Chapter 3. Modulation and Coding

IN THIS CHAPTER

Modulation and channel coding are fundamental components of a digital communication system. Modulation is the process of mapping the digital information to analog form so it can be transmitted over the channel. Consequently every digital communication system has a *modulator* that performs this task. Closely related to modulation is the inverse process, called *demodulation,* done by the receiver to recover the transmitted digital information. The design of optimal demodulators is called *detection* theory. An OFDM system performs modulation and demodulation for each subcarrier separately, although usually in a serial fashion, to reduce complexity.

Channel coding, although not strictly speaking mandatory for digital communication, is essential for good performance. After Claude Shannon published his groundbreaking paper [16] that started *information theory* as a scientific discipline and established the fundamental law equation 3.1 of the capacity *C* of an AWGN communication channel with bandwidth *W* and signal to noise ratio *SNR,* communication scientists and engineers have been searching for methods on how to reach the capacity of the channel.

Equation 3.1

$$C = W \log_2(1 + SNR)$$

Shannon's proof of the channel capacity formula was nonconstructive, and thus did not provide a method to actually reach the capacity *C.* As a consequence, *coding theory* got started as a discipline that research methods to design a communication system that has performance close to the fundamental limit *C.* The methods developed under coding theory have different names; channel coding, forward error correction (FEC), and error control coding are the most usual. Channel coding and basic types of FEC codes were introduced in "Channel Coding" in Chapter 1. Note that channel coding and source coding, introduced in the "Source Coding" section in Chapter 1, although related, are quite different topics.

Our goal in this chapter is to discuss modulation, detection, channel coding and interleaving from an OFDM system of point of view. Channel coding is such an important component of OFDM systems that the term Coded Orthogonal Frequency Division Multiplexing (COFDM) is sometimes used. The reasons for this emphasis on channel coding for OFDM systems will become apparent in the channel coding section of this chapter. Modulation and demodulation or detection are discussed in depth in several books [11, 17, 18]. Channel coding is the topic of a wide variety of books [10, 20] and also books that concentrate on specific types of FEC codes [7, 9].

Modulation

As was discussed in "Modulation" in Chapter 1, modulation can be done by changing the amplitude, phase or frequency of transmitted Radio Frequency (RF) signal. For an OFDM system, the first two methods can be used. Frequency modulation cannot be used because subcarriers are orthogonal in frequency and carry independent information. Modulating subcarrier frequencies would destroy the orthogonality property of the subcarriers; this makes frequency modulation unusable for OFDM systems.

The main design issue of the modulator is the *constellation* used. Constellation is the set of points that can be transmitted on a single symbol. The used constellation affects several important properties of a communication system; for example, bit error rate (BER), peak to average power ratio (PAPR), and RF spectrum shape. The single most important parameter of a constellation is *minimum distance.* Minimum distance ($d_{\min}$) is the smallest distance between any two points in the constellation. Therefore ($d_{\min}$) determines the least amount of noise that is needed to generate a decision error. Actual BER or P_b of a constellation can in many cases be calculated using the Q-function defined in Equation 3.2. The value of the Q-function is equal to the area under the tail of the Probability Density Function (PDF) of a zero mean and unit variance normal random variable.

Equation 3.2

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt \qquad x \geq 0$$

P_b is a function of energy per bit to noise ratio $\frac{E_b}{N_o}$ or signal energy to noise ratio $\frac{E_b}{N_o}$; these ratios are related by a simple scaling factor that depends on the number of bits k transmitted per symbol $E_s = kE_b$. The usefulness of $\frac{E_b}{N_o}$ as a measure of the signal quality will became apparent when different coding schemes are discussed in "Channel Codes" later in this chapter. Then a general form of P_b equations for several important constellations in terms of the Q-function is shown in Equation 3.3.

Equation 3.3

$$P_b \sim Q\left(\sqrt{\frac{E_b}{N_o}}\right)$$

The value of the Q-function gets smaller for larger arguments, hence Equation 3.3 shows that large $\frac{E_b}{N_o}$ implies better performance. Different constellations have different $d_{\min}$ values for the same $\frac{E_b}{N_o}$. The constellation with the largest $d_{\min}$ for a given $\frac{E_b}{N_o}$ has the best performance. In P_b equations like 3.3, this effect results in scaling of the value and the argument of the Q-function.

Minimum distance depends on several factors: number of points M, average power P_{ave} and shape of the constellation. The most important is the number of points in the constellation, which is directly dependent on the number of bits k transmitted in one symbol $M = 2^k$. Average power P_{ave} scales the constellation smaller or larger, depending on the transmitted power. Consequently, to make comparisons between different constellations fair, P_{ave} is usually normalized to one when d_{min} of a constellation is calculated.

Average power of a constellation is evaluated simply by averaging the power of all the M points C_k of the constellation as shown in Equation 3.4.

Equation 3.4

$$P_{ave} = \frac{1}{M}\sum_{k=1}^{M}|c_k|^2$$

Another important factor that influences in $d_{\min}$ is the shape of the constellation. A constellation is basically a set of points located in one dimensional or larger space spanned by the transmitted symbols. Traditional constellations are either one or two dimensional. In principle, these points can be located in any shape whatsoever. Naturally, for communications purposes care must taken to ensure good performance and practical implementation. There are two main goals when constellation shapes are designed. The first is constant amplitude of the transmitted signal. In this case, all the points in the constellation are required to have equal amplitude. The second goal is to improve $d_{\min}$. This is an optimization problem that tries to place all the points such that $d_{\min}$ is maximized for a finite P_{ave}. For the usual case of a two dimensional constellation with large number of points, this results in a circular constellation. Forney and Wei [4] give a through analysis of the behavior of d_{min} for different constellation types.

