the analysis in [39] for smaller numbers of client stations to evaluate how this changes the cross-layer trade-off.

We have assumed that all client stations have the same PER at every PHY rate. This is not the case in a wireless network where clients are in different locations owing to different path loss, shadow fading and interference effects. However, this assumption has little effect on our conclusions. To illustrate this, consider a heterogenous multicast network with two classes of client stations, with M_1 stations in class 1 and M_2 in class 2. All stations in the same class have the same SNR. Class 1 is subjected to noisy reception ($PER \geq 0$); while class 2 experiences reliable reception at any of the available PHY rates ($PER = 0$). Fig. 2.7 shows the maximum goodput and optimal PHY rate versus SNR of class 1 stations for a network with $M_1 = 5$ class 1 stations and $M_2 = 5$ class 2 stations and an AWGN channel. It can be seen that even though the PER is different among client stations, the conclusion remains the same. The reason is that as our objective is for each individual client station to receive all N packets in each block, the number of transmissions is determined by the station(s) with the lowest PER. When the worst client(s) receive(s) the whole block, all other clients with higher SNRs have already received everything. The homogenous setting is therefore the worst-case setting. Even though we may quantitatively get a different maximum goodput value, qualitatively the behaviour will be the same as in the heterogenous case. If the objective changes, *e.g.* to ensure only 95% clients decode successfully, we just need to reduce the number of decoding overheads, hence the conclusion will still hold.

For the setting of this chapter we have assumed a Rayleigh flat fast-fading channel. It is known that multipath-fading will lead to a diversity gain of the order of the number of distinguishable multipath components. Flat fading is the worst-case setting for our problem, in that it will yield the greatest difference between fountain coded and uncoded performances. For purpose of illustration results for HDD Nakagami-6 fast-fading model are shown in Fig. 2.8. Similar behaviours to the AWGN and Rayleigh channels are observed. This indicates that our conclusion is insensitive to the channel model
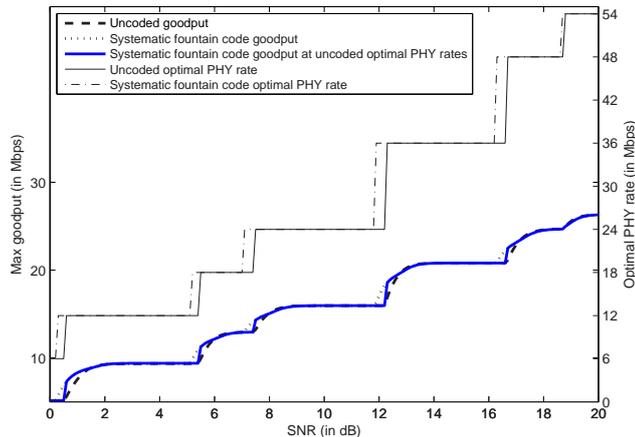
Figure 2.7: Optimal PHY rate maximising goodput vs SNR of class 1 stations, with $M_1 = 5$ class 1 stations and $M_2 = 5$ class 2 stations, block size $N = 50$, AWGN channel. The three goodput curves at the bottom correspond to the left y-axis, and the two PHY rate curves at the top correspond to the right y-axis.

considered.

2.8 Conclusions

In this chapter we consider the joint performance of fountain coding and 802.11a/g PHY modulation/coding. We consider optimality both in terms of maximising goodput and minimising energy, and results are presented for a wide range of channel models. To our knowledge this is the first detailed study on the optimal choices of PHY rate to use with and without fountain codes in standard 802.11a/g WLANs. In contrast to studies in cellular networks, we find that in 802.11a/g WLANs the cross-layer approach of a higher-layer fountain coding with a PHY layer modulation and FEC coding can yield very limited gains, and the PHY rate that maximises uncoded goodput or minimises uncoded energy is also close to optimal for fountain-coded traffic over a wide range of network conditions, principally because of the limited PHY rate-set choice available in 802.11a/g WLANs. This is potentially an important observation as it indicates that in the context of 802.11a/g WLANs the cross-layer rate control for higher-layer fountain coding with physical layer modulation and FEC coding would bring few benefits, and PHY layer rate control can be carried out without regard to the use of fountain
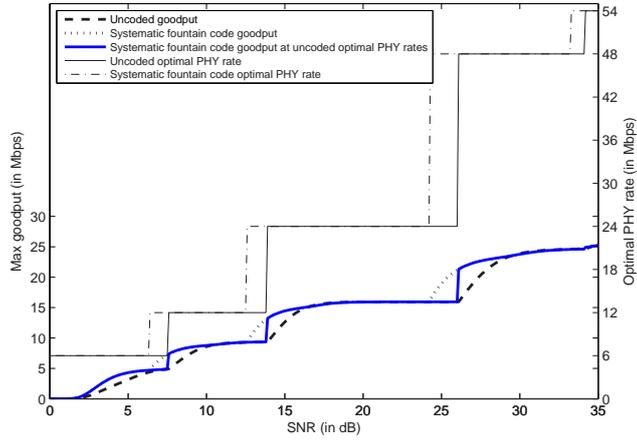
Figure 2.8: Optimal PHY rate maximising goodput vs SNR, $M = 10$ client stations, block size $N = 50$ packets, Nakagami-6 fast-fading channel. The three goodput curves at the bottom correspond to the left y-axis, and the two PHY rate curves at the top correspond to the right y-axis.

coding at higher layers.

Chapter 3

Outdoor 802.11 Frames Provide a Hybrid Binary Symmetric/Packet Erasure Channel

3.1   Introduction

Frames sent over an 802.11 wireless link may be received (i) with a *PHY error*, where the PHY header is corrupted by noise/interference and the receiver cannot demodulate the frame, or (ii) with a *CRC error*, where the PHY header is received correctly and the frame is decoded but then fails a CRC check, or (iii) *without error*. The resulting information channel is often modelled as a packet erasure channel. That is, frames are received in an "all or nothing" fashion with frames having PHY or CRC errors being discarded and only frames received without error retained. However, the fraction of incorrect bits in frames with CRC errors can be small. For example, Fig. 3.1 shows experimental measurements of the fraction of corrupted bits in frames with CRC errors on an 802.11 link (see Section 3.2.1 for details of the experimental setup). It can be seen that even when the frame error rate (FER) is high (91.98% of frames fail the PHY header check or the CRC check), most of the frames received with CRC errors have less than 10% of bits incorrect. Thus, these corrupted frames potentially provide a useful channel through which we can transmit information.
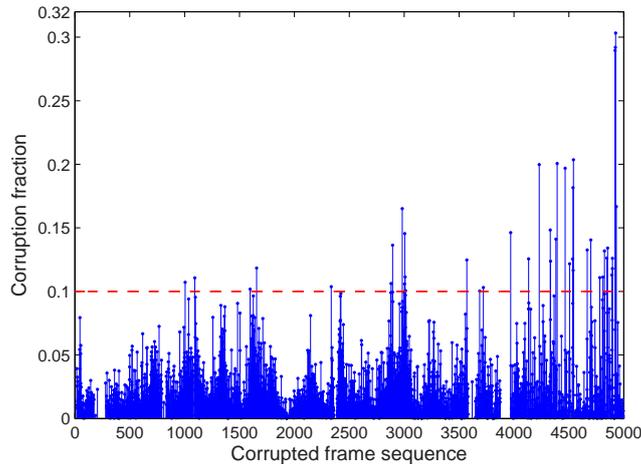
Figure 3.1: Measured fraction of incorrect bits vs frame sequence number. Measurements taken outdoors in an open space with onmidirectional antennas at transmitter and receiver, 802.11g PHY rate 54Mbps. $FER = 91.98\%$.

Taking this observation as our starting point, in this chapter our aim is to characterise the information channel provided by 802.11 frame transmissions. Using measurements taken in an outdoor environment, we demonstrate that the channel provided by corrupted frames alone (*i.e.* ignoring frames with PHY errors and frames received without error) can be accurately modelled as a binary symmetric channel (BSC) provided that appropriate pre- and post- processing is carried out. Also, the channel provided by corrupted frames and other frames combined can be accurately modelled as a hybrid binary symmetric/packet erasure channel. We calculate the capacity of this hybrid channel and show that the potential exists for significant throughput gains.

3.2 Preliminaries

3.2.1 Experimental Setup

Experimental data was collected in an open outdoor space with no other interferers present. The FER was adjusted by varying the distance between sender and receiver. Care was taken to ensure repeatability of results – measurements were taken on an open space (a large playing field), sender and receiver were positioned at fixed heights, antenna orientations were held fixed, human operators

left the vicinity of the experiment during measurements.

Hardware and software

An Asus Eee PC 4G Surf equipped with Atheros AR5BXB63 802.11b/g chipsets (AR2425, MAC 14.2, RF5424, PHY 7.0) was used as the access point, running FreeBSD 8.0 with the RELEASE kernel and using the standard FreeBSD ATH driver. A Fujitsu E series Lifebook equipped with a Netgear dual band 802.11a/b/g wireless PC card WAG511 using Atheros AR5212 chipset was used as a client station, running Ubuntu 11.04 with the RELEASE kernel and using a modified Linux Madwifi driver.

We disabled the Atheros' Ambient Noise Immunity feature which has been reported to cause unwanted side effects [60]. Transmission power of the laptops was fixed and antenna diversity disabled. In previous work we have taken considerable care to confirm that with this hardware/software setup the wireless stations accurately follow the IEEE 802.11 standard, see [20, 24, 60] for further details.

A perl script was used to generate CBR UDP traffic. The content of each UDP packet was a random binary vector. Unless stated otherwise, the UDP payload is 8000 bits, and the inter-packet interval is 20 ms. Packets were transmitted over the WLAN in the broadcast mode and hence there were no MAC level ACKs or retransmissions. The wireless driver at the receiver was modified to record the receiving status for frames with PHY errors, CRC errors or without errors and to transfer the contents of these from the kernel to user space via the high-speed data relay filesystem (relayfs) [3].

Recovering sequence number for corrupted frames

Frames received with CRC errors were compared against the corresponding original content in order to determine which specific bits inside the frame were received corrupted. Since the frame header might also be corrupted, the following fitting and pattern matching procedure was used to indirectly recover the frame sequence number. For each corrupted frame we first searched the set of correctly received frames to find the packet received without error closest in time to the corrupted frame. Since the frames are transmitted at a fixed rate, the interval between two consecutive frame timestamps is roughly constant (about 20ms), but to correct for clock skew between transmitter and receiver it is necessary to estimate the relative clock rate and offset. We estimated these using linear least squares fitting [1] of the received timestamps of neighbouring error-free frames. The timestamp of the

corrupted frame was then used to determine likely candidates for this frame amongst the transmitted frames. From this set of candidates, the sequence number and payload of each were compared with those of the corrupted frame in order to identify the transmitted frame most likely to correspond to the received corrupted frame.

3.2.2 Runs Test

Our statistical analysis makes use of the *runs test* (also called the Wald-Wolfowitz test) [49, 63]. The runs test is a non-parametric test to check the null hypothesis that the elements in a two-valued sequence are independent and identically distributed. Given a 0-1 sequence, a *run* is consecutive entries having the same value *e.g.* in the sequence 1100110111 there is one 0 run, one 00 run, two 11 runs and one 111 run. Under the null hypothesis, the number of runs is a random variable whose conditional distribution is approximately normal with mean $\mu = \frac{2\ N_1\ N_0}{N} + 1$ and variance $\sigma^2 = \frac{(\mu-1)(\mu-2)}{N-1}$ where N_1 and N_0 are respectively the number of 1's and 0's in the sequence, and $N = N_1 + N_0$. Unless otherwise stated we carry out statistical testing at the 5% significance level.

3.3 Channel Modelling

We proceed by first investigating the channel provided by corrupted frames alone *i.e.* ignoring frames with PHY errors and frames received correctly. We find that, to within statistical error, this can be accurately modelled as a BSC. We then consider the channel provided by corrupted frames and other frames combined. We find that this can be accurately modelled as a hybrid binary symmetric/packet erasure channel.

3.3.1 Channel Provided by Corrupted Frames

A binary symmetric channel (BSC) [15], as illustrated in Fig. 3.2, takes binary input $X \in \{0, 1\}$ and maps this to binary output $Y \in \{0, 1\}$. With probability $1 - p$ the channel transmits the input bit correctly and with crossover probability p the input bit is flipped. That is, $\mathbb{P}\{Y = 1 | X = 1\} = 1 - p = \mathbb{P}\{Y = 0 | X = 0\}$ and $\mathbb{P}\{Y = 0 | X = 1\} = p = \mathbb{P}\{Y = 1 | X = 0\}$. Repeated binary channel uses are independent and identically distributed. That is, if a binary vector is transmitted through a BSC,
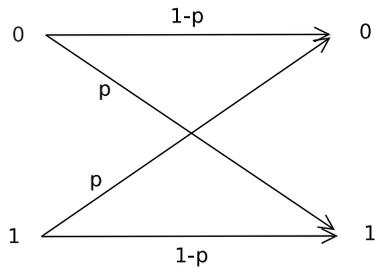
Figure 3.2: Binary symmetric channel

each bit is flipped independently and with identical crossover probability p. Therefore, to establish that a channel taking binary inputs is a BSC, we need to show

(1) Repeated binary channel uses are independent and identically distributed (i.i.d.).

(2) The probability that a 1 is flipped to a 0 after transmission is the same as the probability that a 0 is flipped to a 1 *i.e.* the binary channel is symmetric.

Binary channel uses are i.i.d.

We begin by presenting raw experimental measurement data in Fig. 3.3(a). This figure plots representative measurements of the bit error frequency for each bit position within a corrupted frame. It can be seen that bit errors are not evenly distributed and the bit error frequency sequence is periodic across a frame. This observation is not new, *e.g.* see [23, 18, 66], and clearly violates the independence requirement of a BSC. Nevertheless, Fig. 3.3(b) plots the bit error frequency for the same data after interleaving (randomly permuting the bits in each frame). Interleaving can be readily implemented at the MAC layer – bits in a frame are permuted at the transmitter before a MAC frame goes down to the PHY layer and when the frame is received the inverse permutation is used to recover the original bit order. It can be seen from Fig. 3.3(b) that after interleaving the periodicity of bit errors appears to be removed, although further analysis is required to confirm this.

To analyse the independence of repeated channel uses within each individual interleaved frame we use the runs test. For each corrupted frame we construct a 0-1 sequence by labelling corrupted bits as 1's and correct bits as 0's. We find that the runs test cannot reject the null hypothesis at the 5% significance level that after interleaving bit errors inside a frame are independent. That is, to
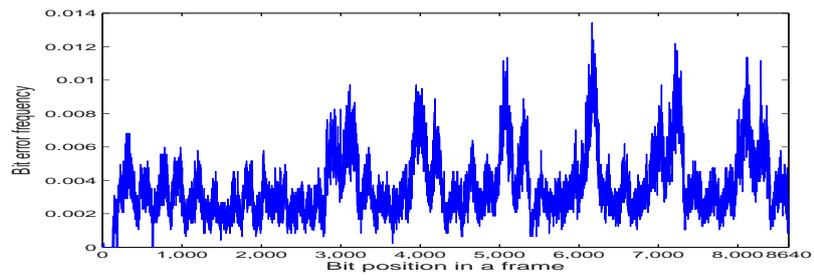
(a) Before interleaving
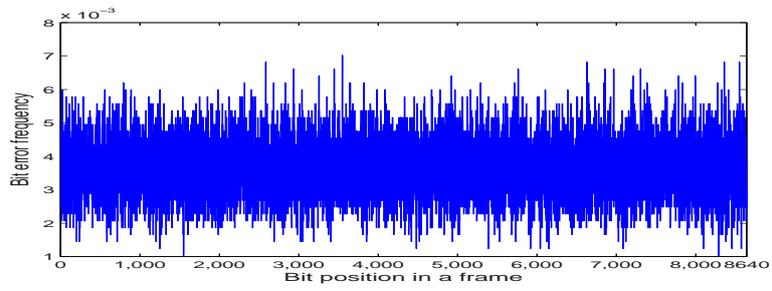
(b) After interleaving

Figure 3.3: Per-bit error frequency pattern across a frame before and after interleaving, outdoor, $FER = 0.5658$, PHY rate 54Mbps, 5000 frames.

within statistical error we can conclude that repeated channel uses inside each interleaved frame are independent and Bernoulli distributed with the bit crossover probability $p = n_c/l$, where n_c is the number of corrupted bits within a frame and l is the frame length.

Next, to analyse the independence of channel uses across multiple frames we concatenate the foregoing binary sequences for successive corrupted frames and apply the runs test. For example, Fig. 3.4 plots the measured per-frame bit crossover probability for a sequence of 10,000 frames. Consecutive corrupted frames that pass the runs test are labelled using the same marker. It can be seen that the experimental run is partitioned into three segments. Within each segment, corrupted frames form a bit-level channel over which repeated channel uses can be considered to be independent and Bernoulli distributed to within statistical error. The first segment spans 3865 frames (around 77.3s given that the inter-packet interval is 20ms) and the third segment spans 6118 frames (about 122.36s). The second segment has only one frame with an unusually high crossover probability. Neglecting this single bad frame out of 10,000 frames, the measurement results indicate that for long periods the sequence of corrupted frames form a bit-level channel over which repeated channel uses can be considered to be i.i.d. The data in Fig. 3.4 is for a FER of 0.0423 and PHY rate of 54Mbps. Fig. 3.5 plots the mean of segment durations over an experimental run of 10000 packets for a range of FERs and PHY rates. It can be seen that the segment duration tends to decrease as the FER increases *i.e.* the duration within which corrupted frames have an i.i.d. bit crossover probability becomes shorter as the FER increases. For FER's less than 20%, at all PHY rates the mean segment duration exceeds 10s. That is, for FER's less than 20% the binary channel provided by corrupted frames is i.i.d. for periods exceeding, on average, 10s. Such time-scales seem sufficient for most channel modelling purposes.

Binary channel is symmetric

Table 3.1 reports the measured bit flip rates for 1 values and 0 values. Results are shown for a range of FER values and PHY rates. It can be seen that the bit flip probabilities for both 1's and 0's are close to each other , and thus, to within experimental error, can be approximately considered as "symmetric".