Coherent Modulations

Coherent modulation can be used by a communication system that maintains a phase lock between the transmitter and receiver RF carrier waves. Coherent modulation improves performance, but requires more complex receiver structure compared to *non-coherent* systems that are discussed later in this chapter. The performance gain of coherent modulation is significant when the system uses large constellations. High speed communication systems, like IEEE 802.11a, are usually coherent. The following sections describe the most common coherent modulations and their performance.

Amplitude Shift Keying

Amplitude Shift Keying (ASK) modulation transmits information by changing the amplitude of the carrier. Equation 3.5 shows the carrier waveform of ASK modulation. The A_k term does the actual modulation by multiplying the carrier wave by the selected amplitude level. The transmitted bits determine which of the possible symbols $\{A_1,\ldots,A_m\}$ is selected.

Equation 3.5

$$s(t) = A_k \cos(\omega_c t)$$

Figure 3.1 shows several different ASK modulations; (a) is a 2-ASK or BPSK, (b) is a 4-ASK, and (c) is a 8-ASK. In the figure, all the modulations have the same minimum distance $d_{\min}^2 = 4$, however the average power of the constellations is not equal. To perform fair comparison of the ASK constellations, Table 3.1 shows the normalized $d_{\min}^2$ and the required increase in SNR to maintain the same BER when changing to a one-step-larger constellation. The SNR increase converges to about 6 dB for each additional bit in the constellation; for this reason, large ASK modulations are rarely used in practice. The 2-ASK modulation does not have an *SNR* increase value, because it is the smallest possible constellation. The IEEE 802.11a system uses only the smallest 2-ASK modulation; for larger constellations, Phase Shift Keying and Quadrature Amplitude Modulations are used in practice. These two methods are the topics of the next sections.

Figure 3.1. ASK modulations, (a) 2-ASK, (b) 4-ASK, (c) 8-ASK.

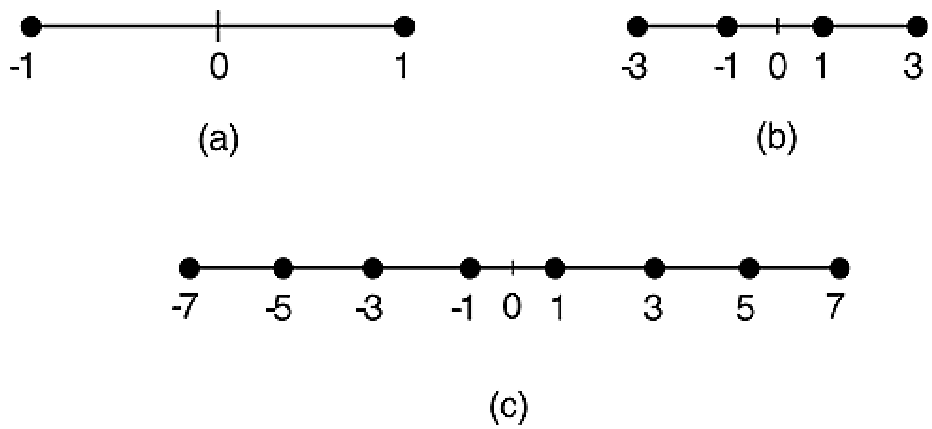
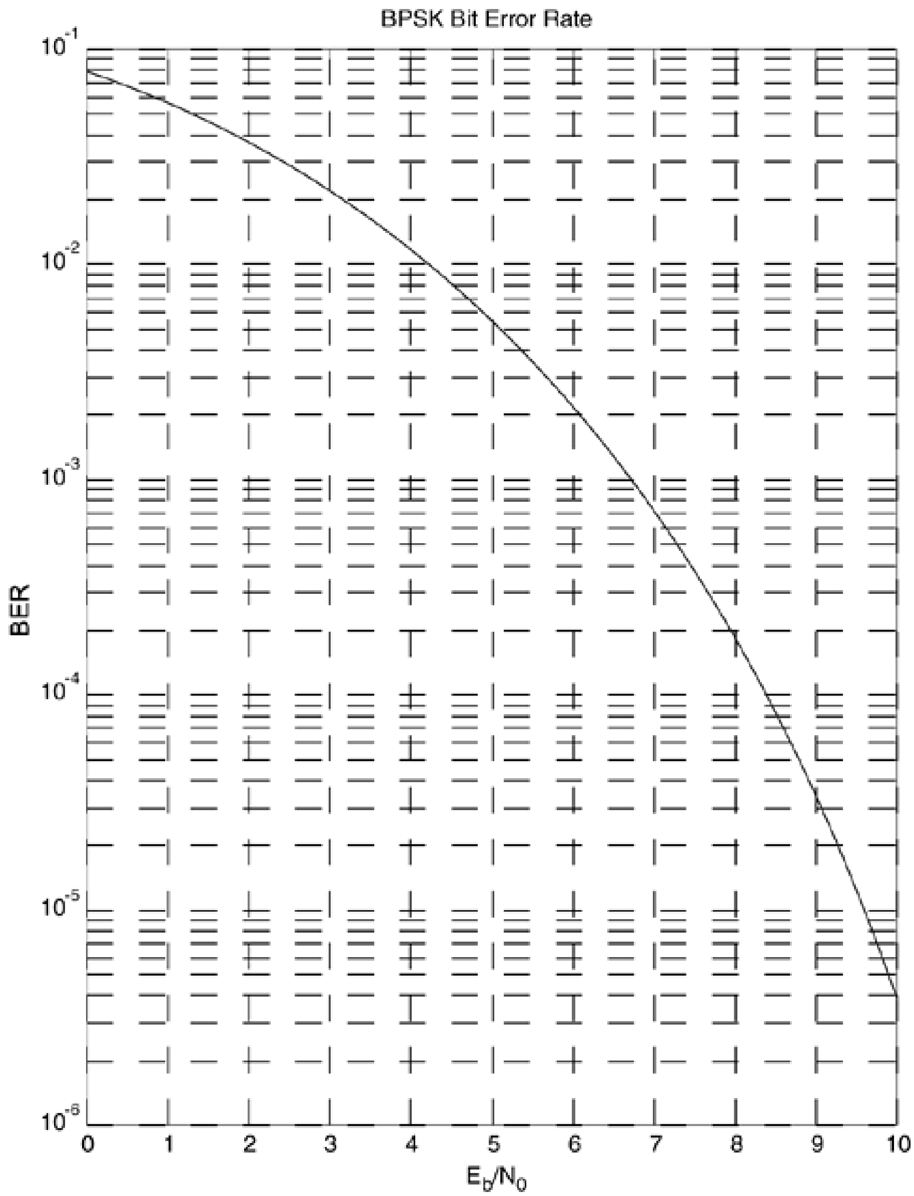


Table 3.1. Distance Properties of ASK Modulations

Modulation	P_{ave}	$d^2_{\min}$ ***Normalized***	***SNR Increase***
2-ASK	1	4	–
4-ASK	5	$\frac{4}{5}$	6.99 dB
8-ASK	21	$\frac{4}{21}$	6.23 dB

Bit error rate P_b of binary ASK is the most basic error probability (see Equation 3.6). The P_b of 2-ASK is plotted in Figure 3.2. It is useful to memorize a reference point from this curve, like $P_b = 2 \cdot 10^{-4}$ at 8.0 dB $\frac{E_b}{N_o}$ or $P_b = 10^{-5}$ at 9.6 dB $\frac{E_b}{N_o}$. These points can serve as a quick check to validate simulation results. As the number of points in the constellation increases, exact analytic P_b equations become quite complicated, therefore either formulas for symbol error rate P_s or approximations to P_b are more commonly used.

Figure 3.2. BPSK bit error rate.



Equation 3.6

$$P_b = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

Symbol error rate P_s of M-ary ASK modulations is equal to Equation 3.7. Bit error rate P_b can be approximated by dividing P_s by the number of bits per symbol as in Equation 3.8. The formula for P_b is an approximation, because generally the number of bit errors that occur for each symbol error can be more than one. Equation 3.8 assumes that each symbol error causes only one bit error. However, with the help of Gray coding, which is discussed in the section "Labeling Constellation Points," for most symbol errors the number of bit errors is equal to one. The quality of this approximation improves at high signal to noise ratios.

Equation 3.7

$$P_s = 2\frac{M-1}{M}Q\left(\sqrt{\frac{A^2}{2N_0}}\right)$$

In Equation 3.7, *A* is the amplitude difference between symbol levels. For example, in the constellation in Figure 3.1, *A=2*.

Equation 3.8

$$P_b \approx \frac{P_s}{\log_2 M} = \frac{P_s}{k}$$

Phase Shift Keying

Phase Shift Keying (PSK) modulations transmit information by changing the phase of the carrier, the amplitude is kept constant; hence PSK modulations are also called *constant amplitude modulations.* Equation 3.9 shows the carrier waveform of PSK signal. Modulation is done by the ϕ_k term.

Equation 3.9

$$s(t) = \cos(\omega_c t + \phi_k)$$

The main benefit of constant amplitude modulation is the peak-to-average power ratio (PAPR) that is practically equal to one. However, this is only true for a single carrier systems. An OFDM signal that is a sum of several modulated subcarriers does not have a constant amplitude, even if the individual subcarriers do. The main benefit of constant amplitude modulation is simplified RF chain design for both the transmitter and receiver. An OFDM signal can never have a constant amplitude, hence having a constant amplitude constellation does *not* benefit OFDM systems. Several techniques to mitigate the large amplitude variations of OFDM signals are discussed in Chapter 5.

Figure 3.3 shows the three smallest PSK constellations, (a) is 2-PSK or binary PSK (BPSK), (b) is 4-PSK or quadrature PSK (QPSK), and (c) is 8-PSK. Note that BPSK is identical to 2-ASK. The figure shows how PSK modulations use both the *Inphase (I)* and *Quadrature (Q)* carrier waves, hence the modulation is two dimensional. This increase in dimensionality of the constellation improves the behavior of $d_{\min}^2$ as the number of points in the constellation increases.

Figure 3.3. The three smallest PSK modulations, (a) BPSK, (b) QPSK, (c) 8-PSK.

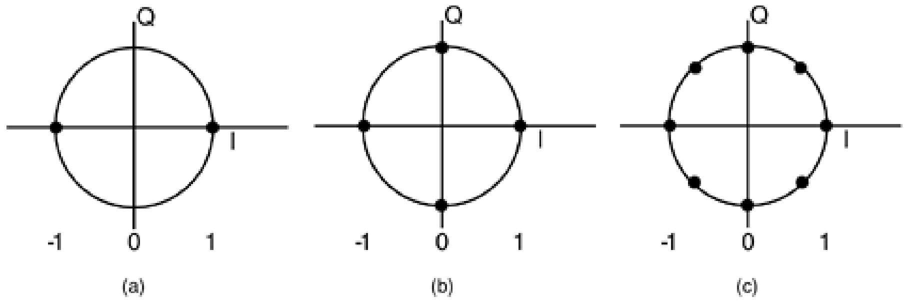


Table 3.2 shows the required increase in *SNR* for 2- to 16-point PSK constellations. Note particularly that the increase from BPSK to QPSK is only 3 dB, compared to 2-ASK to 4-ASK increase of 6.99 dB. However, as the number of points is increased to 16, *SNR* increase for one additional bit per symbol converges towards the 6 dB figure as for ASK modulation. This behavior can be attributed to the fact that although PSK constellation is two dimensional, it still has only one degree of freedom: the phase of the carrier. Thus PSK does not fully use the two-dimensional space to locate the points of the constellation. Therefore larger than 8-point PSK modulations are not commonly used.