In summary, after interleaving the bit errors in corrupted frames are, to within statistical error, independent and identically distributed with a symmetric crossover probability, *i.e.* the information channel provided by these corrupted frames can be accurately modelled as a BSC.
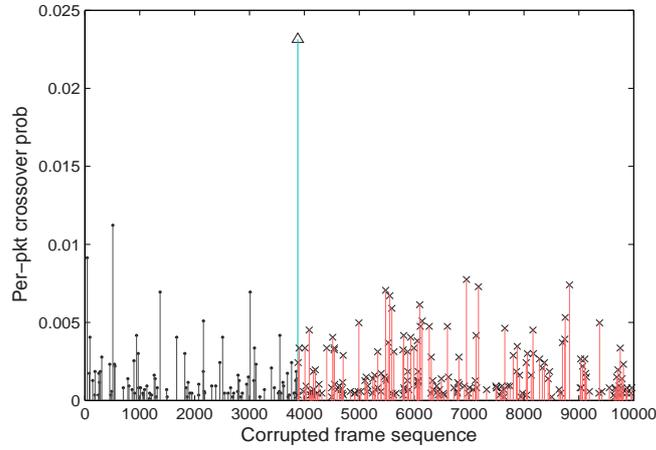
Figure 3.4: Per-frame bit crossover probability for a sequence of 10000 packets, PHY rate 54Mbps, $FER = 0.0423$.
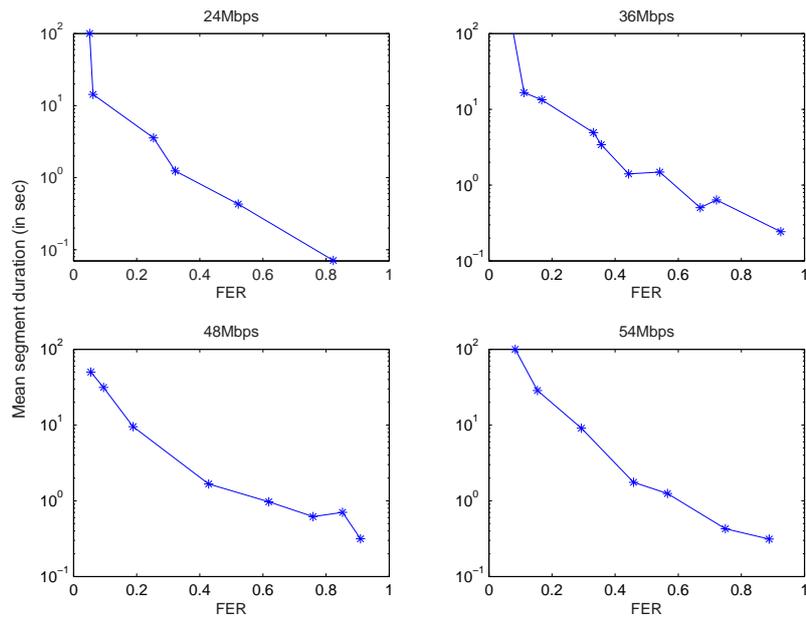
Figure 3.5: Mean segment duration versus FER at different PHY rates.

PHY rate	FER	Flip rate for 1's		Flip rate for 0's		PHY rate	FER	Flip rate for 1's		Flip rate for 0's	
		μ_1	$\sigma_1/\sqrt{N_1}$	μ_0	$\sigma_0/\sqrt{N_0}$			μ_1	$\sigma_1/\sqrt{N_1}$	μ_0	$\sigma_0/\sqrt{N_0}$
54Mbps	0.0835	0.0018	2.34×10^{-5}	0.0018	2.26×10^{-5}	36Mbps	0.0649	0.0014	2.72×10^{-5}	0.0013	2.48×10^{-5}
	0.0984	0.0019	2.40×10^{-5}	0.0020	2.34×10^{-5}		0.1673	0.0019	1.69×10^{-5}	0.0020	1.65×10^{-5}
	0.1540	0.0023	1.97×10^{-5}	0.0023	1.88×10^{-5}		0.3314	0.0023	1.31×10^{-5}	0.0023	1.22×10^{-5}
	0.2932	0.0031	1.67×10^{-5}	0.0032	1.62×10^{-5}		0.4427	0.0032	1.34×10^{-5}	0.0033	1.30×10^{-5}
	0.4584	0.0042	2.19×10^{-5}	0.0041	2.04×10^{-5}		0.5416	0.0037	1.19×10^{-5}	0.0035	1.10×10^{-5}
	0.5658	0.0067	2.45×10^{-5}	0.0065	2.31×10^{-5}		0.6697	0.0052	1.43×10^{-5}	0.0054	1.38×10^{-5}
	0.7492	0.0066	1.52×10^{-5}	0.0068	1.48×10^{-5}		0.7214	0.0052	2.27×10^{-5}	0.0048	2.08×10^{-5}
	0.8881	0.0120	1.92×10^{-5}	0.0121	1.83×10^{-5}		0.9247	0.0122	3.41×10^{-5}	0.0115	3.15×10^{-5}
48Mbps	0.0547	0.0015	3.21×10^{-5}	0.0017	3.16×10^{-5}	24Mbps	0.0617	0.0048	7.66×10^{-5}	0.0044	6.99×10^{-5}
	0.0949	0.0021	2.91×10^{-5}	0.0020	2.71×10^{-5}		0.3219	0.0041	1.84×10^{-5}	0.0039	1.73×10^{-5}
	0.1883	0.0021	1.68×10^{-5}	0.0020	1.59×10^{-5}		0.5220	0.0098	2.68×10^{-5}	0.0094	2.50×10^{-5}
	0.4319	0.0034	1.43×10^{-5}	0.0034	1.35×10^{-5}		0.8229	0.0306	3.19×10^{-5}	0.0293	2.97×10^{-5}
	0.5818	0.0036	1.81×10^{-5}	0.0035	1.70×10^{-5}	18Mbps	0.0449	0.0045	1.44×10^{-5}	0.0046	1.38×10^{-5}
	0.6185	0.0044	1.38×10^{-5}	0.0046	1.32×10^{-5}		0.3481	0.0049	2.15×10^{-5}	0.0050	2.07×10^{-5}
	0.7586	0.0045	1.76×10^{-5}	0.0043	1.64×10^{-5}		0.6415	0.0076	1.99×10^{-5}	0.0075	1.89×10^{-5}
	0.8522	0.0080	2.30×10^{-5}	0.0077	2.15×10^{-5}		0.8376	0.0091	1.93×10^{-5}	0.0091	1.84×10^{-5}
	0.9091	0.0146	2.09×10^{-5}	0.0145	1.98×10^{-5}		0.9776	0.0129	3.79×10^{-5}	0.0123	3.53×10^{-5}

Table 3.1: Bit flip rates for 1's and 0's, μ_i the mean flip rate of bit i, $\sigma_i/\sqrt{N_i}$ the standard deviation of flip rate of bit i, N_i the total number of bit i in corrupted frames, $N = N_0 + N_1$.
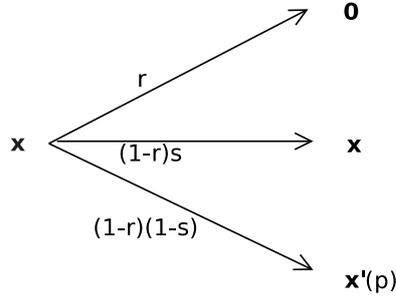
Figure 3.6: Hybrid BSC/packet erasure channel model.

3.3.2 Hybrid Binary Symmetric/Packet Erasure Channel

Our hypothesis is that the channel provided by 802.11 frames can be accurately modelled as a mixed packet erasure/binary symmetric channel. Formally, the channel takes an n-bit binary vector $\boldsymbol{x} \in \{0,1\}^n$ as input, and outputs received vector $\boldsymbol{y} \in \{0,1\}^n$. The vector $\boldsymbol{y}$ can be received with three possible states: (i) $\boldsymbol{0}$ (erased), (ii) $\boldsymbol{x}'$ (corrupted) and (iii) $\boldsymbol{x}$ (without error). Repeated channel uses are i.i.d. The probability that a received vector is erased is $\mathbb{P}\{\boldsymbol{y} = \boldsymbol{0}\} = r$. The probability that a non-erased vector is received without error is $\mathbb{P}\{\boldsymbol{y} = \boldsymbol{x} \mid \boldsymbol{y} \neq \boldsymbol{0}\} = s$ and so the probability that a vector is correctly received is $\mathbb{P}\{\boldsymbol{y} = \boldsymbol{x}\} = (1-r)s$. The probability that a vector is corrupted is $\mathbb{P}\{\boldsymbol{y} = \boldsymbol{x}'\} = (1-r)(1-s)$. In a corrupted vector bits are flipped symmetrically and independently with crossover probability p. This channel is illustrated schematically in Fig. 3.6.
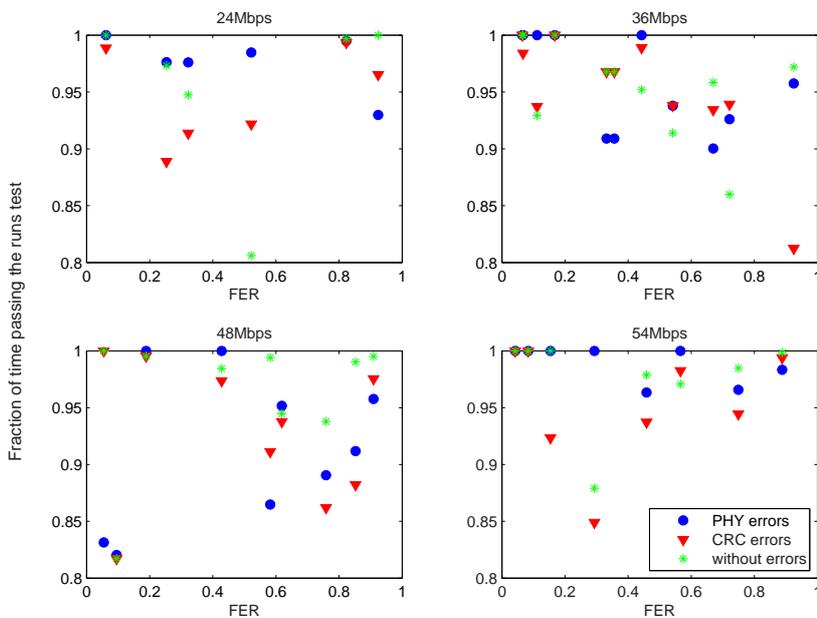
Figure 3.7: Fraction of time passing the runs test for PHY errors, CRC errors and good frames.

We can relate this hybrid model to 802.11 by associating the input vectors with transmitted frames, erasures with PHY errors and corrupted vectors with CRC errors. To establish an equivalence we need to show that (i) frames are received with PHY errors, CRC errors and without error in an i.i.d. fashion, (ii) corrupted frames provide a BSC. We have already established (ii) in Section 3.3.1 but it remains to establish (i).

Our hypothesis is that PHY errors, CRC errors and frames received without errors are mutually independent across time. To investigate this hypothesis, we again use the runs test. We construct a 0-1 sequence by labelling frames received with PHY errors as a 1 and other frames as a 0, and then apply the runs test for each individual segment which passes the runs test for bit error independence check. Similarly, we do the same respectively for frames received with CRC errors and without errors. Fig. 3.7 plots the fraction of time in an experimental run of 10,000 frames within which the runs tests pass for a range of FERs and PHY rates. It can be seen that the runs tests pass over at least 80% of the time in each experimental run, *i.e.* in an experimental run of 10,000 frames over 80% of the time, to within statistical error, frames are received with PHY errors, CRC errors and without error in an i.i.d. fashion.

3.4 Channel Capacity

The capacity of the hybrid channel is

$$C = R(1-r)\big(s + (1-s)\big(1 - H(p)\big)\big)$$

where $H(p)$ is the entropy function of a Bernoulli(p) random variable, *i.e.* $H(p) = -p\log_2(p) - (1 - p)\log_2(1-p)$ [15] and R is the PHY bit rate.

Using our experimental data, we binned frames according to their RSSI (Received Signal Strength Indication) and calculated the measured crossover probability in corrupted frames and the frame error rate. As we could not measure the RSSI for PHY erasures using our hardware, we do not include these here. Fig. 3.8 shows the resulting measured capacity vs RSSI. For comparison we also plot the experimentally measured packet erasure channel capacity *i.e.* the capacity when corrupted frames are discarded. It is important to note that these measured curves cannot be directly compared with theoretical calculations since the mapping between RSSI and SNR is not well defined. It can be seen that the capacity of the hybrid channel is strictly greater than the erasure channel capacity, as expected. More interestingly, it can be seen that the hybrid channel offers capacity increases of more than 100% over a wide range of RSSIs. This is a striking observation as it indicates that the potential exists for significant network throughput gains if the information contained in corrupted packets is exploited.

In addition to the hybrid channel capacity and the PEC capacity, we also plot a lower bound on the hybrid channel capacity. We assume that bit errors are i.i.d across all received frames *i.e.* both corrupted frames and error-free frames. Hence, each received frame can be simply considered as a binary vector in which an unknown subset of bits have been "flipped". The frame channel model can thus be simplified to a BSC. Similarly using our experimental data, frames are binned according to their RSSI, and the crossover probability across both corrupted frames and correct frames is $p^- = n_c/\big((N_g + N_c)l\big)$, in which n_c is the number of corrupted bits totaled over all corrupted frames with the same RSSI value; N_g is the number of frames received without error; N_c is the number of frames received with CRC errors, *i.e.* corrupted frames; l is the frame length, which is taken to be 8640 bits in our experiement. The channel capacity of this BSC is then

$$C^- = R\big(1 - H(p^-)\big)$$

It can be seen in Fig. 3.8 that the capacity difference between the BSC model and the hybrid channel
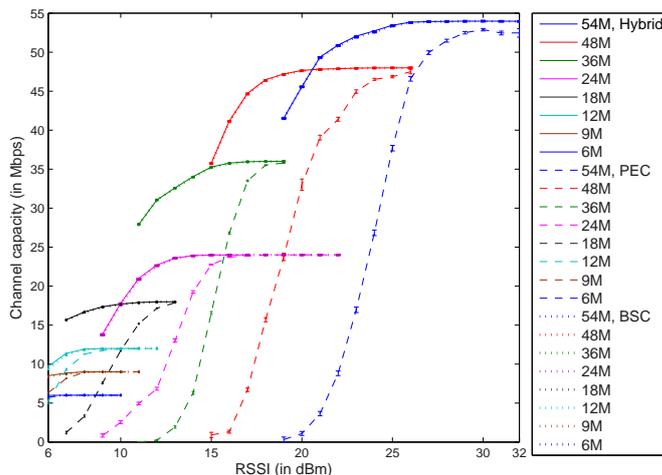
Figure 3.8: Outdoor experimental channel capacity

model is minor *i.e.* the simple approximate BSC model provides a tight lower bound on the hybrid channel capacity.

3.5 Conclusions

In this chapter, using measurements taken in an outdoor environment, we demonstrate that for 802.11 wireless links the channel provided by corrupted frames alone (i.e. ignoring frames with PHY errors and frames received without error) can be accurately modelled as a BSC provided that appropriate pre- and post- processing is carried out. Also, the channel provided by corrupted frames and other frames combined can be accurately modelled as a hybrid binary symmetric/packet erasure channel. Importantly, we find that this hybrid channel offers capacity increases of more than 100% compared to a conventional PEC over a wide range of RSSIs. This is a striking observation as it indicates that the potential exists for significant network throughput gains if the information contained in 802.11 corrupted packets is exploited. Since 802.11n includes the 802.11a/g modulation and coding schemes, our conclusions apply directly. We do, however, expect that indoor links will behave differently from outdoor links due to multipath effects and temporal variations in the environment, and this will be the subject of future work.

Chapter 4

Multi-destination Aggregation with Coding in a Binary Symmetric Channel of 802.11 WLANs

4.1   Introduction

In this chapter we consider the potential benefits of adopting a BSC paradigm in 802.11 WLANs, as opposed to a more conventional PEC paradigm. Chapter 3 establishes that for outdoor 802.11 wireless links the channel provided by frame transmissions can be modelled as a hybrid binary symmetric/packet erasure channel. Importantly, this hybrid channel offers capacity increases of more than 100% compared to a PEC over a wide range of RSSIs. By simply viewing a received frame as a binary vector with an unknown subset of bits being "flipped", the hybrid channel can be simplified to a BSC, and we have shown that the approximate BSC model provides a tight lower bound on the hybrid channel capacity. In this chapter we adopt a BSC paradigm for 802.11 frame transmissions and consider the potential performance gains for multi-destination aggregation in 802.11 WLANs.

This BSC paradigm dovetails with the trend towards greater decoupling of the unit of transmission (*i.e.* frames) used at the MAC/PHY layer from the unit of transmission (*i.e.* packets) used at the

IP layer. The latest 802.11n standard [31] supports PHY data rates up to a maximum of 600Mbps. Increasing PHY rates leads to faster transmission of the MAC frame payload, but overheads associated with each transmission such as PHY header and MAC contention time typically do not decrease at the same rate and thus begin to dominate the frame transmission time. To maintain throughput efficiency at high PHY rates, 802.11n proposes frame aggregation, a scheme of aggregating multiple short packets to form a single long transmission packet. In such a way, overheads associated with a single transmission are amortised across multiple packets, and consequently the ratio of the MAC frame payload transmission time over the overall frame transmission time is increased and thus higher MAC layer throughputs are delivered, *e.g.* see [37].

A logical extension is to consider aggregation of packets destined to different receivers into a single large packet. Such multi-destination aggregation is currently the subject of much interest [44, 61, 64, 34] because we can expect that often there simply may not be enough traffic to an individual destination to always allow large packets to be formed, and the network efficiency will quickly degrade when small packets are used. This scenario is particularly important for higher PHY rates and when short-packet applications like VoIP [61, 64, 34], email and web-browsing are considered. Also, due to the multi-user aspect of a WLAN, *i.e.* transmissions are inherently broadcast in nature, the multi-destination aggregated packet can be heard by all the intended receivers in an WLAN, which guarantees multi-destination aggregation is practically feasible.

One of the key issues in multi-destination aggregation is the choice of PHY transmission rate for the aggregated frame. Although the multi-destination aggregation allows simultaneous transmissions to multiple receivers, the channel quality between the transmitter and each receiver, *e.g.* the AP and each client station in a WLAN, is generally different, and thus the optimal PHY transmission rate which matches the channel quality of each receiver is also different. The current 802.11 standard constraints the bits transmitted in the same packet to be sent at the same PHY rate. The state of the art in 802.11 WLANs is to send multi-destination transmissions at the highest PHY rate which the receiver with the worst channel quality can support such that every receiver is capable of decoding the received packet; this is also the recommendation for multicast rate control in the current 802.11 standard [30]. Clearly this is inefficient. Our analysis in Section 4.6 shows the quantification of this inefficiency. The ideal way is to send each segment in an aggregated frame at an individual PHY rate which is the highest rate that each of the receivers supports. But this requires complex hardware modifications.

We propose a more practical approach which still uses a single PHY transmission rate for the entire aggregated frame , but encodes information destined to different receivers with different levels of protection by using higher-layer BSC-based coding schemes. Recent breakthroughs in efficient capacity-approaching error correction codes, such as LDPC codes [56] and superposition coding [8], make this practically achievable. In this manner bits within the same packet could be transmitted to different destinations at different information rates while still using the same PHY rate. We have demonstrated in Section 4.6 and Section 4.7 that with use of this approach for multi-destination aggregation, increases in network throughput of more than 100% are possible over a wide range of channel conditions. Moreover, since the coding is introduced above the MAC layer, there is no need for any hardware changes, the performance gains hence essentially come for "free".

Our contributions include:

- The first detailed theoretical analysis of the potential performance gains of adopting a BSC paradigm in 802.11 WLANs. This includes important 802.11 MAC features such as the framing overhead and the overhead of CSMA/CA contention and collisions.

- Consideration of the multi-user channel aspect of a WLAN in the context of the BSC paradigm, *i.e.* frame transmissions are inherently broadcast in nature and thus may be received by multiple stations. We explore this for both unicast and multicast multi-destination aggregations.

4.2   802.11a/g Rayleigh Binary Symmetric Channel Model

In Chapter 3 we take experimental measurements in an outdoor environment and model the channel provided by outdoor 802.11a/g links as a hybrid binary symmetric/packet erasure channel. The outdoor channel capacity performance is shown in Fig. 3.8 for both the PEC and the hybrid channel. We model the indoor fading environment as a Rayleigh channel. The performance analysis for PEC in an 802.11a/g Rayleigh channel is given in Section 2.3.2. In this section we derive the crossover probability of a BSC in an 802.11a/g Rayleigh channel when the Viterbi hard decision decoder is used. Since 802.11n uses very similar modulation and FEC (adding a small number of additional OFDM tones and a new 5/6 code rate [31]), our analysis carries over essentially unchanged for single antenna systems. We leave consideration of MIMO 802.11n systems as future work.

In the Viterbi hard decision decoding, an upper bound on the bit crossover probability is the sum of
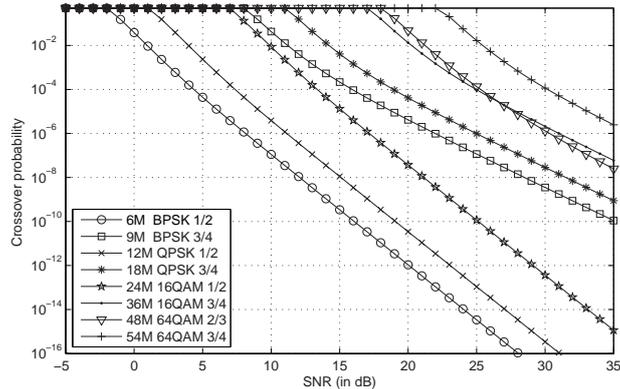
Figure 4.1: Crossover probability vs. SNR, Rayleigh channel

the expected number of erroneous bits for all possible incorrectly selected paths. For a binary-tree convolutional code, the expression for the upper bound is

$$P_B \leq \sum_{d=d_{free}}^{\infty} \beta_d P_2(d) \tag{4.1}$$

where $P_2(d)$ is the probability that an incorrect path with degree d is selected when assuming that the correct path is the all zero path, which is given by Eqn. (2.14) in Chapter 2. β_d is the number of bit errors totaled over all paths with degree d. The upper bound P_B is practically approximated by the summation of the first few dominant terms. In our simulations, we consider the first 10 terms. The values of β_d and d_{free} for the three convolutional coding rates used in 802.11a/g are listed in Tab. 4.1 [13, 22]. Fig. 4.1 shows the resulting crossover probability vs SNR curves for each of the

FEC rate	PHY rate (Mb/s)	d_{free}	β_d
1/2	6, 12, 24	10	36, 0, 211, 0, 1404, 0, 11633, 0, 77433, 0
3/4	9, 18, 36, 54	5	42, 201, 1492, 10469, 62935, 379644, 2253373, 13073811, 75152755, 428005675
2/3	48	6	3, 70, 285, 1276, 6160, 27128, 117019, 498860, 2103891, 8784123

Table 4.1: d_{free} and β_d values for the three convolutional coding rates used in 802.11a/g

802.11a/g OFDM PHY rates in a Rayleigh channel.

Fig. 4.2 plots the overall capacities for both PEC and BSC. The capacity curves are obtained by selecting the PHY rate R that maximises the capacity at each SNR given a channel model. It can be seen that capacity improvements of 100% or more are indicated over a wide range of SNRs. Note that these curves do not include the MAC layer framing overheads, contention time, collision losses and
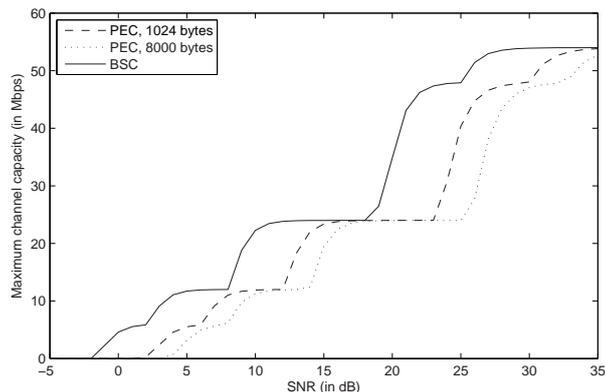
Figure 4.2: 802.11a/g BSC and PEC capacities vs. SNR, Rayleigh physical channel. Packet erasure capacities are shown for frame sizes of both 1024 bytes and 8000 bytes.

etc. In the following sections we extend the analysis to include these overheads for both unicast and multicast traffic. We also extend the analysis to take account of the multi-user nature of the wireless channel, including the multi-destination aggregation.

4.3   BSC-based Coding in Multi-user Channels

In a multi-user network the channel quality between the transmitter and every receiver is generally different for every receiver owing to a multitude of reasons, such as differences in distance between transmitter and receivers, obstacles such as walls when operating indoors and differences in the local interference environment. To make things concrete we consider an example of an 802.11 WLAN with an AP and two classes of client stations. The channel qualities for two classes are quite different, *e.g.* class 1 is located far from the AP such that stations in class 1 have low SNRs and are thus subjected to noisy reception, but class 2 is located close to the AP where transmissions at any available PHY rate can be reliably received. The state of the art is that the AP transmits multi-destination aggregated packets to all stations at the highest PHY rate which the station with the worst channel quality can support. This causes an inefficiency in network throughput. The BSC paradigm allows transmission of a packet with different segments being encoded with different levels of protection. In this manner we can transmit at different information rates to different destinations while using a single PHY rate exploiting developments in multi-terminal information theory [15].
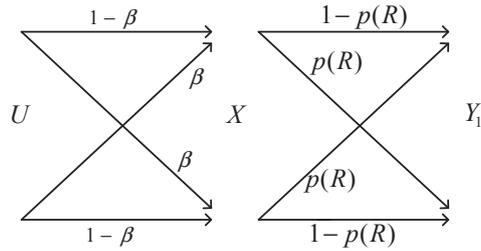
Figure 4.3: Physically degraded binary symmetric broadcast channel

4.3.1 Superposition Coding

The specific example of two classes of stations with different levels of SNR forms what is known as a physically degraded binary symmetric broadcast channel [15]. For this class of channels superposition coding [8] is known to be capacity-achieving. Superposition coding works by first ordering users by decreasing channel quality and constructing the code of each user based upon the codes of all the users before. Decoding starts from the last user for whom the decoder treats every other user's signal as noise. After the last user is decoded, its contribution is subtracted and the remaining users are decoded in a similar nested fashion. In more detail, for the example of two classes of stations, in superposition coding the message (*i.e.* a binary vector) U destined to class 1 and the message V destined to class 2 are summed, modulo 2, and transmitted. The binary vector Y_1 received by class 1 can be viewed as message U corrupted by a bit flips due to a combination of channel noise and the summation with V, that is, the message U destined to the class 1 stations is first passed through a BSC with crossover probability β determined by the entropy of V, and then transmitted over the physical BSC with crossover probability $p(R)$ given PHY rate R. This is illustrated in Fig. 4.3. The channel capacity for class 1 in bits per channel use is given by

$$C_1 = 1 - H(\beta \circ p(R)) \tag{4.2}$$

where $H(\cdot)$ is the entropy function of a Bernoulli random variable, *i.e.* $H(x) = -x \log_2(x) - (1 - x) \log_2(1 - x)$, and $\beta \circ p(R) = \beta(1 - p(R)) + (1 - \beta)p(R)$.

The message received by class 2 Y_2 is the transmitted message X, i.e. $Y_2 = X$, provided that the channel of class 2 is error-free. The channel capacity for class 2 is thus

$$C_2 = I(X; X|U) = H(X|U) = H(\beta) \tag{4.3}$$

4.3.2　Time-sharing FEC Coding

From the discussion above it is clear that superposition coding can be a complex operation. A simpler but demonstrably near-optimal choice is a time-sharing based coding scheme [15]. For our setting and with 2 classes of stations, time-sharing based coding is such that each MAC frame is partitioned (*i.e.* time-shared) into a portion intended for class 1 and a portion intended for class 2. Using the same setting that was illustrated above for superposition coding, the portion intended for class 2 is effectively error-free and thus does not need further protection. However, the portion intended for class 1 is protected by an FEC code that allows information to be extracted even when bits within the frame are corrupted; the information rate is obviously reduced compared to a noise-free channel. We assume an ideal FEC code with the coding rate matched to the BSC capacity. Provided the aggregated frame is sufficiently large, capacity-approaching codes, *e.g.* LDPC codes [56], exist.

In this chapter we present a performance analysis of superposition coding, which is known to be capacity-achieving in multi-user BSCs, and of the simpler time-sharing FEC coding scheme. We show that there is minimal loss of performance when using the time-sharing coding scheme. For both schemes our analysis indicates the potential for substantial performance gains over the conventional PEC paradigm. To our knowledge, this is the first such analysis of multi-user coding in 802.11 WLANs.

4.4　Unicast Throughput Modelling

The vast majority of network traffic is unicast, and contention between multiple stations in an 802.11 WLAN, with associated collision losses and increased CSMA/CA countdown time, is the norm. We still consider the example that an 802.11 WLAN has an AP and two classes of client stations, with n_1 stations in class 1 and n_2 in class 2. We assume that all stations in the same class have the same SNR. Class 1 is located far from the AP with low SNR such that stations in this class are subjected to noisy reception; while class 2 lies within a region where stations have high SNR and thus experience reliable reception at any of the available PHY data rates. The analysis can be readily generalised to encompass situations where each client station has a different SNR, but the two-class case is sufficient to capture performance features of heterogeneous settings in WLANs.

The AP has N_1 unicast downlink flows destined to n_1 class 1 stations and N_2 unicast downlink flows to n_2 class 2 stations. In general $N_1 \neq n_1$ and $N_2 \neq n_2$. For the unicast scenario we consider we

set $N_1 = n_1$ and $N_2 = n_2$. The AP aggregates these $N_1 + N_2$ downlink flows into a single large MAC frame and then transmits it at a single PHY rate. Each client station also has an uplink flow for the AP. For simplicity we assume that all stations are saturated, although the analysis could be extended to include unsaturated operation using, for example, the approach in [46]. We also assume that uplink transmissions by client stations are immediately acknowledged by the AP (rather than, for example, using a block ACK proposed in 802.11e [29]). Similarly, we assume that downlink transmissions are immediately acknowledged by client stations and, to make our analysis concrete, we adopt the approach described in [17] which uses the orthogonality of OFDM subcarriers to allow a group of client stations to transmit feedback signals at the same time, and thereby ACK collisions are avoided. However, these assumptions really just relate to the calculation of the MAC overheads and our analysis could be readily modified to account for alternative acknowledgement mechanisms[1].

To ensure a fair comparison amongst different schemes it is not sufficient to simply compare sum-throughputs. Rather we also need to ensure that schemes provide comparable throughput fairness, since an approach may achieve throughput gains at the cost of increased unfairness. In the following we take a max-min fair approach and impose the fairness constraint that all flows achieve the same throughput. Extension of the analysis to other fairness criteria is, of course, possible.

4.4.1 MAC Model

In this section, we give the detailed theoretical throughput performance analysis for three aggregation approaches: i) standard frame aggregation in a PEC paradigm; ii) aggregation with superposition coding in a BSC paradigm; iii) aggregation with time-sharing coding in a BSC paradigm.

Transmissions by the AP are subjected to collisions with competing uplink transmissions, while transmissions by client stations are subjected to collisions with the AP and uplink transmissions from other client stations. In the PEC paradigm transmissions involving class 1 stations are also subjected to noise losses, while those involving class 2 stations are not subjected to noise. Therefore the probability that a transmission from a class 1 station fails (due to collision and/or loss) is

$$p_{f1} = 1 - (1 - p_{cU1})(1 - p_{eU1}) \tag{4.4}$$

---

[1]In particular, for operation on standard 802.11n hardware we might tunnel data packets for multiple destinations via multicast MPDUs aggregated into an A-MPDU (thereby achieving standards-compliant multi-destination aggregation) and generate appropriate acknowledgements either by modifying the receiver NIC driver or at the application layer.

where p_{eU1} is the probability that an uplink packet from a class 1 station is erased due to noise, and p_{cU1} is the probability that an uplink transmission from a class 1 station collides, given by,

$$p_{cU1} = 1 - (1 - \tau_1)^{n_1 - 1}(1 - \tau_2)^{n_2}(1 - \tau_0) \tag{4.5}$$

in which τ_0, τ_1 and τ_2 are respectively the attempt probability of AP, class 1 stations and class 2 stations. As the AP and class 2 stations back off only on collisions, according to Bianchi model [9], we have $\tau_0 = \tau_2$. The probability that a transmission from the AP or a class 2 station fails is thus

$$p_{f2} = 1 - (1 - \tau_1)^{n_1}(1 - \tau_2)^{n_2} \tag{4.6}$$

The usual Bianchi [9] expression gives the relation between the attempt probability τ and the probability p_f that a transmission fails. However, we make use of expression (4.7) that builds upon the Bianchi expression taking into account a finite number of retransmission attempts and losses due to decoding errors [51].

$$\tau = \begin{cases} \dfrac{2(1-2p_f)(1-p_f^{m+1})}{(1-p_f)W(1-(2p_f)^{m+1})+(1-2p_f)(1-p_f^{m+1})} & m \leq m', \\ \dfrac{2(1-2p_f)(1-p_f^{m+1})}{(1-p_f)W(1-(2p_f)^{m+1})+(1-2p_f)(1-p_f^{m+1})+W2^{m'}p_f^{m'+1}(1-2p_f)(1-p_f^{m-m'})} & m > m'. \end{cases} \tag{4.7}$$

in which $W = CW_{min}$; m denotes the 802.11 retry limit number, and m' represents the number of doubling the CW size from CW_{min} to CW_{max}.

4.4.2 Packet Erasure Paradigm

Similarly to the approach used in 802.11n A-MPDUs [31], we consider a situation where messages addressed to distinct destinations are aggregated together to form a single large MAC frame. We do not present results here without aggregation since the throughputs are strictly lower than when aggregation is used [37]. To keep our discussion concrete, we assume the frame format is as shown in Fig. 4.4. However, it is important to stress that this really just relates to the calculation of the MAC overheads and our analysis could be readily modified to account for alternative frame formats. In Fig. 4.4 a sub-header is prefixed to each IP packet to indicate its receiver address, source address and packet sequence information. An FCS checksum is used to detect corrupted packets. Since the sub-header already contains the receiver address, source address and sequence control, the MAC header removes these three fields, but keeps other fields unchanged from the standard 802.11 MAC header. For simplicity, we assume that the MAC header is transmitted at the same PHY rate as the PLCP header and thus can be assumed error-free in the following analysis, although we will relax this
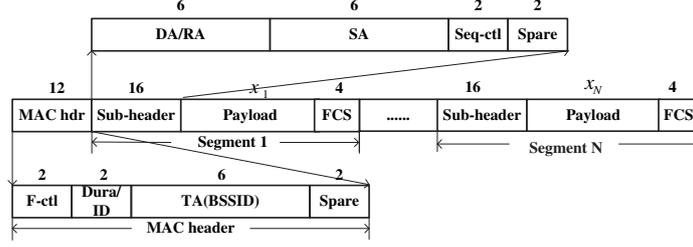
Figure 4.4: Erasure channel frame format.

assumption in future work. The frame format is known to all user stations, so that each station can correctly locate its packet even if some bits in the frame are corrupted.

In the PEC scenario, the downlink transmission rate is determined by the worst client which has the lowest SNR, hence we have the downlink PHY rate R_0 used by the AP equal to the uplink PHY rate of class 1 stations R_1 in order to meet the max-min fairness objective. For a given SNR and R_0, the union bound on the first-event error probability of Viterbi decoding is $p_e(R_0)$. The packet erasure rate of an uplink packet from class 1 stations $p_{eU1}(R_0)$ is then

$$p_{pU1}(R_0) = 1 - (1 - p_e(R_0))^{L_{U1}(R_0)} \tag{4.8}$$

where $L_{U1}(R_0)$ is the uplink frame size in bits from a class 1 station, given by

$$L_{U1}(R_0) = DBPS(R_0) \times \left\lceil \frac{(x_{U1}(R_0) + L_{machdr} + L_{FCS}) \times 8 + 6 + 16}{DBPS(R_0)} \right\rceil \tag{4.9}$$

where $DBPS(R_0)$ represents data bits per symbol at PHY rate R_0, $x_{U1}(R_0)$ is the payload size in bytes of an uplink packet from a class 1 station, and L_{machdr} and L_{FCS} are the length in bytes of the standard MAC header and the FCS field. The expected payload delivered by an uplink packet of a class 1 station is

$$E_{U1}(R_0) = x_{U1}(R_0)(1 - p_e(R_0))^{L_{U1}(R_0)} \tag{4.10}$$

Let x_{U2} denote the payload size in bytes of class 2 uplink transmissions. As uplink transmissions by class 2 stations are assumed to be loss-free at all supported rates and in our performance analysis we consider the range of OFDM-based PHY rates supported by 802.11a/g, we take the transmission rate for uplink class 2 transmissions $R_2 = 54$Mbps. The expected payload of an uplink packet from a class 2 station at R_0 is

$$E_{U2}(R_0) = x_{U2}(R_0) \tag{4.11}$$

Turning now to the AP, similar to the approach used in 802.11n, the aggregated MAC frame then consists of $N_1 + N_2$ unicast packets. Let $x_{D1}(R_0)$ and $x_{D2}(R_0)$ denote the downlink payload size in bytes to class 1 and class 2 stations at R_0 respectively. The length of a MAC frame L is thus

$$L = N_1 x_{D1}(R_0) + N_2 x_{D2}(R_0) + (N_1 + N_2)(L_{subhdr} + L_{FCS}) \qquad (4.12)$$

where L_{subhdr} is the sub-header length. The expected payload delivered to a class 1 station in a downlink packet is

$$E_{D1}(R_0) = x_{D1}(R_0) \times (1 - p_e(R_0))^{(x_{D1}(R_1) + L_{subhdr} + L_{FCS}) \times 8} \qquad (4.13)$$

The expected payload delivered to a class 2 station in a downlink packet is

$$E_{D2}(R_0) = x_{D2}(R_0) \qquad (4.14)$$

To equalize the throughput of each flow, we require

$$x_{D2}(R_0) = x_{D1}(R_0)(1 - p_e(R_0))^{(x_{D1}(R_0) + L_{subhdr} + L_{FCS}) \times 8} \qquad (4.15)$$

$$x_{U2}(R_0) = x_{D2}(R_0) \qquad (4.16)$$

$$\tau_1(1 - \tau_2)(1 - p_e(R_0))^{L_{U1}(R_0)} x_{U1}(R_0) = \tau_2(1 - \tau_1) x_{U2}(R_0) \qquad (4.17)$$

For a given R_0 and a fixed L we can solve equations (4.12) and (4.15) to obtain x_{D1} and x_{D2}. As τ_1 and τ_2 also depend on R_0, combining the expressions (4.7) for τ_1 and τ_2 from the MAC model with expression (4.17), we can solve to obtain τ_1 , τ_2 and x_{U1}.