Table 3.2. Distance Properties of PSK Modulations

Modulation	P_{ave}	$d^2_{\min}$ *Normalized*	*SNR Increase*
BPSK	1	4.00	–
QPSK	1	2.00	3.00 dB
8-PSK	1	0.5858	5.33 dB
16-PSK	1	0.1522	5.85 dB

Symbol error rate of PSK modulations can be derived exactly for BPSK and QPSK; the former case was already shown in Equation 3.6. For QPSK, the P_s expression is somewhat more complicated than BPSK as shown in Equation 3.10.

Equation 3.10

$$P_s = 2Q\left(\sqrt{2\frac{E_b}{N_0}}\right)\left[1-\frac{1}{2}Q\left(\sqrt{2\frac{E_b}{N_0}}\right)\right]$$

Higher order PSK modulation P_s can be approximated by Equation 3.11. Bit error rate can again be approximated by dividing P_s by the number of bits $k = \log_2 M$ in the constellation.

Equation 3.11

$$P_s = 2Q\left(\sqrt{\frac{E_s}{N_0}}\right)\sin\left(\frac{\pi}{M}\right)$$

Quadrature Amplitude Modulation

Quadrature Amplitude Modulation (QAM) changes both the amplitude and phase of the carrier, thus it is a combination of both ASK and PSK. Equation 3.12 shows the QAM signal in so-called *IQ*-form that presents the modulation of both the *I*- and *Q*-carriers. QAM can also be described by Equation 3.13 that shows how amplitude and phase modulations are combined in QAM.

Equation 3.12

$$s(t) = I_k \cos(\omega_c t) - Q_k \sin(\omega_c t)$$

Equation 3.13

$$= A_k \cos(\omega_c t + \phi_k)$$

The amplitude and phase terms of Equation 3.13 are calculated from Equations 3.14 and 3.15.

Equation 3.14

$$A_k = \sqrt{I_k^2 + Q_k^2}$$

Equation 3.15

$$\phi_k = \tan^{-1}\left(\frac{Q_k}{I_k}\right)$$

Figure 3.4 shows three QAM constellations: (a) 4-QAM or QPSK, (b) 8-QAM, and (c) 16-QAM. Especially QPSK and 16-QAM are very common modulations. On the other hand, 8-QAM is not used as often, probably due to its somewhat inconvenient shape; instead 8-PSK is commonly used when 8-point constellations are required. Figure 3.4 (b) is not the only possible shape for a 8-QAM constellation.

Figure 3.4. QAM constellations, (a) QPSK, (b) 8-QAM, (c) 16-QAM.

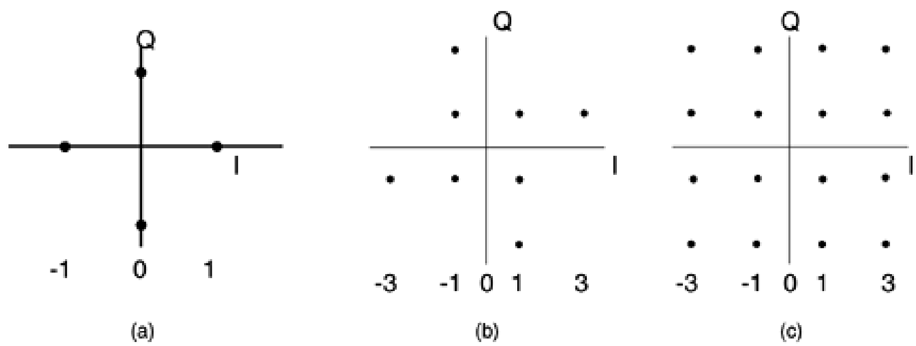


Table 3.3 shows how the $d^2_{\min}$ of QAM behaves as a function of the constellation size. The step from QPSK to 8-QAM is somewhat anomalous, and it is actually possible to improve the minimum distance of our example 8-QAM modulation. However, we have used the present example because of its regular form. The step from 16-QAM to 32-QAM shows a 3 dB increase in required SNR to maintain constant BER. This 3 dB SNR increase for each additional bit per symbol is a general rule for QAM constellations.

Symbol error rate of QAM modulation is approximated by considering QAM as two independent ASK modulation on both *I*- and *Q*-carriers. This is shown in Equation 3.16.

Equation 3.16

$$P_s \approx 4\left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{3E_s}{(M-1)N_0}}\right)$$

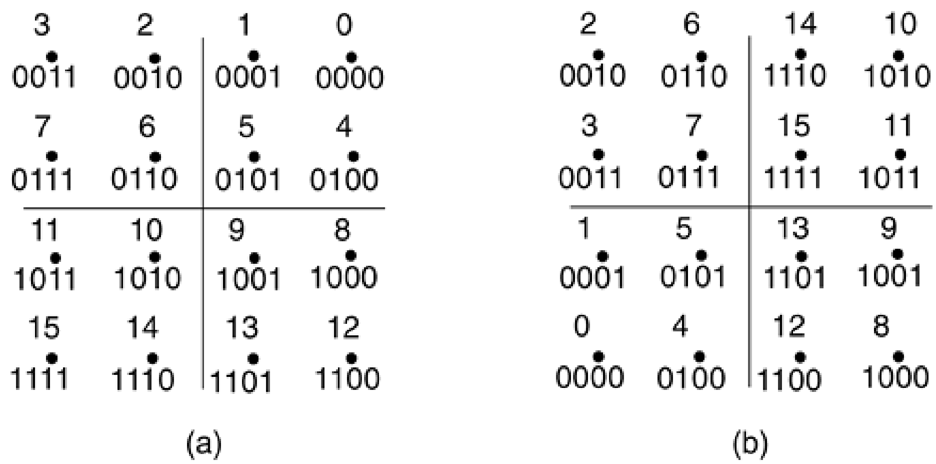
Table 3.3. Distance Properties of QAM Modulations