To obtain the throughputs it remains to derive expressions for the expected duration of a MAC slot. There are four possible types of MAC slot. We consider each in turn.

- Type 1 - AP transmits: Observe that the transmission duration of an AP frame is larger than that of the client stations (due to aggregation). If the AP transmits during the slot, then regardless of whether it suffers from a collision the duration of the slot in μs is

$$T_{AP} = T_{phyhdr1} + T_D(R_0) + T_{sifs} + T_{phyhdr} + T_{ack} + T_{difs} \qquad (4.18)$$

where $T_{phyhdr1}$ is the PHY/MAC header duration that we define for an aggregated frame; T_{phyhdr} is the standard PHY header duration. T_{ack} is the transmission duration of an ACK frame; $T_D(R_0)$ is the transmission duration of a downlink MAC frame,

$$T_D(R_0) = \lceil (L \times 8 + 6 + 16)/DBPS(R_0) \rceil \times 4 \qquad (4.19)$$

63

- Type 2 - class 1 transmits: Observe that the duration of class 1 frames is larger than that of class 2 stations (due to channel noises for class 1). If a class 1 station gets the transmission opportunity, and its transmission does not collide with a downlink transmission, but might collide with other uplink transmissions, the duration in μs is

$$T_1 = T_{phyhdr} + T_{U1}(R_0) + T_{sifs} + T_{phyhdr} + T_{ack} + T_{difs} \tag{4.20}$$

where $T_{U1}(R_0)$ is the transmission duration of an uplink MAC frame from a class 1 station, given by

$$T_{U1}(R_0) = 4 \times \lceil \frac{(x_{U1}(R_0) + L_{machdr} + L_{FCS}) \times 8 + 22}{DBPS(R_1)} \rceil \tag{4.21}$$

The probability that a collision occurs among uplink packets (not involving downlink packet) is

$$p_{cUP} = (1-\tau_0)\big(1-(1-\tau_1)^{n_1}(1-\tau_2)^{n_2} - n_1\tau_1(1-\tau_1)^{n_1-1}(1-\tau_2)^{n_2} - n_2\tau_2(1-\tau_2)^{n_2-1}(1-\tau_1)^{n_1}\big) \tag{4.22}$$

The probability that a collision occurs only among uplink packets from class 2 stations is

$$p_{cUP2} = (1-\tau_0)(1-\tau_1)^{n_1}\big(1-(1-\tau_2)^{n_2} - n_2\tau_2(1-\tau_2)^{n_2-1}\big) \tag{4.23}$$

Hence, the probability that the duration is T_1 is

$$p_{T_1} = n_1\tau_1(1-\tau_1)^{n_1-1}(1-\tau_2)^{n_2}(1-\tau_0) + p_{cUP} - p_{cUP2} \tag{4.24}$$

- Type 3 - only class 2 transmits: If a class 2 station makes a transmission which does not collide with an AP or class 1 station transmissions, but might collide with uplink packets from other class 2 stations, the duration within a slot time in μs is

$$T_2 = T_{phyhdr} + T_{U2}(R_0) + T_{sifs} + T_{phyhdr} + T_{ack} + T_{difs} \tag{4.25}$$

where $T_{U2}(R_0)$ is the transmission duration in μs of an uplink MAC frame from a class 2 station,

$$T_{U2}(R_0) = 4 \times \lceil \frac{(x_{U2}(R_0) + L_{machdr} + L_{FCS}) \times 8 + 22}{54 \times 4} \rceil \tag{4.26}$$

The probability that the duration is T_2 is given by

$$p_{T_2} = n_2\tau_2(1-\tau_2)^{n_2-1}(1-\tau_0)(1-\tau_1)^{n_1} + p_{cUP2} \tag{4.27}$$

- Type 4 - idle slot: If no transmission occurs, the duration is a PHY slot σ. This event occurs with probability

$$p_{Idle} = (1-\tau_1)^{n_1}(1-\tau_2)^{n_2}(1-\tau_0) \tag{4.28}$$

64

Combining these yields the expected MAC slot duration,

$$E_T = p_{Idle}\sigma + \tau_0 T_{AP} + p_{T_1}T_1 + p_{T_2}T_2 \tag{4.29}$$

The network throughput is then given by

$$S(R_0) = \frac{N_1 X_1 + N_2 X_2}{E_T} \tag{4.30}$$

where

$$X_1 = \tau_0(1-\tau_1)^{n_1}(1-\tau_2)^{n_2}E_{D1} + \tau_1(1-\tau_1)^{n_1-1}(1-\tau_2)^{n_2}(1-\tau_0)E_{U1} \tag{4.31}$$

$$X_2 = \tau_0(1-\tau_1)^{n_1}(1-\tau_2)^{n_2}E_{D2} + \tau_2(1-\tau_1)^{n_1}(1-\tau_2)^{n_2-1}(1-\tau_0)E_{U2} \tag{4.32}$$

We select the downlink PHY rate R_0^* (equal to the uplink PHY rate of class 1 stations) from the set $\mathcal{R}$ of supported 802.11a/g rates so as to maximise this throughput given the channel SNR.

4.4.3 BSC Time-sharing Coding

For the BSC paradigm we start by considering the simpler time-sharing coding scheme. As in the erasure channel case, MAC frames are constructed by aggregating two portions intended for the different classes of stations with the portion meant for class 1 stations coded (based on BSC crossover probability) and the remainder (for class 2 stations) being uncoded. Note that each portion is in itself composed of sub-frames meant for the different stations. We also apply similar coding to protect uplink transmissions from class 1 stations to allow information to be recovered from corrupted uplink frames.

Let x_{D1} denote the downlink information payload size for a class 1 station and x_{D2} for a class 2 station. Suppose a downlink PHY rate R_0 is chosen. The crossover probability for class 1 stations is $p(R_0)$ and the number of coded bits for x_{D1} is $x_{D1}/(1 - H(p(R_0)))$. To equalize the downlink throughputs of stations in both classes, we therefore require

$$x_{D1}(R_0) = x_{D2}(R_0) \tag{4.33}$$

Given a frame size L, we have that

$$L = N_1 \cdot \frac{x_{D1}(R_0) + L_{subhdr} + L_{FCS}}{1 - H(p(R_0))} + N_2 \cdot (x_{D2}(R_0) + L_{subhdr} + L_{FCS}) \tag{4.34}$$

and hence $x_{D1}(R_0)$ and $x_{D2}(R_0)$ can be solved. As downlink packets to class 1 stations are erasure-free in the BSC paradigm, the expected payload delivered to a class 1 station is $E_{D1} = x_{D1}(R_0)$.

To equalize the uplink and downlink throughput from/to class 2 stations we require $x_{U2}(R_0) = x_{D2}(R_0)$. Since erasure-free frames are delivered, $p_{eU1} = 0$ and thus $\tau_0 = \tau_1 = \tau_2$. Hence, to equalize the uplink and downlink throughput from/to class 1 stations we require $x_{U1}(R_0) = x_{D1}(R_0)$.

The uplink PHY rate of a class 1 station R_1^* is selected in terms of maximising its BSC capacity, *i.e.*

$$R_1^* = \arg \max_{r \in \mathcal{R}} \ r \cdot \Big(1 - H\big(p(r)\big)\Big) \tag{4.35}$$

The expected duration in a slot time E_T and the network throughput $S(R_0)$ are derived in a similar manner to the erasure case. We select the downlink rate R_0^* so as to maximise the network throughput given the channel SNR.

4.4.4 BSC Superposition Coding

The MAC frames in this setting are constructed in two steps. Once a value of β has been determined, the V and U bit vectors are generated by aggregating IP packets of each class, and are then summed, modulo 2, to generate the MAC frame. Despite the scheme being more complicated, the analysis with superposition coding is similar to the BSC time-sharing case. The main difference lies in the calculation of the downlink payload size.

Suppose the downlink PHY rate used by the AP is R_0, the downlink BSC capacity in bits per channel use between the AP and a class 1 station is $1 - H(\beta \circ p(R_0))$, and that between the AP and a class 2 station is $H(\beta)$. The MAC frame body is formed by superimposing N_2 downlink unicast packets destined to class 2 stations to N_1 downlink unicast packets destined to class 1 stations. Let x_{D1} denote the downlink information payload size for a class 1 station, and x_{D2} denote the downlink information payload size for a class 2 station. Given the MAC frame size L, we have

$$L = \frac{N_1(x_{D1}(R_0) + L_{subhdr} + L_{FCS})}{1 - H(\beta \circ p(R_0))} = \frac{N_2(x_{D2}(R_0) + L_{subhdr} + L_{FCS})}{H(\beta)} \tag{4.36}$$

To equalize the downlink throughputs of stations in both classes, we require

$$x_{D1}(R_0) = x_{D2}(R_0) \tag{4.37}$$

Thus we find the relationship $N_1 H(\beta) = N_2\Big(1 - H\big(\beta \circ p(R_0)\big)\Big)$. The ratio N_1/N_2 fixes the value of β. With the value of β determined, the downlink unicast payload size for each client station (class 1

or class 2) is then given by

$$x_D(R_0) = \max\left(\frac{LH(\beta)}{N_2} - L_{subhdr} - L_{FCS}, 0\right)$$
$$= \max\left(\frac{L\big(1 - H(\beta \circ p(R_0))\big)}{N_1} - L_{subhdr} - L_{FCS}, 0\right) \tag{4.38}$$

The uplink PHY rate of class 1 stations R_1^* is selected in the same manner as the BSC time-sharing case. We also protect uplink transmissions using the ideal code. Again, the PHY rate R_0^* is chosen to maximise system throughput.

4.5 Multicast Throughput Modelling

In the multicast scenario we still consider the example of an 802.11 WLAN with an AP and two classes of stations, as described in Section 4.4. But the AP multicasts only two downlink flows which are aggregated into a large MAC frame. Flow 1 is communicated to n_1 class 1 stations and flow 2 is communicated to n_2 class 2 stations, respectively. There are no competing uplink flows. Therefore, we can compute the throughput using the analysis in Section 4.4 by setting the following parameter values: $N_1 = N_2 = 1$; $x_{U1} = x_{U2} = 0$; $p_{eU1} = p_{cU1} = 0$; $\tau_1 = \tau_2 = 0$; $\tau_0 = 2/(W_0 + 1)$, where W_0 is the minimum contention window size. The expected payload and MAC slot duration can be calculated using the same method as the unicast analysis, but for a multicast network, we consider per-station multicast saturation throughput,

$$S(R_0) = \frac{\tau_0 E_{D1}(R_0)}{E_T(R_0)} = \frac{\tau_0 E_{D2}(R_0)}{E_T(R_0)} \tag{4.39}$$

The optimal PHY rate R_0^* is selected to maximise the per-station throughput.

4.6 Theoretical Performance

In this section, we compare the throughput performance of the PEC and BSC schemes for both unicast and multicast scenarios respectively considered in Section 4.4 and Section 4.5. By combining the above theoretical analysis (which gives the throughput as a function of the channel error rate, *i.e.* the packet erasure rate for PEC and the bit crossover probability for BSC) with the experimental measurement results shown in Fig. 3.8 (which gives the channel error rate as a function of RSSI and

PHY rate), we can determine the optimal transmission PHY rates for downlink and uplink flows and obtain the maximum network throughput for a range of RSSIs for both PEC and BSC approaches. The protocol parameters used in the simulations are listed in Table 4.2.

The experimental PEC capacity shown in Fig. 3.8 is for a frame length of 8640 bits. To obtain the PEC capacity for any other values of frame length, we need to first derive the first event error probability of Viterbi decoding for convolutional codes, which is given by

$$P_e = 1 - (1 - FER)^{(1/l)} \tag{4.40}$$

where FER is the measured frame erasure rate at a given RSSI, and l is the frame length in the experiment, $i.e.$ 8640 bits. Using this first event error probability P_e, the frame erasure rate for a frame length of L is, in turn, given by $1 - (1 - P_e)^L$.

4.6.1 Unicast

We first consider the unicast traffic scenario. We compare the throughput performance for four different approaches: 1) PEC paradigm; 2) BSC time-sharing coding with the entire frame transmitted at a single PHY rate; 3) BSC superposition coding; 4) BSC time-sharing coding with segments transmitted at different PHY rates, $i.e.$ segments destined to class 2 are transmitted at the highest PHY rate available, which is 54Mbps in 802.11a/g, and the downlink PHY rate for class 1 is selected to maximise the network thorughput. Fig. 4.5 shows the sum-throughputs achieved by different approaches for a network consisting of 20 client stations, 10 in class 1 and 10 in class 2. This is quite a large number of saturated stations for an 802.11 WLAN and suffers from a high level of collision losses. Comparing with Fig. 3.8, it can be seen that the throughput is significantly reduced due to the various protocol overheads and collisions that have now been taken into account. Nevertheless, the relative throughput gain of the BSC-based approaches compared to the erasure channel approach continues to exceed 50% for a wide range of RSSIs. BSC time-sharing coding achieves very similar performance to the more sophisticated superposition coding. The approach of using different PHY rates for different BSC-based time-sharing coding segments achieves even higher throughputs than using the same PHY rate. The gains are especially high at low RSSIs. This is because when the entire frame is transmitted using the same PHY rate, the optimal PHY rates for PEC and BSC schemes are usually not far different, $i.e.$ it is impossible that the PEC scheme chooses 6Mbps but the BSC scheme chooses 54Mbps. However when segments destined to distinct receivers use different PHY rates, $e.g.$

T_{sifs} (μs)	16	L_{subhdr} (bytes)	16
T_{difs} (μs)	34	L_{FCS} (bytes)	4
Idle slot duration σ (μs)	9	L_{machdr} (bytes)	24
T_{ack} (μs)	24	CW_{min}	16
T_{phyhdr} (μs)	20	CW_{max}	1024
$T_{phyhdr1}$ (μs)	36	Retry limit	7

Table 4.2: Protocol parameters used in simulations

Figure 4.5: Unicast maximum network throughput vs. RSSI of class 1 stations, $L = 8000$ bytes, with $n_1 = n_2 = 10$ stations. TS and SPC indicate time-sharing coding and superposition coding respectively.

in our two-class example, the portion for class 2 always uses a quite high PHY rate such as 54Mbps, while the portion for class 1 could use a very low PHY rate, especially at low RSSIs.

Fig. 4.6 shows the corresponding results for a smaller number of client stations, 5 in class 1 and 5 in class 2. The overall throughput is higher than that with 20 stations because of the lower chance of collisions, and the gain offered by BSC approaches is even higher *i.e.* more than 75% over a wide range of RSSIs.

Fig. 4.7 illustrates how the number of stations affects these results. The decrease in network throughput with increasing number of stations is evident, as is the significant performance gain offered by the
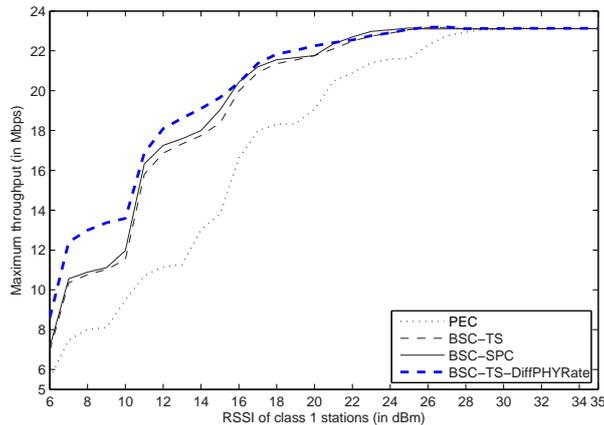
Figure 4.6: Unicast maximum network throughput vs. RSSI of class 1 stations, $L = 8000$ bytes, with $n_1 = n_2 = 5$ stations. TS and SPC indicate time-sharing coding and superposition coding respectively.

BSC schemes. For smaller numbers of stations (which is perhaps more realistic), the throughput gain offered by the BSC approaches is larger *e.g.* nearly up to 70% for 2 stations and falling to around 30% with 20 stations. The proportion of class 1 and class 2 stations can be expected to affect the relative performance of the PEC and BSC schemes. This is because we now have multiple transmitting stations, and each station defers its contention window countdown on detecting transmissions by other stations. Since class 1 transmissions are of longer duration than class 2 transmissions, we expect that the network throughput will rise as the number of class 1 stations falls and indeed we find that this is the case. See, for example, Fig. 4.8 which plots the network throughput versus the varying ratio of the number of class 2 stations over the total number while maintaining the total number of client stations constant as $n_1 + n_2 = 10$.

4.6.2 Multicast

For multicast, we compare the per-station throughput for those four aggregation approaches. Fig. 4.9 shows the per-station throughput for a network with $n_1 = 10$ class 1 stations and $n_2 = 10$ class 2 stations. The throughput is much higher than the unicast case as shown in Fig. 4.5 because of the absence of collisions with uplink flows. Nevertheless, both of the BSC schemes (time-sharing and superposition coding) continue to offer substantial performance gains over the erasure channel approach, increasing throughput by almost 100% over a wide range of RSSIs. The superposition

70
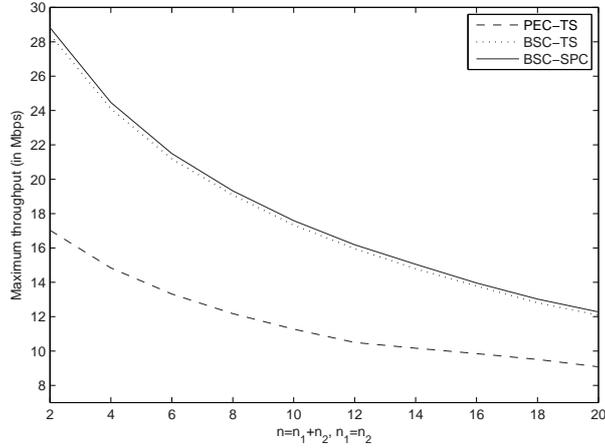
Figure 4.7: Unicast maximum network throughput vs. varying total number of stations for a fixed proportion of class 2 stations $n_1 = n_2$, $RSSI = 13$dBm, $L = 8000$ bytes. TS and SPC indicate time-sharing coding and superposition coding respectively.
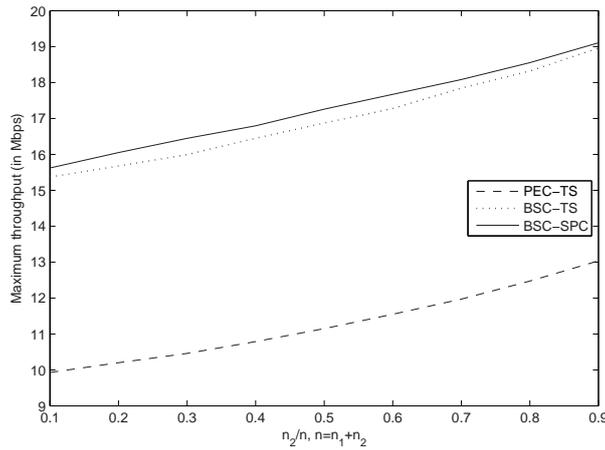
Figure 4.8: Unicast maximum network throughput vs. varying proportion of class 2 stations for a fixed total number of stations $n_1 + n_2 = 10$, $RSSI = 12$dBm, $L = 8000$ bytes. TS and SPC indicate time-sharing coding and superposition coding respectively.
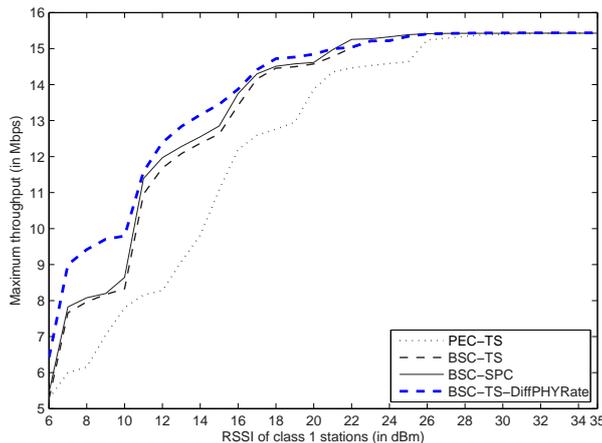
71
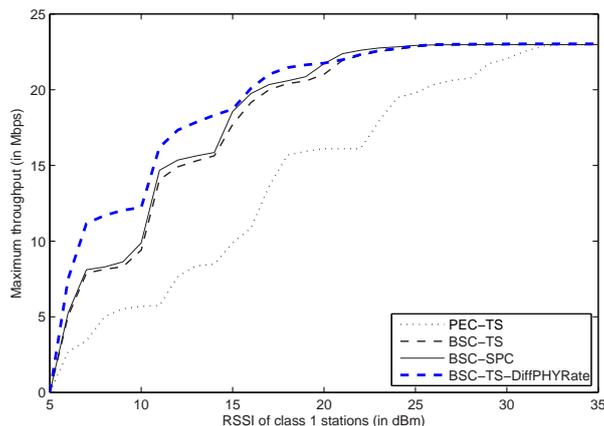
Figure 4.9: Multicast per-station maximum throughput vs RSSI of class 1 stations, $L = 8000$ bytes, $n_1 = n_2 = 10$ stations. TS and SPC indicate time-sharing coding and superposition coding respectively.

coding scheme performs slightly better than the time-sharing scheme, but the difference is minor. Fig. 4.10 shows the corresponding results with a larger MAC frame size of 65536 bytes, which is the maximum frame size allowed in the 802.11n standard [31]. The performance gain offered by the BSC approaches increases as the frame size is increased. Since the per-station multicast throughput is independent of the number of stations, we only show results for one value of n_1 and n_2.

4.7 NS-2 Simulations

The theoretical performance in Section 4.6 does not consider the impact of queueing which affects the availability of packets to be aggregated together. In this section we implement the PEC-based scheme and the BSC-based time-sharing coding scheme in the network simulator NS-2 (see Appendix C). In addition to the throughput performance, we also compare the delay which may be caused by queueing. We consider the multicast scenario with an AP and two classes of stations. Again, we assume that class 1 is subjected to noisy reception, while class 2 experiences successful reception at any of the available PHY rates. Both classes have the same number of stations, and all stations in the same class have the same SNR. The AP has an individual downlink flow for each of the stations. Each flow is generated using a CBR (constant bit rate) traffic agent. The CBR packet size is 1500 bytes in our
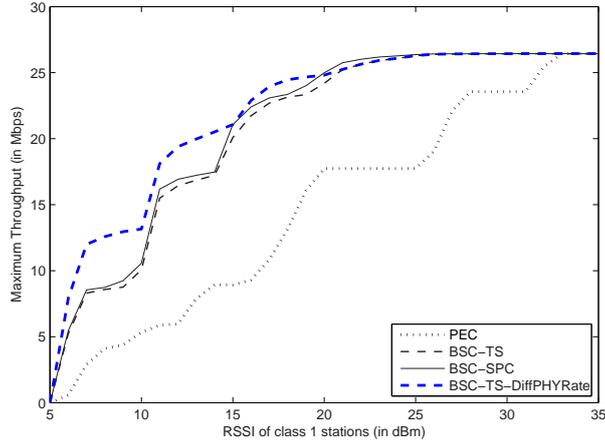
Figure 4.10: Multicast per-station maximum throughput vs. RSSI of class 1 stations, $L = 65536$ bytes, $n_1 = n_2 = 10$ stations. TS and SPC indicate time-sharing coding and superposition coding respectively.

simulations.

We consider a few different assumptions from the theoretical analysis in Section 4.5:

- Fixed sub-frame size: In NS-2 simulations, we assume that each sub-frame has a fixed length, which is the CBR packet size plus protocol headers of UDP, RTP, IP, MAC layers and the sub-header. The maximum aggregated MAC frame size is 65535 bytes.

- No fairness requirements: Due to the fixed sub-frame size (not the fixed large MAC frame size), we cannot adjust the sub-frame size for each class to achieve fairness amongst stations, hence we do not consider fairness issues in NS-2 simulations.

- No MAC acknowledgements: To achieve standard compliance, we use the broadcast/multicast mode to transmit multi-destination aggregated packets in NS-2 simulations, and thus there are no MAC acknowledgements. We assume that acknowledgement packets are from the application layer.

We use two metrics to evaluate the performance:

- Per-station throughput (in Mbps): Let n_i denote the number of packets received by station i

73

during the simulation duration t. Packets have a length of L in bytes. The throughput of station i is given by $8Ln_i/t$. The per-station throughput is the mean over all stations, which is then given by $\sum_{i=1}^{N} 8Ln_i/(tN)$, where N is the total number of stations in a WLAN.

- Mean delay (in second): We define the delay of a packet as the period since it arrives at the interface queue (IFQ) of the transmitter till it arrives at the MAC layer of the receiver. The mean delay is the mean over all packets. In our simulation, we use a DropTail FIFO queue, and the maximum queuing size is 500 in packets.

Fig. 4.11 shows an example of two-class setup in a Rayleigh channel. We assume class 1 has a SNR of 22dB, and class 2 has a SNR of 30dB. In the PEC paradigm, we use the PHY rate of 24Mbps, while in the BSC paradigm we choose a higher PHY rate of 48Mbps (Note that the choice of PHY rates here is not optimal to achieve the maximum throughput). Given a SNR value and PHY rate, the PEC first event error probability and the BSC crossover probability can be obtained from the theoretical Rayleigh channel model, respectively described in Section 2.3.2 and Section 4.2. The CBR traffic bit rate is 2Mbps.

It can be seen that the BSC time-sharing coding scheme offers almost 100% increase in per-station throughput, but the mean delay is only half that of the PEC scheme. When the station number is below 20, the mean delay of BSC time-sharing scheme is even close to 0. This is because in the BSC paradigm, with such small numbers of flows, the queue in the AP is not full yet. There may not be enough packets available to aggregate into the maximum packet size. As the number of stations increases, the queue gets full, and the mean delay includes the duration of a packet waiting in the queue.

We also did NS-2 simulations using our outdoor experimental measurement data. We assume that class 1 has a RSSI of 12dBm, and class 2 has a RSSI of 35dBm. The PEC approach uses the PHY rate of 18Mbps, while the BSC approach uses the PHY rate of 36Mbps. The CBR bit rate is set to be 1Mbps. Fig. 4.12 shows the corresponding results. Besides the substantial increase in throughput and decrease in delay, we also find that the queue in BSC paradigm can support more stations before overflowing due to the shorter delay for each packet. It can be seen from Fig. 4.12(b) that the queue overflows when there are 20 stations in the PEC scheme, but in the BSC scheme it does not overflow until the station number increases up to 32.

74
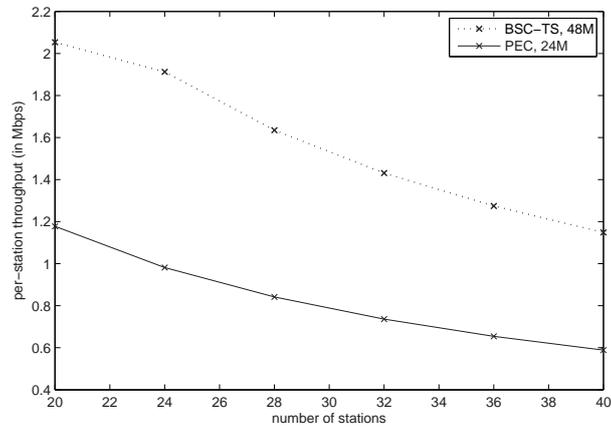
(a) Per-station throughput
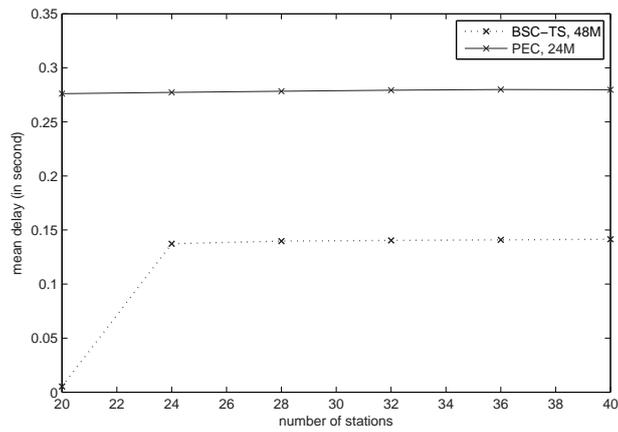
(b) Mean delay

Figure 4.11: Per-station throughput and mean delay vs number of stations, Rayleigh channel, SNR of class 1 22dB, SNR of class 2 30dB
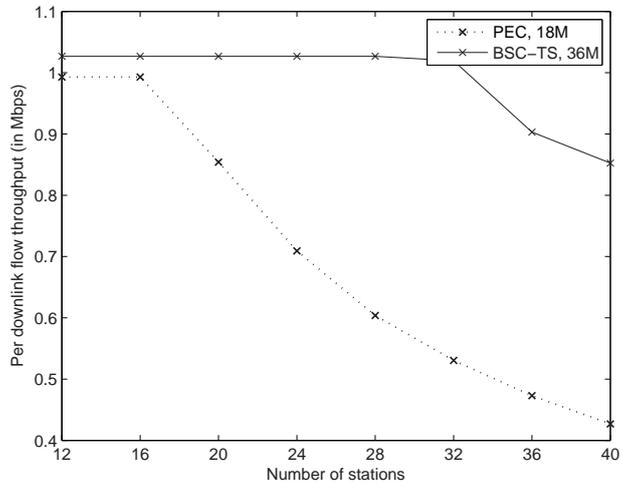
(a) Per-station throughput
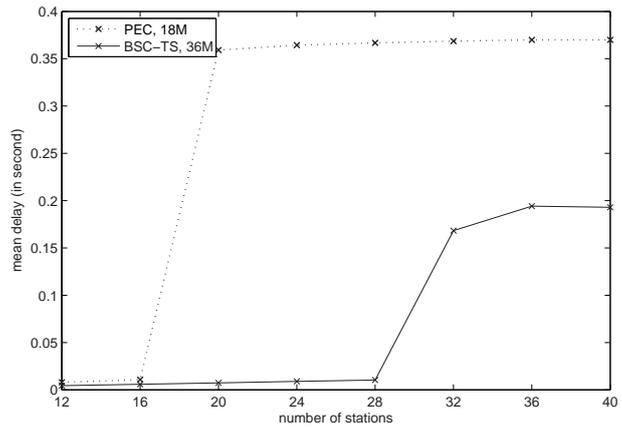
(b) Mean delay

Figure 4.12: Per-station throughput and mean delay vs number of stations, Outdoor measurement channel data, RSSI of class 1 12dBm, RSSI of class 2 35dBm

4.8 Conclusions

In this chapter we consider the potential benefits of viewing the channel provided by an 802.11 WLAN as a binary symmetric channel, as opposed to the conventional packet erasure channel. We present the theoretical analysis of BSC-based superposition coding and a simpler time-sharing coding for multi-destination aggregation in 802.11 WLANs. The performance results for both unicast and multicast traffic, taking account of important MAC layer overheads such as contention time and collision losses, demonstrate that increases in network throughput of more than 100% are possible over a wide range of channel conditions and that the much simpler time-sharing scheme yields most of these gains and have minimal loss of optimality. Importantly, these performance gains involve software rather than hardware changes, and thus essentially come for "free". We also consider the multi-user channel aspect of a WLAN *i.e.* transmissions are inherently broadcast in nature. To our knowledge, this is the first detailed analysis of multi-user BSC coding in 802.11 WLANs.

Chapter 5

Proportional Fair Coding for 802.11 WLANs

5.1 Introduction

Proportional fairness in 802.11 WLANs has been the subject of a considerable body of literature [6, 16, 36, 32, 50]. The CSMA/CA scheduling used in 802.11 differs fundamentally from wired and TDMA wireless networks due to the carrier sense deferral of contention window countdown and the occurrence of collisions, both of which lead to the coupling of station transmissions within a WLAN and the rate region being non-convex. Hence, well established utility fairness techniques from wired and TDMA wireless networks cannot be directly applied to random access CSMA/CA wireless networks.

Recently, [12] provides the first rigorous analysis of proportional fairness in 802.11 WLANs and shows that it is the total airtime (including both successful and colliding transmissions) that is equalised in a proportional fair rate allocation rather than the airtime used for successful transmissions or the TXOP burst duration (both proposed previously). In this chapter we extend that work in a number of different directions. Firstly, the analysis in [12] assumes that transmissions are loss-free while in this chapter we relax this assumption. Secondly, building upon the work in Chapter 3, we view the channel provided by the 802.11 WLAN links as a BSC, as opposed to a more conventional PEC. That is, rather than simply discarding corrupted frames, we consider viewing a received frame as a binary vector in

which an unknown subset of bits have been "flipped". Thirdly, we extend the analysis in [12] to allow delay deadline constraints to be included. Modern networks often contain a mix of traffic types with differing delay and loss sensitivities *e.g.* data, video, voice. An immediate question is how we allocate network resources (airtime etc) between these flows in an effective manner. Current approaches in 802.11 networks largely take a diffserv type of approach (as embodied in 802.11e/802.11n [30, 31]) with ad hoc choice of diffserv traffic parameters, combined with separate MAC layer rate control that takes little account of traffic QoS requirements. The utility optimal joint allocation of airtime and coding rate, the subject of this chapter, allows the throughput/loss/delay trade–off amongst flows sharing network resources to be performed in a principled manner. Our analysis builds on the approach in [52] for TDMA wireless networks.

We show that the joint optimisation of coding rate and airtime decomposes into decoupled allocation tasks, *i.e.* partitioning into layers is optimal. This property of 802.11 differs from and contrasts with the results in [52] for TDMA wireless networks. Further, we establish that the proportional fair coding rate and airtime allocation (i) assigns equal total airtime (*i.e.* airtime including both successful and failed transmissions) to every station in a WLAN, (ii) the station airtimes sum to unity (ensuring operation at the rate region boundary), and (iii) the optimal coding rate is selected to maximise goodput (treating packets decoded after the delay deadline as losses).

5.2   Network Modelling

We consider an 802.11 WLAN with $n \geq 2$ stations. We consider per station throughput fairness and hence define a flow to be a sequence of packets transmitted from the same station. Let $\mathcal{F}$ denote the set of flows, with $|\mathcal{F}| = n$.

5.2.1   BSC Channel

The channel provided by 802.11 frames can be approximately modelled as a BSC (see Chapter 3). That is, at the receiver side rather than simply discarding corrupted frames, we view a decoded frame as a binary vector in which an unknown subset of bits have been "flipped". Let α_f denote the crossover probability of flow f, *i.e.* a bit of flow f is corrupted with probability α_f. For 802.11a/g, experimental measurements presented in Fig. 3.1 indicate that even with a packet erasure rate above 90% typically

only a small fraction (usually $< 10\%$) of bits within corrupted packets are in error. Thus, although noisy, the corrupted packets potentially provide a reasonable channel through which we can transmit information.

5.2.2 MDS Coding

To protect information against channel errors, in addition to the PHY layer convolutional coding provided by the standard 802.11 chipset (see Section 1.1.2), we consider using a block code at the MAC layer for additional forward error correction. An (n, k, d) block code has the following properties. The encoder takes a sequence of k information symbols as input, and generates a sequence of $n \geq k$ coded symbols as output. The decoder takes a sequence of n coded symbols as input, and outputs a sequence of k information symbols. These information symbols will be error-free provided no more than $\lfloor \frac{d-1}{2} \rfloor$ of the coded symbols are corrupted. The Singleton bound [43] tells us that $d \leqslant n - k + 1$, with equality for maximum–distance separable (MDS) codes. Thus, an MDS code can correct up to

$$\left\lfloor \frac{d-1}{2} \right\rfloor = \left\lfloor \frac{n-k}{2} \right\rfloor \tag{5.1}$$

errors. Examples for MDS codes include Reed-Solomon codes [65], and MDS-convolutional codes [54]. In [54], the authors show the existence of MDS–convolutional codes for any code rate. Hereafter, we will make use of Eqn. (5.1), and so, confine consideration to MDS codes. However, the analysis can be readily extended to other types of code provided a corresponding bound on d is available.

5.2.3 Flow Decoding Delay Deadline

An information frame of flow $f \in \mathcal{F}$ is assumed of size k_f symbols. $D_f \in \mathbb{Z}^+$ frames form a block of $D_f k_f$ information symbols. Each block of $D_f k_f$ information symbols is encoded into a block of $D_f l_f$ coded symbols, with the coding rate $r_f = k_f / l_f, 0 < r_f \leq 1$. l_f is the number of coded symbols in a transmitted packet, *i.e.* the transmitted packet size. We assume that the network limits the packet size to be no more than a maximal value $\bar{l}_f$, *e.g.* the 802.11 standard limits the maximum packet size to be 2304B [30]. The quantity D_f is an application supplied QoS parameter. It specifies the decoding delay deadline for flow f, *i.e.* the receiving station must decode an information packet of flow f after collecting at most D_f successive coded packets. Packets decoded after this threshold are treated as losses.

5.2.4 Decoding Error Probability

Associating binary random variable $E_f[i]$ with the i'th coded symbol transmitted by flow f, $E_f[i] = 0$ when the symbol is received correctly, otherwise $E_f[i] = 1$. For an MDS code the probability of a block for flow f being decoded incorrectly is

$$\widetilde{e}_f = \mathbb{P}\left\{ \sum_{i=1}^{D_f l_f} E_f[i] > \frac{D_f l_f - D_f k_f}{2} \right\} \tag{5.2}$$

The random variables $E_f[1], E_f[2], \cdots, E_f[D_f l_f]$ are independent and Bernoulli distributed. Hence the $\sum_{i=1}^{D_f l_f} E_f[i]$ is a binomial random variable, and the probability of a block decoding error can be computed exactly. However, the exact expression is combinatorial in nature, and is not tractable for further analysis. We therefore proceed by obtaining upper and lower bounds on it, and show that the bounds are the same up to a prefactor, and that the prefactor decreases as the block size $D_f l_f$ increases. We hence pose the network utility maximisation problem based on the upper bound on the decoding error probability. Also, we relax the following constraints: $l_f \in \mathbb{Z}_+$ and $k_f \in \mathbb{Z}_+$, and allow them to take positive real values, *i.e.* $l_f \in \mathbb{R}_+$ and $k_f \in \mathbb{R}_+$.