Modulation	***P_{ave}***	***$d^2_{\min}$ Normalized***	***SNR Increase***
QPSK	1	2.00	–
8-QAM	6	0.67	4.77 dB
16-QAM	10	0.40	2.22 dB
32-QAM	20	0.20	3.01 dB

Labeling Constellation Points

Labeling constellation points means assigning a bit pattern to all the points. The selected labeling scheme has an impact on the performance and thus must be optimized. Figure 3.5 shows two common ways to assign bit patterns to 16-QAM constellation points: (a) is called natural order, and (b) is called Gray coding. Figure 3.5 shows the decimal value of the label above the point and binary value below the point. Natural ordering has appeal in its straightforward assignment of labels using decimal numbers from 0 to 15 to the points. The disadvantage of natural ordering is in the actual bit patterns representing constellation points.

Figure 3.5. (a) QAM natural order and (b) Gray coded labeling schemes.



For example, consider an error where point number 1 was transmitted, but it was received as 2. This mistake between neighboring points is the most common type error the receiver does. Now look at the bit patterns representing 1 (0001) and 2 (0010); two bits have changed resulting in two bit errors for one symbol error. Gray coding eliminates all these two-bit errors for symbol errors between neighboring points, thus reducing bit error rate for the same symbol error rate. In the Gray coded constellations the points of the previous example are labeled 14 (1110) and 6 (0110); note that only left most bit is different. All the constellations in IEEE 802.11a and HiperLAN/2 standards are Gray coded.

Exercise 1 *Plot the different bit error probability functions corresponding to the modulations used in IEEE 802.11a and compare the curves with simulation results.*

Exercise 2 *Compare the performance difference predicted by minimum distace values of different constellations to the analytical bit error probability curves.*

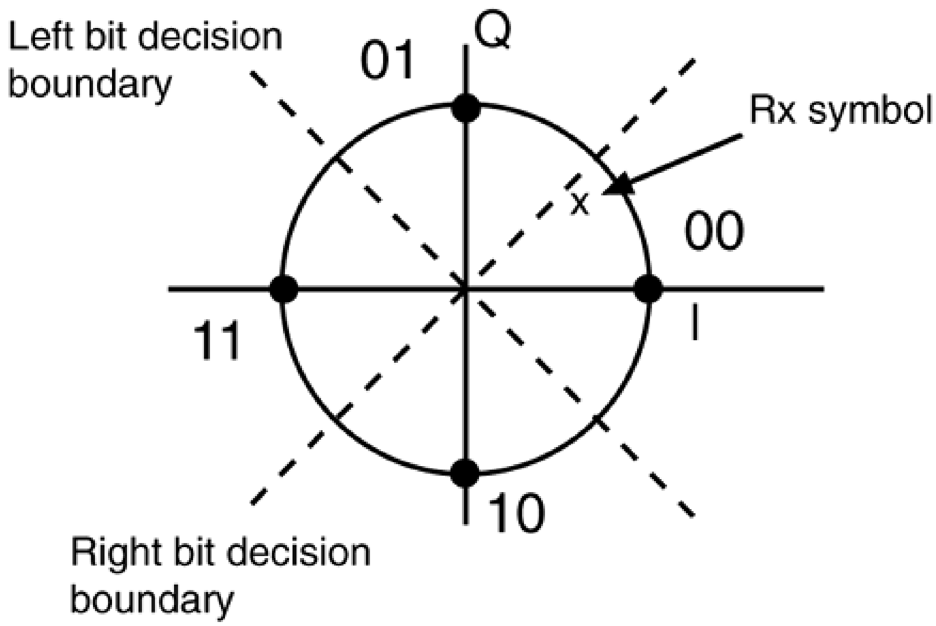
Detection of Coherent Modulations

After the receiver has performed all the required synchronization operations, discussed in Chapter 2, the next job is to decide what was actually received. This is done by a detector or demodulator that decides for each received symbol the most likely transmitted bits. The decisions are divided into *hard* and *soft* decisions, depending on how much information about each transmitted bit is produced. Both types of detectors are discussed next.

Hard Decision Detection

A hard decision *demodulator* or *slicer* makes a definite determination of whether a zero or one bit was transmitted, thus the output of the demodulator are zeros and ones. A hard decision demodulator is also defined based on the number of possible inputs to the modulator in the transmitter. If this number is equal to the number of possible outputs from the demodulator in the receiver, the demodulator uses hard decision. If the demodulator can output more information than the input to the modulator, the system uses soft decision, which is discussed later. Figure 3.6 shows the *decision boundaries* for QPSK constellation. The decision boundaries determine how received symbols are mapped to bits. Essentially, the maximum-likelihood decision is the constellation point that is closest to the received symbol, and the hard decisions are the bits assigned to that constellation point. For example, in Figure 3.6, the *x* marks the location of the received symbol; it is closest to the constellation point on the positive *I*-axis, hence the hard decision bits are 00.

Figure 3.6. Hard decision boundaries for QPSK constellation.