Upper and lower bounds

Lemma 1 (Upper Bound). *The decoding error probability $\widetilde{e}_f$ for flow f satisfies*

$$
\begin{aligned}
\widetilde{e}_f &\leq \exp(-D_f l_f[\theta_f v_f - \log(1 - \beta_f + \beta_f e^{\theta_f})]) \\
&=: e_f(\theta_f, l_f, v_f).
\end{aligned}
\tag{5.3}
$$

where $v_f := \frac{1-r_f}{2}$, $0 \leq v_f < \frac{1}{2}$, $r_f = \frac{k_f}{l_f}$ is the coding rate; $\theta_f > 0$ is the Chernoff–bound parameter; β_f is the symbol error probability. Assuming that a MDS code symbol contains $m \geq 1$ bits, the symbol error probability is $\beta_f = 1 - (1 - \alpha_f)^m$.

Proof.

$$\widetilde{e}_f = \mathbb{P}\left\{ \sum_{i=1}^{D_f l_f} E_f[i] > \frac{D_f l_f - D_f k_f}{2} \right\}$$

$$= \mathbb{P}\left\{ \sum_{i=1}^{D_f l_f} E_f[i] > D_f l_f \left(\frac{1-r_f}{2} \right) \right\}$$

$$\overset{(a)}{\leqslant} \exp\left(-D_f l_f \cdot \theta_f \frac{1-r_f}{2} \right) \mathbb{E}\left[\exp\left(\theta_f \sum_{i=1}^{D_f l_f} E_f[i] \right) \right]$$

$$= \exp\left(-D_f l_f \cdot \theta_f \frac{1-r_f}{2} \right) \left[\mathbb{E}\left[\exp\left(\theta_f E_f[1] \right) \right] \right]^{D_f l_f} \tag{5.4}$$

$$= \exp\left(-D_f l_f \cdot \theta_f \frac{1-r_f}{2} \right) \exp\left(D_f l_f \log\left(\mathbb{E}\left[e^{\theta_f E_f[1]} \right] \right) \right)$$

$$= \exp\left(-D_f l_f \left[\theta_f \frac{1-r_f}{2} - \log\left(\mathbb{E}\left[e^{\theta_f E_f[1]} \right] \right) \right] \right)$$

$$= \exp\left(-D_f l_f \left[\theta_f \frac{1-r_f}{2} - \log\left(1 - \beta_f + \beta_f e^{\theta_f} \right) \right] \right)$$

where inequality (a) follows from the Chernoff bound. $\qquad\square$

Lemma 2 (Lower Bound). *The decoding error probability $\widetilde{e}_f$ for flow f satisfies*

$$\widetilde{e}_f \geq \Gamma \cdot \exp\left(-D_f l_f H(\mathcal{B}(v_f)) \right) \tag{5.5}$$

in which

$$\Gamma = \frac{\beta_f}{1-\beta_f} \exp\left(-D_f l_f I(\mathcal{B}(v_f) \| \mathcal{B}(\beta_f)) \right) \tag{5.6}$$

and $\mathcal{B}(x)$ is the Bernoulli distribution with parameter x, $H(\mathcal{P})$ is the entropy of probability mass function (pmf) $\mathcal{P}$, and $I(\mathcal{P}\|\mathcal{Q})$ is the information divergence between the pmfs $\mathcal{P}$ and $\mathcal{Q}$.

Proof.

$$\widetilde{e}_f = \mathbb{P}\left\{ \sum_{i=1}^{D_f l_f} E_f[i] > \frac{D_f l_f - D_f k_f}{2} \right\}$$

$$= \mathbb{P}\left\{ \sum_{i=1}^{D_f l_f} E_f[i] > D_f l_f \frac{1-r_f}{2} \right\} \tag{5.7}$$

$$= \sum_{i=D_f l_f \frac{1-r_f}{2}+1}^{D_f l_f} \binom{D_f l_f}{i} \beta_f^i (1-\beta_f)^{D_f l_f - i}.$$

The binomial coefficients can be bounded as follows:

$$1 \leqslant \binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{1 \cdot 2 \cdots k} \leqslant n^k. \tag{5.8}$$

Hence,

$$
\begin{aligned}
\widetilde{e}_f &\geqslant \sum_{i=D_f l_f \frac{1-r_f}{2}+1}^{D_f l_f} \beta_f^i (1-\beta_f)^{D_f l_f - i} \\
&\geqslant \frac{\beta_f}{1-\beta_f} \beta_f^{D_f l_f \left(\frac{1-r_f}{2}\right)} (1-\beta_f)^{D_f l_f \left(\frac{1+r_f}{2}\right)} \\
&= \frac{\beta_f}{1-\beta_f} \exp\left(-D_f l_f \frac{1-r_f}{2} \log\left(\frac{1}{\beta_f}\right)\right) \exp\left(-D_f l_f \frac{1+r_f}{2} \log\left(\frac{1}{1-\beta_f}\right)\right) \\
&= \frac{\beta_f}{1-\beta_f} \exp\left(-D_f l_f \frac{1-r_f}{2} \log\left(\frac{1-r_f}{2\beta_f}\right)\right) \exp\left(-D_f l_f \frac{1+r_f}{2} \log\left(\frac{1+r_f}{2(1-\beta_f)}\right)\right) \\
&\quad \exp\left(D_f l_f \frac{1-r_f}{2} \log\left(\frac{1-r_f}{2}\right)\right) \exp\left(D_f l_f \frac{1+r_f}{2} \log\left(\frac{1+r_f}{2}\right)\right) \\
&= \frac{\beta_f}{1-\beta_f} \exp\left(-D_f l_f I\left(\mathcal{B}\left(\frac{1-r_f}{2}\right) \| \mathcal{B}(\beta_f)\right)\right) \exp\left(-D_f l_f H\left(\mathcal{B}\left(\frac{1-r_f}{2}\right)\right)\right)
\end{aligned}
\tag{5.9}
$$

$\square$

Tightness of bounds

Since $\theta_f > 0$ is a free parameter, we can select the optimal θ_f^* that minimises $e_f(\theta_f, l_f, v_f)$ and thus provides the tightest of this upper bound. It can be verified (by inspection of the second derivative) that $e_f(\theta_f, l_f, v_f)$ is convex in θ_f and hence the KKT conditions are necessary and sufficient for an optimum. The KKT condition here is

$$
\frac{\partial e_f(\theta_f, l_f, v_f)}{\partial \theta_f} = e_f(-D_f l_f)\left(v_f - \frac{\beta_f e^{\theta_f}}{1-\beta_f + \beta_f e^{\theta_f}}\right) = 0
\tag{5.10}
$$

which is solved by

$$
\theta_f^* = \log\left(\frac{v_f}{\beta_f}\right) - \log\left(\frac{1-v_f}{1-\beta_f}\right).
\tag{5.11}
$$

provided $v_f > \beta_f$. Substituting for θ_f^*,

$$
\begin{aligned}
\min_{\theta_f > 0} e_f\left(\theta_f, l_f, v_f\right) &= e_f\left(\theta_f^*, l_f, v_f\right) \\
&= \exp\left(-D_f l_f \left(v_f \log\left(\frac{v_f}{\beta_f}\right) + (1-v_f)\log\left(\frac{1-v_f}{1-\beta_f}\right)\right)\right) \\
&= \exp\left(-D_f l_f I\left(\mathcal{B}\left(x_f\right) \| \mathcal{B}(\beta_f)\right)\right)
\end{aligned}
\tag{5.12}
$$

and by Lemmas 1 and 2, the probability $\widetilde{e}_f$ of a decoding error satisfies

$$
\Gamma \cdot e^{-D_f l_f I(\mathcal{B}(v_f) \| \mathcal{B}(\beta_f))} \leq \widetilde{e}_f \leq e^{-D_f l_f I(\mathcal{B}(v_f) \| \mathcal{B}(\beta_f))}.
\tag{5.13}
$$

It can be seen that the upper and lower bounds are the same to within prefactor Γ, and the gap between these bounds decreases exponentially as the block size $D_f l_f$ increases.

5.2.5 Station Throughput

We consider the scenario that the RTS/CTS mechanism is used to make fast recovery from collisions. On a collision only a pair of RTS/CTS packets are sent. Each 802.11 MAC time slot may be a PHY idle slot, a successful transmission or a colliding transmission. Let τ_f denote the probability that the station carrying flow f attempts to transmit in a given MAC slot. The probability that a time slot is idle is

$$P_i = \prod_{f \in \mathcal{F}} (1 - \tau_f) \tag{5.14}$$

The probability that the station carrying flow f makes a successful transmission is

$$P_{s,f} = \tau_f \prod_{g \in \mathcal{F}, g \neq f} (1 - \tau_g) = \frac{\tau_f}{1 - \tau_f} P_i \tag{5.15}$$

The probability that a time slot is a successful transmission is then

$$P_s = \sum_{f \in \mathcal{F}} P_{s,f} = P_i \sum_{f \in \mathcal{F}} \frac{\tau_f}{1 - \tau_f} \tag{5.16}$$

The throughput of flow f is given by

$$s_f = \frac{P_{s,f} l_f}{\sigma P_i + \sum_{f \in \mathcal{F}} T_{s,f} P_{s,f} + T_c (1 - P_i - P_s)} \tag{5.17}$$

in which σ is the duration of a PHY idle slot; $T_{s,f}$ is the duration of a successful transmission by flow f; $T_c = T_{RTS} + T_{SIFS} + T_{CTS} + T_{DIFS}$ is the duration of a collision. Note that T_c is the same for all flows due to the use of RTS/CTS handshaking but we do not assume that flows use the same packet successful transmission duration $T_{s,f}$. We relate packet successful transmission duration $T_{s,f}$ to the packet size l_f via the PHY rate w_f of flow f. Namely, $T_{s,f} = l_f / w_f + T_o$ where T_o is the 802.11 protocol overhead associated with each successful transmission, *i.e.* $T_o = T_{RTS} + T_{CTS} + 3 \times T_{SIFS} + T_{PHYhdr} + T_{ACK} + T_{DIFS}$.

Letting $x_f = \tau_f / (1 - \tau_f)$ the throughput expression can be rewritten as

$$s_f = \frac{x_f l_f}{X \cdot T_c} \tag{5.18}$$

in which

$$X = \frac{\sigma}{T_c} + \sum_{f \in \mathcal{F}} \left(\frac{l_f}{w_f T_c} + \frac{T_o}{T_c} - 1 \right) x_f + \prod_{f \in \mathcal{F}} (1 + x_f) - 1 \tag{5.19}$$

When all flows use the same packet size and PHY rate, throughput expression (5.18) is identical to that previously used in [35, 12].

5.3 Proportional Fair Allocation

5.3.1 Non-convex Network Utility Optimisation Problem

Define:

- $\boldsymbol{l} := [l_f]_{f \in \mathcal{F}}$, the vector of packet sizes

- $\boldsymbol{x} := [x_f]_{f \in \mathcal{F}}$, the vector of station attempt rate parameters

- $\boldsymbol{\theta} := [\theta_f]_{f \in \mathcal{F}}$, the vector of Chernoff parameters

- $\boldsymbol{v} := [v_f]_{f \in \mathcal{F}}$, the vector of coding rate parameters

For proportional fairness, we select the sum of the log of the flow goodputs as our network utility U. The proportional fair rate allocation problem is defined as

$$\max_{\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{\theta}} \quad U(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{\theta}) := \sum_{f \in \mathcal{F}} \log g_f$$

$$\text{s.t.} \quad 0 < l_f \leq \bar{l}_f, \ \theta_f > 0, \ x_f > 0, \ 0 \leq v_f < 1/2 \qquad \forall f \in \mathcal{F}$$

(5.20)

in which

$$g_f = s_f (1 - 2v_f)(1 - e_f(l_f, v_f, \theta_f))$$

(5.21)

is the goodput of flow $f \in \mathcal{F}$ after decoding.

The constraints are convex, however it can be verified by inspection of the second derivative that $\log(1 - e_f(l_f, v_f, \theta_f))$ is not jointly concave in (θ_f, l_f, v_f) and hence the maximisation problem is not a standard convex optimisation task.

5.3.2 Reformulation as Sequential Optimisations

We proceed by making the following key observation.

Lemma 3. *For convex sets $\mathcal{Y}$ and $\mathcal{Z}$, and for a function $f : \mathcal{Y} \times \mathcal{Z} \rightarrow \mathbb{R}$ that is concave in $y \in \mathcal{Y}$ and in $z \in \mathcal{Z}$, but not jointly in (y, z), the solution to the joint optimisation problem*

$$\max_{y \in \mathcal{Y}, z \in \mathcal{Z}} f(y, z)$$

(5.22)

85

is unique, and is the same as the solution to

$$\max_{z \in \mathcal{Z}} \max_{y \in \mathcal{Y}} f(y, z) \tag{5.23}$$

if $f(y^*(z), z)$ is a concave function of z, where for each $z \in \mathcal{Z}$, $y^*(z) := \arg\max_{y \in \mathcal{Y}} f(y, z)$.

Proof.

For any $z \in \mathcal{Z}$, the function $f(y, z)$ is concave in y. Hence, for each z, there exists a unique maximum $y^+(z)$, which is given by

$$f(y^+(z), z) = \max_{y \in \mathcal{Y}} f(y, z) =: g(z) \tag{5.24}$$

If $f(y^+(z), z)$ is a concave function of z, then there exists a unique maximiser, which is denoted by z^+, *i.e.*

$$z^+ = \arg\max_{z \in \mathcal{Z}} f(y^+(z), z). \tag{5.25}$$

We show that $(y^+(z^+), z^+)$ is an optimum solution to Eqn. (5.22). Since z^+ is the maximiser of g, we have for any $z \in \mathcal{Z}$,

$$
\begin{aligned}
g(z^+) &\geqslant g(z) \\
\text{or} \quad f(y^+(z^+), z^+) &\geqslant f(y^+(z), z)
\end{aligned}
\tag{5.26}
$$

For any given $z \in \mathcal{Z}$, $y^+(z)$ is the maximiser of $f(y, z)$ over all $y \in \mathcal{Y}$, *i.e.*

$$f(y^+(z), z) \geqslant f(y, z) \tag{5.27}$$

and hence, for all $(y, z) \in \mathcal{Y} \times \mathcal{Z}$,

$$f(y^+(z^+), z^+) \geqslant f(y^+(z), z) \geqslant f(y, z). \tag{5.28}$$

We note that $y^+(\cdot)$ maps $\mathcal{Z}$ into $\mathcal{Y}$, and hence, $(y^+(z^+), z^+) \in \mathcal{Y} \times \mathcal{Z}$. Hence, $(y^+(z^+), z^+)$ is a global maximiser. $\square$

This lemma establishes conditions under which a non-convex optimisation can be transformed into a sequence of convex optimisations. Hence, we carry out the non-convex optimisation sequentially *i.e.* first finding $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} U(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{\theta})$ and then $\max_{\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{x}} U(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{\theta}^*)$. Provided $U(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{\theta})$ is concave in $\boldsymbol{\theta}$ and $U(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{\theta}^*)$ is jointly concave in $(\log \boldsymbol{l}, \log \boldsymbol{v}, \log \boldsymbol{x})$ (see Lemma 4), it is easy to see that we have a sequence of convex optimisations which are equivalent to the original optimisation.

5.3.3 Optimal Chernoff Parameter θ_f^*

We begin by considering the optimisation

$$\max_{\boldsymbol{\theta}} U(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{\theta}) \qquad \text{s. t.} \quad \theta_f > 0 \quad \forall f \in \mathcal{F} \tag{5.29}$$

It can be verified that U is concave in $\boldsymbol{\theta}$ (U is separable in the θ_f's and it can be verified by inspection of the second derivative that U is concave for each θ_f). Hence, the problem is a convex optimisation problem and from the KKT conditions the optimal θ_f is

$$\theta_f^*(v_f) = \log\left(\frac{v_f}{\beta_f}\right) - \log\left(\frac{1 - v_f}{1 - \beta_f}\right) \tag{5.30}$$

Observe that when $v_f \leq \beta_f$, $\theta_f \leq 0$. To ensure $\theta_f > 0$, it is required that $v_f > \beta_f$. The optimal θ_f^* guarantees that the optimisation establishes upon the tightest upper bound on the block decoding error probability.

5.3.4 Optimal Airtime, Coding Rate and Transmit Rate l_f^*, v_f^*, x_f^*

We now solve

$$\max_{\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{x}} \quad U(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{\theta}^*(\boldsymbol{v}))$$
$$\text{s.t} \quad 0 < l_f \leq \bar{l}_f, \ x_f > 0, \ \beta_f < v_f < 1/2 \quad \forall f \in \mathcal{F} \tag{5.31}$$

Unfortunately $U(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{\theta}^*)$ is not jointly concave in $\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{x}$. We therefore change to work in terms of log-transformed variables $\tilde{l}_f = \log l_f$, $\tilde{I}_f(v_f) = \log I_f(v_f)$ and $\tilde{x}_f = \log x_f$, in which

$$I_f(v_f) = I\left(\mathcal{B}(v_f) \parallel \mathcal{B}(\beta_f)\right) = v_f \log\left(\frac{v_f}{\beta_f}\right) + (1 - v_f) \log\left(\frac{1 - v_f}{1 - \beta_f}\right) \tag{5.32}$$

Since $\tilde{I}_f(v_f)$ is a monotone increasing function of v_f (verified by inspection of first derivative), the inverse mapping from $\tilde{I}_f$ to v_f exists and is one-to-one, that is, we can work interchangeably with either v_f or $\tilde{I}_f$. With the obvious abuse of notation we use $v_f(\tilde{I}_f)$ to denote the v_f value corresponding to a value of $\tilde{I}_f$. Note that I_f arises naturally in our problem since $e_f(l_f, v_f, \theta_f^*(v_f)) = e^{-D_f l_f I_f(v_f)}$. The optimisation becomes

$$\max_{\tilde{\boldsymbol{l}}, \tilde{\boldsymbol{I}}, \tilde{\boldsymbol{x}}} \quad U_1(\tilde{\boldsymbol{l}}, \tilde{\boldsymbol{I}}, \tilde{\boldsymbol{x}}) := U(e^{\tilde{\boldsymbol{l}}}, \boldsymbol{v}(\tilde{\boldsymbol{I}}), e^{\tilde{\boldsymbol{x}}}, \boldsymbol{\theta}^*(\boldsymbol{v}(\tilde{\boldsymbol{I}})))$$
$$\text{s.t} \quad \tilde{l}_f \leq \log \bar{l}_f, \ \beta_f < v_f(\tilde{I}_f) < 1/2 \quad \forall f \in \mathcal{F} \tag{5.33}$$

Importantly, we have the following lemma:

Lemma 4. $U_1(\tilde{\boldsymbol{l}}, \tilde{\boldsymbol{I}}, \tilde{\boldsymbol{x}})$ *is jointly concave in* $\tilde{\boldsymbol{l}}, \tilde{\boldsymbol{I}}, \tilde{\boldsymbol{x}}$.

Proof.

$$
\begin{aligned}
U_1(\tilde{\boldsymbol{l}}, \tilde{\boldsymbol{I}}, \tilde{\boldsymbol{x}}) &= \sum_{f \in \mathcal{F}} \log s_f + \log(1 - 2v_f) + \log(1 - e_f) \\
&= \sum_{f \in \mathcal{F}} \tilde{x}_f - \log X(\tilde{\boldsymbol{l}}, \tilde{\boldsymbol{x}}) + \tilde{l}_f - \log T_c + \log(1 - 2v_f(\tilde{I}_f)) + \log(1 - e^{-D_f e^{\tilde{l}_f + \tilde{I}_f}})
\end{aligned}
\tag{5.34}
$$

in which

$$
\begin{aligned}
X(\tilde{\boldsymbol{l}}, \tilde{\boldsymbol{x}}) &= \frac{\sigma}{T_c} + \sum_{f \in \mathcal{F}} \left(\frac{e^{\tilde{l}}}{T_c w_f} + \frac{T_o}{T_c} - 1 \right) e^{\tilde{x}_f} + \prod_{f \in \mathcal{F}} \left(1 + e^{\tilde{x}_f} \right) - 1 \\
&= \frac{\sigma}{T_c} + \sum_{f \in \mathcal{F}} \frac{e^{\tilde{l}} e^{\tilde{x}_f}}{T_c w_f} + \frac{T_o}{T_c} \sum_{f \in \mathcal{F}} e^{\tilde{x}_f} + \prod_{f \in \mathcal{F}} \left(1 + e^{\tilde{x}_f} \right) - 1 - \sum_{f \in \mathcal{F}} e^{\tilde{x}_f} \\
&= \frac{\sigma}{T_c} + \frac{1}{T_c w_f} \sum_{f \in \mathcal{F}} e^{\tilde{l}} e^{\tilde{x}_f} + \frac{T_o}{T_c} \sum_{f \in \mathcal{F}} e^{\tilde{x}_f} + \sum_{k=2}^{N} \sum_{A \subseteq \mathcal{F}, |A|=k} \prod_{j \in A} e^{\tilde{x}_j}
\end{aligned}
\tag{5.35}
$$

The logarithm of a sum of exponentials is a convex function [10], and thus $\log X$ is convex in the transformed variables $\tilde{\boldsymbol{l}}, \tilde{\boldsymbol{x}}$.

U_1 is separable in $(\tilde{l}_f, \tilde{I}_f)$, and now we show that the last two terms $\log(1 - 2v_f(\tilde{I}_f)) + \log(1 - e^{-D_f e^{\tilde{l}_f + \tilde{I}_f}})$ are jointly concave in $(\tilde{l}_f, \tilde{I}_f)$.

Since, for $v_f \in (\beta_f, 0.5)$, I_f is a monotone function of v_f, and $\tilde{I}_f$ is a monotone function of I_f, it is clear that $\tilde{I}_f$ is invertible. Note that

$$
\begin{aligned}
\tilde{I}_f &= \log(I_f) \\
\frac{d\tilde{I}_f}{dv_f} &= \frac{dI_f}{dv_f} \cdot \frac{1}{I_f} = \frac{\theta_f^*(v_f)}{I_f} \\
\frac{dv_f}{d\tilde{I}_f} &= \frac{I_f}{\theta_f^*(v_f)} \\
\frac{d^2 v_f}{d\tilde{I}_f^2} &= \frac{I_f}{\theta_f^*(v_f)} \left[1 - \frac{I_f}{\theta_f^*(v_f)^2} \frac{1}{v_f(1 - v_f)} \right]
\end{aligned}
\tag{5.36}
$$

Define $y(v_f) := v_f(1 - v_f)\theta_f^*(v_f)^2 - I_f$. Note that $y'(v_f) = (1 - 2v_f)\theta_f^*(v_f)^2 + \theta_f^*(v_f) > 0$, $y(v_f)$ is thus increasing with v_f, and $y(v_f) > y(\beta_f) = 0$. Hence, $\frac{d^2 v_f}{d\tilde{I}_f^2} > 0$ and $v_f(\tilde{I}_f)$ is strictly convex.

88

Define $h(x, y) = e^{x+y}$. Consider the function

$$f(x, y) = \log(1 - e^{-D_f h(x,y)})$$

$$\frac{\partial f}{\partial x} = \frac{D_f e^{-D_f h}}{1 - e^{-D_f h}} \frac{\partial h}{\partial x}$$

$$\frac{\partial f}{\partial x} = \frac{D_f h e^{-D_f h}}{1 - e^{-D_f h}}.$$

$$\text{Similarly,} \quad \frac{\partial f}{\partial y} = \frac{D_f h e^{-D_f h}}{1 - e^{-D_f h}}.$$

(5.37)

Also,

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{-D_f^2 h^2 e^{-2D_f h}}{(1 - e^{-D_f h})^2} + \frac{D_f h e^{-D_f h}}{1 - e^{-D_f h}} - \frac{-D_f^2 h^2 e^{-D_f h}}{1 - e^{-D_f h}}$$

$$= \frac{-D_f h}{(1 - e^{-D_f h})^2} \cdot \left[D_f h e^{-2D_f h} + D_f h (1 - e^{-D_f h}) e^{-D_f h} - (1 - e^{-D_f h}) e^{-D_f h} \right]$$

$$= \frac{-D_f h}{(1 - e^{-D_f h})^2} \left[D_f h - (1 - e^{-D_f h}) e^{-D_f h} \right]$$

(5.38)

Similarly, one can show that $\frac{\partial^2 f}{\partial x^2} = \frac{\partial^2 f}{\partial y^2} = \frac{\partial^2 f}{\partial x \partial y}$. Define $\ell(x) = D_f x - (1 - e^{-D_f x}) e^{-D_f x}$. If $\ell(x) > 0$, then $f(x, y)$ is (strictly) concave. Note that $\ell'(x) = D_f(1 - e^{-2D_f x}) + D_f e^{-D_f x}(1 - e^{-D_f x}) > 0$ when $x > 0$. Hence, $\ell(x) > \ell(0) = 0$. Therefore, $\log(1 - e^{-D_f e^{\tilde{l}_f + \tilde{I}_f}})$ is jointly concave in $\tilde{l}_f, \tilde{I}_f$ and $\log(1 - 2v_f(\tilde{I}_f))$ is concave in $\tilde{I}_f$. $\qquad \square$

By Lemma 4, the transformed maximisation problem is a convex optimisation problem. The Lagrangian is

$$L = U_1 - \sum_{f \in \mathcal{F}} p_f \left(\tilde{l}_f - \log \bar{l}_f \right)$$

(5.39)

where multiplier $p_f \geq 0$ and we have dropped terms involving the constraints on v_f in order to streamline notation and because these are almost never active in practical cases. The KKT optimality conditions corresponding to $\tilde{x}_f$, $\tilde{I}_f$ and $\tilde{l}_f$ are respectively

$$\frac{nx_f}{X} \left(\frac{l_f}{T_c w_f} + \frac{T_o}{T_c} - 1 + \prod_{g \in \mathcal{F}, g \neq f} (1 + x_g) \right) = 1$$

(5.40)

$$\frac{2}{1 - 2v_f} = \frac{e_f(l_f, v_f)}{1 - e_f(l_f, v_f)} D_f l_f \theta_f^*(v_f)$$

(5.41)

$$1 - \frac{nx_f}{X} \frac{l_f}{T_c w_f} + \frac{e_f(l_f, v_f)}{1 - e_f(l_f, v_f)} D_f l_f I_f(v_f) - p_f = 0$$

(5.42)

89

Optimal packet size l_f^* and coding rate v_f^*

Combining Eqns. (5.40), (5.41) and (5.42), the optimal solution satisfies

$$\left(1 + \frac{2I_f}{(1-2v_f)\theta_f} - p_f \right) \times \left(1 + \left(\frac{T_o}{T_c} - 1 + \prod_{g \in \mathcal{F}, g \neq f} (1 + x_g) \right) \frac{w_f T_c}{l_f} \right) = 1 \qquad (5.43)$$

Note that $\left(\frac{T_o}{T_c} - 1 + \prod_{g \in \mathcal{F}, g \neq f} (1 + x_g) \right) \frac{w_f T_c}{l_f} > 0$ and $\frac{2I_f}{(1-2v_f)\theta_f} > 0$. Hence at the optimum we must have $p_f > 0$. By complementary slackness, the constraint on the flow packet size must therefore be tight, i.e. the optimal packet size is

$$l_f^* = \bar{l}_f \qquad (5.44)$$

The optimal coding rate v_f^* can now be found by solving (5.41).

Optimal transmission attempt parameter x_f^*

It remains to determine the optimal station transmission rate parameter x_f^*. This can be found using standard subgradient descent techniques, e.g. using the algorithm detailed in Alg. 1.

Algorithm 1 Calculate optimal $\boldsymbol{x}$

1: Initialise $\boldsymbol{x}^{(1)} = [x_1^{(1)}, x_2^{(1)}, \cdots, x_n^{(1)}]$, $t = 0$.

2: **repeat**

3:    $t = t + 1$; $a^{(t)} = 1/t^2$

4:    $\forall f \in \mathcal{F}$, calculate

$$y_f^{(t)} = 1 - \frac{n x_f^{(t)}}{X(\bar{\boldsymbol{l}}, \boldsymbol{v}^*)} \left(\frac{\bar{l}_f}{T_c w_f} + \frac{T_o}{T_c} - 1 + \prod_{g \in \mathcal{F}, g \neq f} (1 + x_g^{(t)}) \right)$$

$$x_f^{(t+1)} = \left(x_f^{(t)} + a^{(t)} y_f^{(t)} \right)^+$$

5: **until** $|y_f^{(t)}| \leq \epsilon \; \forall f \in \mathcal{F}$, where $\epsilon > 0$.

5.4   Discussion

5.4.1   Equal Air-time

Due to the presence of collision losses and the coupling of station transmissions via carrier sense, the flow air-time in a WLAN is not simply the successful transmission duration but also includes airtime expended in collisions. Following [12], define the *flow total air-time* as the fraction of time used for transmissions by flow f, including both successful transmissions and collisions, which is given by

$$t_f = \frac{P_{s,f}T_{s,f} + (1 - P_{s,f})T_c}{\sigma P_i + \sum_{f \in \mathcal{F}} T_{s,f}P_{s,f} + T_c(1 - P_i - P_s)}$$
$$= \frac{1}{X}\left(\frac{\tau_f}{P_i} + x_f\left(\frac{T_{s,f}}{T_c} - 1\right)\right) \tag{5.45}$$

KKT condition (5.40) can be rewritten as

$$\frac{1}{X}\left(\frac{\tau_f}{P_i} + x_f\left(\frac{T_{s,f}}{T_c} - 1\right)\right) = \frac{1}{n} \tag{5.46}$$

Hence, the proportional fair allocation assigns equal flow total air-times amongst flows, and the flow total air-times sum to unity. It is worth pointing out that since the flow air-time usage overlaps due to collisions, the flow total air-times summing to unity does not imply that the channel idle probability $P_i = 0$.

5.4.2   Decoupled Allocation Tasks

It is the packet size l_f that acts to couple the flow coding rate and transmission rate in the optimisation objective U_1,

$$U_1(\tilde{\boldsymbol{l}}, \tilde{\boldsymbol{I}}, \tilde{\boldsymbol{x}}) = \sum_{f \in \mathcal{F}} \underbrace{\tilde{x}_f - \log X(\tilde{\boldsymbol{l}}, \tilde{\boldsymbol{x}}) + \tilde{l}_f}_{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{l}}} - \log T_c + \underbrace{\log(1 - 2v_f(\tilde{\boldsymbol{I}})) + \log(1 - e^{-D_f e^{\tilde{l}_f + \tilde{I}_f}})}_{\tilde{\boldsymbol{I}}, \tilde{\boldsymbol{l}}} \tag{5.47}$$

Our analysis establishes that the optimal choice is to select the maximum possible packet size $l_f^* = \bar{l}_f$. Substituting for this known value, U_1 separates into a term in $\tilde{\boldsymbol{x}}$ and a term in $\tilde{\boldsymbol{I}}$. That is, the optimisation problem decomposes into decoupled flow transmission rate and coding rate allocation tasks. Inspection of KKT condition (5.41) reveals that the optimal $\boldsymbol{I}$ is dependent on channel condition parameter $\boldsymbol{\beta}$ and delay requirement $\boldsymbol{D}$, while from KKT condition (5.40) it can be seen that the

optimal $\boldsymbol{x}$ is dependent on PHY rate $\boldsymbol{w}$ and the number of stations n in the WLAN. Therefore a layered approach that separates MAC scheduling and packet coding rate selection is optimal.

5.4.3   Decentralised 802.11 Implementation

The optimal l_f^*, v_f^*, x_f^* can be determined by the station carrying flow f without the need for any message passing. Solving (5.41) to find optimal v_f^* requires only local information which is available at each station. In Alg. 1 observe that (i) the flow throughput s_f can be measured by station f, and the quantity X can then be computed based on (5.18); (ii) the channel idle probability P_i can be observed by all stations in a WLAN (via carrier sense, *e.g.* [45]), and hence we have the quantity $\prod_{g \in \mathcal{F}, g \neq f}(1 + x_g) = \frac{1}{P_i} \cdot \frac{1}{1+x_f}$. Therefore, the update of x_f in Alg. 1 can be carried out locally at each station.

5.5   Examples

5.5.1   Impact of BER

Consider a WLAN with two stations with equal transmitted packet sizes $\bar{l}_1 = \bar{l}_2$ and PHY rates $w_1 = w_2$. The proportional fair allocation assigns equal attempt probabilities to both stations. Fig. 5.1 plots the optimal coding rates for both flows as the symbol error rate β_2 of flow 2 is varied while keeping β_1 of flow 1 fixed. It can be seen that when the flows have the same delay deadlines, the flow with poorer channel conditions is allocated with a lower coding rate. When the delay deadline for flow 2 is increased to infinity, corresponding to delay-insensitive applications, the coding rate of flow 2 increases. That is, a tighter delay deadline comes at the cost of using a lower coding rate and so lower throughput – this is to be expected since delay insensitive flows can use a larger code block size and so gain more efficient error protection. It can be also seen that changing the symbol error rate or delay threshold for flow 2 does not affect the coding rate of flow 1. This is because the optimal coding rate for a flow is only dependent on its own symbol error rate and delay deadline.
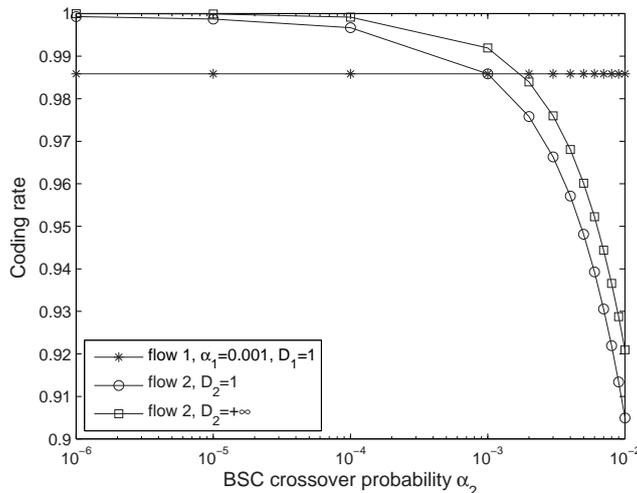
Figure 5.1: Single WLAN with 2 flows, packet size $\bar{l}_1 = \bar{l}_2 = 8000$ bits, PHY rates $w_1 = w_2 = 54$Mbps

5.5.2 Impact of Packet Size

Now consider a WLAN with 6 flows which have identical delay deadlines, symbols error rates and PHY rates, but each flow has a different maximum transmitted packet length. Fig. 5.2(a) plots the optimal station attempt probability versus the successful transmission durations $T_{s,f}$. Fig. 5.2(b) plots the corresponding successful flow air-times and flow total air-times. It can be seen that the proportional fair allocation assigns equal flow total air-times but unequal successful flow air-times. Although flows with longer successful transmission durations are allocated with lower attempt probabilities, they still have longer successful flow air-times. The optimal coding rate for each flow is also different, since the different packet sizes yield different code block sizes (recall the delay deadlines are the same for all flows). That is, neither the optimal coding rate nor the optimal attempt probability is the same for each flow. Similar behaviour occurs when the packet sizes for flows are the same but the PHY rates w_f differ.

5.6 Conclusions

We derive the joint allocation of coding rate and airtime that achieves the proportionally fair goodput allocation in an 802.11 WLAN. We consider BSC lossy links and take the delay constraint into account.
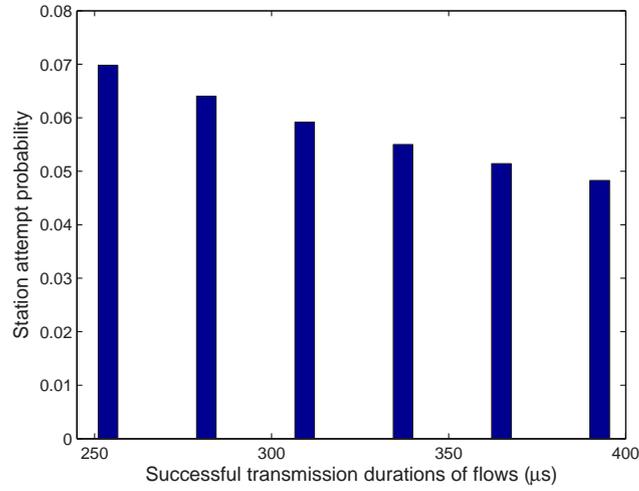
(a) Station attempt probability
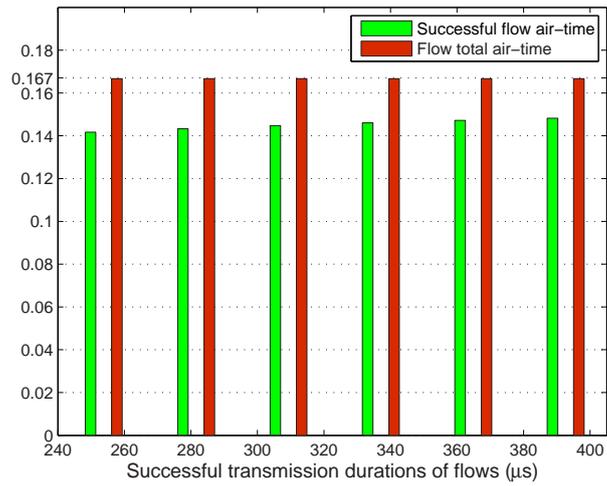
(b) Air-time

Figure 5.2: Single WLAN with 6 flows. For each flow, the PHY rate $w = 54$Mbps, delay deadline $D = 1$ and BSC crossover probability $\alpha = 10^{-3}$.