Soft Decision Detection

Soft decision demodulator outputs "*soft*" bits, that in addition to indicating a zero or one bit retain information about the reliability of the decision. This additional information can greatly improve the performance of channel coding schemes; the effect will be discussed in "Channel Codes." To perform soft decisions, the demodulator has to consider the received bits individually. Consider again the received symbol in Figure 3.6. It is located very close to the decision boundary between symbols corresponding to bits 00 and 01. For the left bit to change to 1, the symbol would have to move over the left bit decision boundary. The received symbol is quite far from this boundary, hence the left bit is quite reliable. On the other hand, the right bit changes from 0 to 1; if the received symbol would be above the right bit decision boundary, instead of just below it. This flip does not require much additional noise, hence the right bit is quite unreliable. To reflect this difference of reliability of the soft decision, the soft bits have different values: a large absolute value for the first bit and a small value for the second bit. The sign of the soft decision indicates a 0 or 1 bit. The absolute value of each soft decision is the distance to the decision boundary. We will show the performance effect of using either hard or soft decision with the convolutional error correcting code used in IEEE 802.11a in "Channel Codes."

Non-Coherent Modulations

Non-coherent modulations can be used by a communication system that does not maintain a phase lock between transmitter and receiver, or have knowledge of the amplitude change of the transmitted symbol caused by the channel. This means that the received symbols are rotated and scaled arbitrarily compared to the transmitted symbol. Therefore the ASK, PSK, or QAM modulations cannot be used because they require the received symbol phase and amplitude to be very close to that transmitted phase and amplitude. The solution is to use differential PSK (DPSK) or differential APSK (DAPSK) modulation. Differential modulations encode the transmitted information to a phase, or phase and amplitude change from one transmitted symbol to the next. This encoding introduces memory to the signal, because transmitted symbols depend on previous symbols. As a consequence, the demodulator has to consider two consecutive symbols when making decisions. The next two sections describe these two modulation methods.

The main benefit of differential encoding is significantly simplified receiver structure. Several of the synchronization algorithms presented in Chapter 2 are not needed in a non-coherent receiver. Specifically, phase tracking and channel estimation are not needed, because absolute knowledge of carrier phase and the channel effect is not needed. Carrier frequency estimation could also be removed, if the system can tolerate the performance due to intercarrier interference caused by lost orthogonality between the subcarriers. Regardless of these simplifications in receiver design, non-coherent modulations have not achieved popularity among high speed communications systems. All the IEEE 802.11a and HiperLAN/2 modulations are coherent. The main reason is unavoidable performance loss associated with differential approach. In contrast, low data rate systems do use differential techniques, mainly DPSK modulations. For example, the European Digital Audio Broadcasting (DAB) system uses differential modulation.

Differential Phase Shift Keying

Differential phase shift keying changes the carrier phase from its current value according to the data bits. For binary DPSK, the encoding operation can be expressed at bit level, as in Equation 3.17, which shows how the transmitted bit b_n is calculated from the data bits d_n using binary XOR operation.

Equation 3.17

$$b_n = d_n \oplus d_{n-1}$$

Alternatively, encoding can be expressed directly as carrier phase change as in Equation 3.18

Equation 3.18

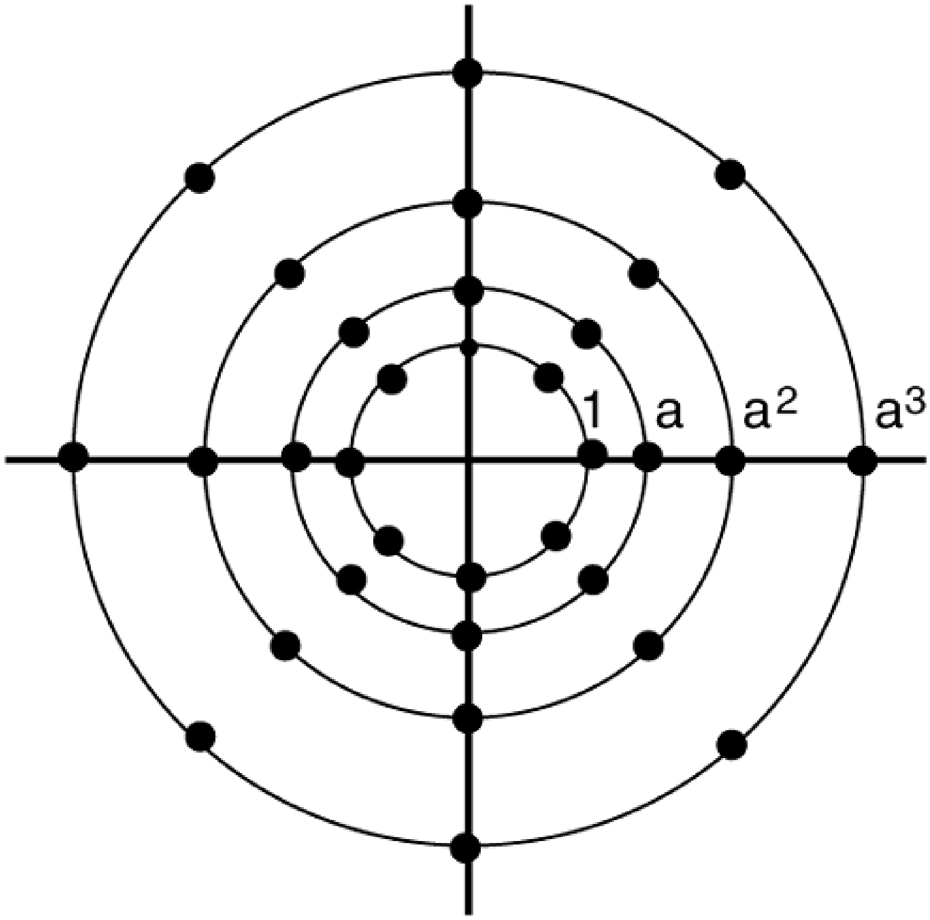
$$s(t) = \cos(\omega_c t + \Delta\phi_n + \phi)$$

where ϕ is unknown phase difference between transmitter and receiver and $\Delta\phi_n = \phi_n - \phi_{n-1}$ is the phase difference of two PSK modulated signals. DPSK is used in communications systems that require simple receiver structures. The constellation for M-DPSK modulation is identical with the corresponding M-PSK constellation; only the mapping of bits to constellation points is changed.