We show that the joint optimisation decomposes into decoupled allocation tasks and (i) assigns equal total airtime to stations, (ii) selects the station transmission attempt probabilities such that the airtimes sum to unity, (iii) selects the optimal coding rate to maximise goodput (treating packets decoded after the delay deadline as losses).

Chapter 6

Conclusions

This thesis discusses cross-layer coding design in 802.11 WLANs. We are interested in the tradeoffs between higher-layer coding rate and PHY layer coding/MAC layer transmission scheduling.

We begin by considering the joint performance of higher-layer fountain coding concatenated with 802.11a/g OFDM PHY modulation/coding in a PEC paradigm in Chapter 2. We are interested in the optimal choice of PHY modulation/coding rates with and without higher layer fountain coding for standard 802.11 multicast WLANs. We find that, in contrast to studies in cellular networks, in 802.11a/g WLANs the PHY modulation/rate that optimizes uncoded multicast performance is also close to optimal for fountain-coded multicast traffic, which indicates that in 802.11a/g WLANs cross-layer rate control for higher-layer fountain coding concatenated with physical layer modulation and FEC would bring few benefits. This is principally because the current 802.11a/g WLANs do not provide a rich set of PHY rates to support the mutlicast network always operating at low PER regions. Although our conclusion is "negative", it is important and interesting for future research on the use of fountain codes in wifi-like systems, especially on the multicast network setting and the content delivery objective. Instead of the standard multicast setting considered in this thesis, there are some more efficient and interesting ways to make good use of fountain coding in 802.11 multicast networks, e.g. in a heterogenous multicast network traffic can be delivered in an ad-hoc fashion, that is, clients which finish decoding earlier can forward coded packets to the remaining clients at reasonable PHY rates.

In Chapter 3 we take experimental measurements for outdoor 802.11 links and demonstrate that the channel provided by corrupted frames alone can be accurately modelled as a BSC provided appropriate pre- and post- processing, and the channel provided by corrupted frames and other frames combined can be accurately modelled as a hybrid binary symmetric/packet erasure channel. Importantly, this hybrid channel offers capacity increases of more than 100% compared to a conventional PEC over a wide range of RSSIs. This is a significant observation as it indicates that in current 802.11 hardware there is a great deal of useful information wasted in corrupted frames because of the package erasure paradigm. Techniques of exploiting the information contained in corrupted frames are required. In this thesis measurements were taken only in an outdoor environment. We can expect that due to the multipath effects caused by indoor obstacles and increased interferences, the channel provided by indoor 802.11 links can be fast time-varying and have more complicated statistical properties. We leave the channel modelling for indoor 802.11 links as future work.

In Chapter 4 we give a method of making use of the capacity increase provided by the hybrid channel. We consider the potential performance gains of adopting a BSC paradigm for multi-destination aggregations in 802.11 WLANs. Two BSC-based higher-layer coding approaches, *i.e.* superposition coding and a simpler time-sharing coding, are considered for multi-destination aggregated packets. The performance results for both unicast and multicast traffic, taking account of important MAC layer overheads, demonstrate that increases in network throughput of more than 100% are possible over a wide range of channel conditions, and that the simpler time-sharing approach yields most of these gains and has minimal loss of optimality. Importantly, these performance gains involve software rather than hardware changes, and thus essentially come for "free". Not only for mutli-destination aggregated packets, in a BSC paradigm normal packets can also be protected with use of an additional coding scheme, and the throughput will also increase.

In Chapter 5 we consider the fairness issue in the design of cross-layer coding. We are interested in the proportional fair allocation of high-layer coding rates and MAC airtimes in a single-hop 802.11 WLAN. The analysis assumes BSC lossy links and takes the delay constraint into consideration. We show that the joint optimisation of higher-layer coding rate and airtime decomposes into decoupled allocation tasks, *i.e.* a layered approach of separating MAC scheduling and higher-layer coding rate selection is optimal. This property of 802.11 differs from and contrasts with TDMA wireless networks. Further, we establish that the proportional fair coding rate and airtime allocation (i) assigns equal total airtime to every station in a WLAN, (ii) the station airtimes sum to unity, and (iii) the optimal coding rate is selected to maximise goodput. The analysis can be extended to the per-flow fairness

in a single-hop WLAN, *i.e.* flows transmitted from the same station with different destinations have individual optimal coding rates, and further, the per-flow fairness in mesh wireless networks.

Bibliography

[1] Least squares fitting. http://mathworld.wolfram.com/LeastSquaresFitting.html/.

[2] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[3] Relay (formerly relayfs). http://relayfs.sourceforge.net/.

[4] V. Badarla, V. G. Subramanian, and D. J. Leith. Low-delay dynamic routing using fountain codes. *IEEE Communications Letters*, 13(7):552–554, 2009.

[5] A. R. Bahai and B. R. Saltzberg. *Multi-Carrier Digital Communications: Theory and Applications of OFDM*. Plenum Publishing Corp., 1999.

[6] A. Banchs, P. Serrano, and H. Oliver. Proportional fair throughput allocation in multirate IEEE 802.11e wireless LANs. *Wireless Networks*, 13(5):649C662, 2006.

[7] C. R. Berger, S. Zhou, Y. Wen, K. Pattipati, and P. Willett. Optimizing joint erasure- and error-correction coding for wireless packet transmissions. *IEEE Transaction on Wireless Communications*, 7(11):4586–4595, 2008.

[8] P. P. Bergmans and T. M. Cover. Cooperative broadcasting. *IEEE Transactions on Information Theory*, 20(3):317–324, 1974.

[9] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–548, 2000.

[10] S. Boyd and L. Vandenberghe. *Convex Optimisation*. Cambridge University Press, 2008.

[11] J. W. Byers, M. Luby, and M. Mitzenmacher. A digital fountain approach to asynchronous reliable multicast. *IEEE Journal on Selected Areas in Communications*, 20(8):1528–1540, 2002.

[12] A. Checco and D. J. Leith. Proportional fairness in 802.11 wireless LANs. *IEEE Communications Letters*, 15:807–809, 2011.

[13] J. Conan. The weight spectra of some short low-rate convolutional codes. *IEEE Transaction on Communnications*, 32(9):1050–1053, 1984.

[14] T. A. Courtade and R. D. Wesel. A cross-layer perspective on rateless coding for wireless channels. In *Proc. of IEEE International Conference on Communications (ICC 2009)*, pages 1–6, 2009.

[15] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006.

[16] J. Dunn, M. Neufeld, A. Sheth, D. Grunwald, and J. Bennett. A practical cross-layer mechanism for fairness in 802.11 networks. In *Proc. of the 1st Annual International Conference on Broadband Networks (BroadNets 2004)*, pages 355 – 364, 2004.

[17] A. Dutta, D. Saha, D. Grunwald, and D. Sicker. SMACK - a smart acknowledgment scheme for broadcast messages in wireless networks. In *Proc. of the ACM SIGCOMM 2009 conference on Data communication*, pages 15–26, 2008.

[18] P. Fuxjager and F. Ricciato. Collecting broken frames: Error statistics in IEEE 802.11b/g links. In *Proc. of the 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (Wiopt 2008)*, pages 30–35, 2008.

[19] M. S. Gast. *802.11 Wireless networks*. O'Reilly, 2002.

[20] D. Giustiniano, D. Malone, D. J. Leith, and K. Papagiannaki. Measuring transmission opportunities in 802.11 links. *IEEE/ACM Transactions on Networking*, 18(5):1516–1529.

[21] A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[22] D. Haccoun and G. Bégin. High-rate punctured convolutional codes for Viterbi and sequential decoding. *IEEE Transaction on Communnications*, 37(11):1113 – 1125, 1989.

[23] B. Han, L. Ji, S. Lee, B. Bhattacharjee, and R. R. Miller. All bits are not equal - a study of IEEE 802.11 communication bit errors. In *Proc. of the 28th Conference on Computer Communications (IEEE INFOCOM 2009)*, pages 1602–1610, 2009.

[24] K. D. Huang, K. R. Duffy, and D. Malone. On the validity of IEEE 802.11 MAC modeling hypotheses. *IEEE/ACM Transactions on Networking*, 18(6):1935–1948.

[25] IEEE. *Wirless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE std 802.11-1997 edition, 1997.

[26] IEEE. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications High-speed Physical Layer in the 5 GHz Band*, IEEE std 802.11a-1999 edition, 1999.

[27] IEEE. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, IEEE std 802.11b-1999 edition, 1999.

[28] IEEE. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band*, IEEE std 802.11g-2003 edition, 2003.

[29] IEEE. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, IEEE std 802.11e edition, 2005.

[30] IEEE. *IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE std 802.11-2007 edition, 2007.

[31] IEEE. *IEEE 802.11n-2009 - Amendment 5: Enhancements for Higher Throughput*, IEEE std 802.11n edition, 2009.

[32] L. B. Jiang and S. C. Liew. Proportional fairness in wireless LANs and ad hoc networks. In *Proc. of IEEE Wireless Communications and Networking Conference 2005*, page 1551C1556, 2005.

[33] J. Lassing, T. Ottosson, and E. Strom. On the union bound applied to convolutional codes. In *Proc. of the 54th IEEE Vehicular Technology Conference*, pages 2429–2433, 2001.

[34] S. Lawrence, A. Biswas, and A. A. Sahib. A comparative analysis of VoIP support for HT transmission mechanisms in WLAN. In *Proc. of the 27th International Conference on Distributed Computing Systems Workshops (ICDCSW '07)*, 2007.

[35] D. Leith, V. G. Subramanian, and K. Duffy. Log-convexity of rate region in 802.11e WLANs. *IEEE Communications Letters*, 14(1):57–59, 2010.

[36] L. Li, M. Pal, and Y. R. Yang. Proportional fairness in multi-rate wireless LANs. In *Proc. of the 27th Conference on Computer Communications (INFOCOM 2008)*, pages 1004 – 1012, 2008.

[37] T. Li, Q. Ni, D. Malone, D. Leith, and Y. Xiao. Aggregation with fragment retransmission for very high-speed WLANs. *IEEE/ACM Transactions on Networking,*, 17(2):591–604, 2009.

[38] M. Luby. LT codes. In *Proc. of 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 271–280, 2002.

[39] M. Luby, M. Watson, T. Gasiba, and T. Stockhammer. Mobile data broadcasting over MBMS: tradeoffs in forward error correction. In *Proc. of the 5th International Conference on Mobile and Ubiquitous Multimedia*, 2006.

[40] M. Luby, M. Watson, T. Gasiba, T. Stockhammer, and W. Xu. Raptor codes for reliable download delivery in wireless broadcast systems. In *Proc. of IEEE Consumer Communications and Networking Conference (CCNC 2006)*, pages 192–197, 2006.

[41] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[42] D. J. C. MacKay. Fountain codes. In *IEE Proceedings Communications*, volume 152, pages 1062–1068, 2005.

[43] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*. North-Holland Publishing Co., 1977.

[44] A. Majeed and N. B. Abu-Ghazaleh. Packet aggregation in multi-rate wireless LANs. In *Proc. of the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 452–460, 2012.

[45] D. Malone, P. Clifford, and D. J. Leith. Mac layer channel quality measurement in 802.11. *IEEE Communications Letters*, 11(2):143–145, 2007.

[46] D. Malone, K. Duffy, and D. Leith. Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions. *IEEE/ACM Transaction on Networking*, 15(1):159–172, 2007.

[47] S. Mangold, S. Choi, and N. Esseling. An error model for radio transmissions of wireless LANs at 5GHz. In *Proc. of the 10th Aachen Symposium on Signal Theory*, pages 209–214, 2001.

[48] R. C. Manso. Performance analysis of M-QAM with Viterbi soft-decision decoding. Master's thesis, Naval Postgraduate School Monterey CA, 2003.

[49] A. M. Mood. The distribution theory of runs. *The Annals of Mathematical Statistics*, 11(4):367–392, 1940.

[50] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving MAC layer fairness in wireless packet networks. In *Proc. of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00)*, pages 87–98, 2000.

[51] Q. Ni, T. Li, T. Turletti, and Y. Xiao. Saturation throughput analysis of error-prone 802.11 wireless networks. *Wiley Journal of Wireless Communications and Mobile Computing*, 5(8):945–956, 2005.

[52] K. Premkumar, X. Chen, and D. J. Leith. Utility optimal coding for packet transmission over wireless networks – Part I: networks of binary symmetric channels. In *Proc. of the 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1592–1599, 2011.

[53] M. B. Pursley and D. J. Taipale. Error probabilities for spread-spectrum packet radio with convolutional codes and Viterbi decoding. *IEEE Transaction on Communnications*, 35(1):1–12, 1987.

[54] J. Rosenthal and R. Smarandache. Maximum distance separable convolutional codes. *Applicable Algebra in Engineering, Communication and Computing*, 10(1):15–32, 1999.

[55] M. Samokhina, K. Moklyuk, S. Choi, and J. Heo. Raptor code-based video multicast over IEEE 802.11 WLAN. In *Proc. of the 5th IEEE VTS Asia Pacific Wireless Communications Symposiu*, 2008.

[56] A. Shokrollahi. LDPC codes: an introduction. Technical report, Digital Fountain, Inc, 2003.

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

[58] M. K. Simon and M. S. Alouini. *Digital Communication over Fading Channels*. Wiley-IEEE Press, 2005.

[59] V. G. Subramanian and D. J. Leith. On a class of optimal rateless codes. In *Proc. of 46th Annual Allerton Conference on Communication, Control and Computing*, pages 418–425, 2008.

[60] I. Tinnirello, D. Giustiniano, L. Scalia, and G. Bianchi. On the side-effects of proprietary solutions for fading and interference mitigation in IEEE 802.11b/g outdoor links. *Elsevier Computer Networks*, 53(2):141–152.

[61] P. Verkaik, Y. Agarwal, R. Gupta, and A. C. Snoeren. Softspeak: Making VoIP play well in existing 802.11 deployment. In *Proc. of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI'09)*, pages 409–422, 2009.

[62] A. J. Viterbi. Convolutional codes and their performance in communication systems. *IEEE Transaction on Communnications Technology*, 19(5):751–772, 1971.

[63] A. Wald and J. Wolfowitz. On a test whether two samples are from the same population. *Annals of Mathematical Statistics*, 11(2):147–162, 1940.

[64] W. Wang, S. C. Liew, Q. Pang, and V. O. K. Li. A multiplexing-multicast scheme that improves system capacity of Voice-over-IP on wireless LAN by 100%. In *Proc. of the 9th IEEE International Symposium on Computers and Communications (ISCC 04)*, pages 472–477, 2004.

[65] S. Wicker and V. Bhargava. *Reed-Solomon Codes and Their Applications*. Wiley-IEEE Press, 1994.

[66] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz. Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. *IEEE/ACM Transactions on Industrial Electronics*, 49(6):1265 – 1282.

Appendix A

List of Acronyms

Acronym	Meaning
FEC	forward error correction
WLAN	wireless local area network
AP	access point
MAC	media access control
PHY	physical
DIR	diffuse infrared
FHSS	frequency hopping spread spectrum
DSSS	direct sequence spread spectrum
CCK	complementaty code keying
OFDM	orthogonal frequency division multiplexing
MIMO	multiple-input multiple-output
TDMA	time division multiple access
DCF	distributed coordination function
CSMA/CA	carrier sense multiple access with collision avoidance
PCF	point coordination function
DIFS	DCF inter-frame space
SIFS	short inter-frame space
PIFS	PCF inter-frame space
ACK	acknowledgement
NACK	negative acknowledgement

Acronym	Meaning
QoS	quality of service
RTS	request to send
CTS	clear to send
PEC	packet erasure channel
BSC	binary symmetric channel
CRC	cyclic redundancy check
PER	packet error rate
BER	bit error rate
SNR	signal to noise ratio
AWGN	additive white gaussian noise
PAM	pulse amplitude modulation
BPSK	binary phase shift keying
QPSK	quadrature phase shift keying
QAM	quadrature amplitude modulation
HDD	hard decision decoding
SDD	soft decision decoding
PLCP	physical layer convergence procedure
FER	frame error rate
UDP	user datagram protocol
RTP	real-time transport protocol
RSSI	received signal strength indicator
VoIP	voice over internet protocol
CBR	constant bit rate

Appendix B

Mean of the maximum of M independent negative binomial random variables of order n

Suppose we have M independent negative binomial random variables of order n, $\{X_i, i \in \{1, \cdots, M\}\}$. The distribution is given by

$$P(X_i = x) = \binom{x-1}{n-1} p_i^n (1 - p_i)^{x-n} \tag{B.1}$$

for $x = n, n+1, n+2, \cdots$. X_i is the number of Bernoulli trials with parameter p_i needed to get n successes.

Define T as the maximum of these M random variables, $T = \max_{i \in \{1, \cdots, M\}} X_i$. The mean of T is

$$
\begin{aligned}
E(T) &= \sum_{t=n}^{\infty} t P(T = t) \\
&= \sum_{\tau=n}^{\infty} \sum_{t=1}^{\tau} P(T = \tau) \\
&= \sum_{t=1}^{n} 1 + \sum_{t=n+1}^{\infty} P(T \geqslant t) \\
&= n + \sum_{t=n+1}^{\infty} (1 - P(T < t)) \\
&= n + \sum_{t=n}^{\infty} (1 - P(T \leqslant t)) \\
&= n + \sum_{t=n}^{\infty} (1 - F_T(t))
\end{aligned}
\tag{B.2}
$$

where $F_T(t)$ is the cumulative distribution function of T, given by

$$
\begin{aligned}
F_T(t) &= P(T \leqslant t) \\
&= P(\max_{i \in \{1, \cdots, M\}} X_i \leqslant t) \\
&= P(X_1 \leqslant t, X_2 \leqslant t, \cdots, X_M \leqslant t) \\
&= \prod_{i=1}^{M} P(X_i \leqslant t) \\
&= \prod_{i=1}^{M} \sum_{x=n}^{t} P(X_i = x)
\end{aligned}
\tag{B.3}
$$

Therefore, the mean of the maximum of M independent negative binomial random variables of order n is given by

$$
E(T) = n + \sum_{t=n}^{\infty} \left(1 - \prod_{i=1}^{M} \sum_{x=n}^{t} \binom{x-1}{n-1} p_i^n (1 - p_i)^{x-n} \right)
\tag{B.4}
$$

Appendix C

Network Simulator 2

The Network Simulator 2 (NS-2) is a discrete event simulator targeted at networking research. It is built in C++ and provides a simulation interface through OTcl, an object-oriented dialect of Tcl. The user describes a network topology by writing OTcl scripts, and then the main NS program simulates that topology with specified parameters [2].

NS-2 provides substantial support for the simulation of UDP, TCP, routing, and multicast protocols over wired and wireless networks. Therefore it is widely used to simulate the behaviors of 802.11 WLANs.