Differential Amplitude Phase Modulation

Differential Amplitude Phase Modulation (DAPSK) combines differential phase and differential amplitude modulation. Figure 3.7 shows a 32-point DAPSK constellation that uses three bits or eight levels for phase modulation and two bits or four levels for amplitude modulation. The differential phase modulation is analogous to the regular DPSK. Differential amplitude modulation, on the other hand, has to change the constellation shape compared to coherent amplitude modulation. The reason is the unknown scaling of the amplitude of the transmitted symbol caused by the channel. The receiver cannot correctly detect the symbols unless this unknown scaling is removed. Equation 3.19 shows a received OFDM subcarrier symbol R_k when the transmitted symbol was X_k and the subcarrier frequency response is H_k.

Figure 3.7. 32-point differential amplitude and phase modulation.



Equation 3.19

$$R_k = H_k X_k$$

To recover the amplitude modulation the receiver is only concerned with the amplitude of the received symbol, as shown in Equation 3.20.

Equation 3.20

$$|R_k| = |H_k X_k| = |H_k||X_k|$$

A general assumption when differential modulation is used is that the channel and carrier phase are constant during two consecutive symbols. Therefore we can cancel the effect of the channel by dividing two consecutive symbols, as shown in Equations 3.21 – 3.23.

Equation 3.21

$$Y_k = \frac{|H_k||X_k|}{|H_{k-1}||X_{k-1}|}$$

Equation 3.22

$$= \frac{|H_k||X_k|}{|H_k||X_{k-1}|}$$

Equation 3.23

$$= \frac{|X_k|}{|X_{k-1}|}$$

The multiplicative scaling of the channel prevents using additive differential amplitude modulation, whereas for DPSK the channel effect on received symbol phase is additive, hence it can be canceled by subtraction instead of division. The next question is how the amplitude levels of the constellation are selected. This is illustrated in Figure 3.7, which shows amplitude levels 1, a, a^2, a^3. The information is encoded as a jump from one amplitude level to another; for example, bits 01 could mean a jump of one amplitude level. Differential amplitude encoding done with multiplication such as

Equation 3.24

$$Y_k = D_k Y_{k-1}$$

where $D_k \in \left\{\frac{1}{a^3}, \frac{1}{a^2}, \frac{1}{a}, 1, a, a^2, a^3\right\}$. The fractional values are required, because amplitude multiplication is not naturally periodic like phase addition in the case of DPSK modulation. The actual value of D_k depends on the input bits and the value of Y_{k-1}. The value of D_k is selected such that the amplitude jumps wrap around after a^3. For example, if $Y_{k-1} = a^2$ and a two amplitude level jump is needed, then $D_k = \frac{1}{a^2}$, so $Y_k = a^2 \frac{1}{a^2} = 1$. The two amplitude level jump is then $a^2 \longrightarrow a^3 \longrightarrow 1$.

The multiplication encoding method forces the points in the middle of the constellation quite close to each other. This is an unavoidable disadvantage of differential amplitude modulation. The effect is reduced minimum distance for the points in the middle, hence they experience a performance loss compared to the points on the outer amplitude levels.

Detection of Differential Modulations

Detection of differential modulation is done in two steps. First the differential encoding is removed from the signal and then a normal demodulation is performed as explained for regular PSK and QAM constellations. Equations 3.25 and 3.26 show how the differential encoding is removed in DPSK detection by calculating the phase difference of two consecutive symbols. Equation 3.26 also show the reason for the performance loss of differential encoding compared to coherent modulations. Differential detection has to use two symbols, therefore the amount of noise per detected symbol is approximately doubled.

Equation 3.25

$$\hat{\theta}_k = \angle(R_k + N_k) - \angle(R_{k-1} + N_{k-1})$$

Equation 3.26

$$= \theta_k - \theta_{k-1} + \tilde{N}_k + \tilde{N}_{k-1}$$

The performance loss of DPSK compared to coherent modulation varies with the size of the modulation [11], for DBPSK it is between 1-2dB, for DQPSK about 2.3dB and for larger constellations 3dB.

Linear and Nonlinear Modulation

Different modulation methods are generally divided into linear and nonlinear modulations. The differentiating property is whether the transmitted signal can be expressed as a sum of the individual symbols a_k with pulse shape $P(t)$ as in Equation 3.27.

Equation 3.27

$$s(t) = \sum_{k=-\infty}^{\infty} a_k p(t - kT)$$

Nonlinear modulation usually contains memory in the modulator, like differential modulation, so that the transmitted symbols are dependent and cannot be simply delayed and summed together to form the transmitted waveform. Differential modulation is one application of nonlinear modulation; another common one is *continuous phase modulation* (CPM). The main goal of CPM is to control the spectrum of the transmitted signal by forcing the phase of the carrier wave to change continuously between symbols. This narrows the spectrum of the signal, because the change between symbols is less abrupt than without carrier phase control. CPM is not very useful for OFDM, because the spectrum shape is determined by the multicarrier nature and has a brick-like shape, even with linear modulation. Another difficulty in applying CPM to OFDM is that the signal would have to have a continuous phase for all the subcarriers. This is difficult or impossible to achieve, because of the cyclic prefix added to OFDM symbols. Therefore CPM is not used with OFDM systems.

Interleaving

Interleaving aims to distribute transmitted bits in time or frequency or both to achieve desirable bit error distribution after demodulation. What constitutes a desirable error distribution depends on the used FEC code. What kind of interleaving pattern is needed depends on the channel characteristics. If the system operates in a purely AWGN environment, no interleaving is needed, because the error distribution cannot be changed by relocating the bits. Communication channels are divided into *fast* and *slow* fading channels. A channel is fast fading if the impulse response changes approximately at the symbol rate of the communication system, whereas a slow fading channel stays unchanged for several symbols.

WLAN systems generally assume a very slowly fading channel, also called *quasi-stationary,* that does not change during one packet. Another characterization of communications channels is as *flat* or *frequency selective* fading channels. A channel is flat fading if the frequency response is constant over the whole bandwidth of the transmitted signal. A frequency selective channel changes significantly within the band of the signal. WLAN systems are wide bandwidth systems, and therefore usually experience frequency selective fading channel. OFDM technology is well suited for communication over slow frequency selective fading channels.

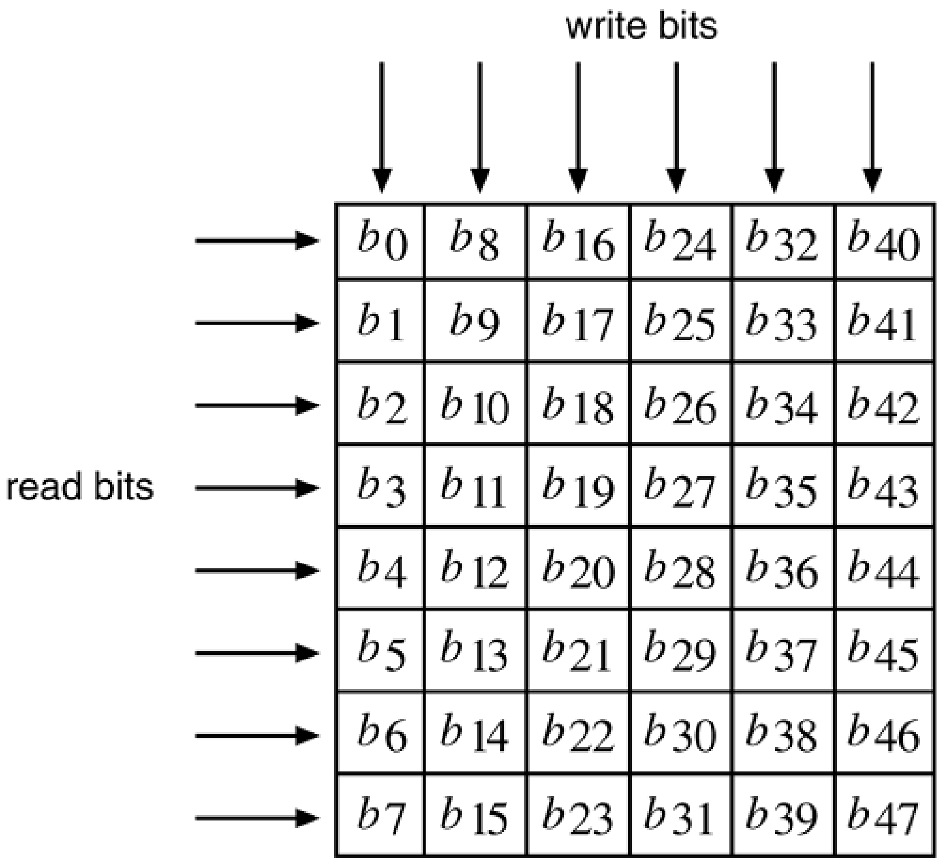
Diversity is the single most important technique to improve the performance of a communication system in a fading channel. Diversity generally refers to methods that take advantage of several different realizations of the channel to make a decision on each received information bit. The more independent channel realizations can be used, the better the performance improvement will be. This is a consequence of the channel fading statistics. The more individual samples of the channel are combined, the less likely it is that all of them are in a faded state. There are several sources of diversity that can be used: time, frequency, and space are the most common ones. OFDM wireless LANs mainly exploit frequency diversity due to their wideband nature. Time diversity cannot be used, because of the slowly fading characteristics of the channel. When the channel is in a bad condition, it will remain so for the duration of the packet, and there is nothing the receiver can do about it. Space diversity is achieved by using multiple transmit and/or receiver antennas. This technique is discussed in detail in Chapter 4.

Interleaving necessarily introduces delay into the system because bits are not received in the same order as the information source transmits them. The overall communication system usually dictates some maximum delay the system can tolerate, hence restricting the amount of interleaving that can be used. For example, cellular telephone systems usually use time diversity, because the channels are fast fading. However the maximum phone to phone delay is usually constrained to 20ms or less, to prevent noticeable degradation in call quality. This means the maximum interleaving delay must be much less than 20ms to allow for other delay sources in the system.

Block Interleaver

Block interleaving operates on one block of bits at a time. The number of bits in the block is called *interleaving depth,* which defines the delay introduced by interleaving. A block interleaver can be described as a matrix to which data is written in columns and read in rows, or vice versa. For example, Figure 3.8 shows a 8x6 block interleaver, hence interleaving depth is 48. The input bits are written in columns as [b_0, b_1, b_2, b_3,…] and the interleaved bits are read by rows as [b_0, b_8, b_{16}, b_{24},…] .Block interleaver is simple to implement using random access memory (RAM) on current digital circuits.

Figure 3.8. Bit write and read structure for a 8x6 block interleaver.

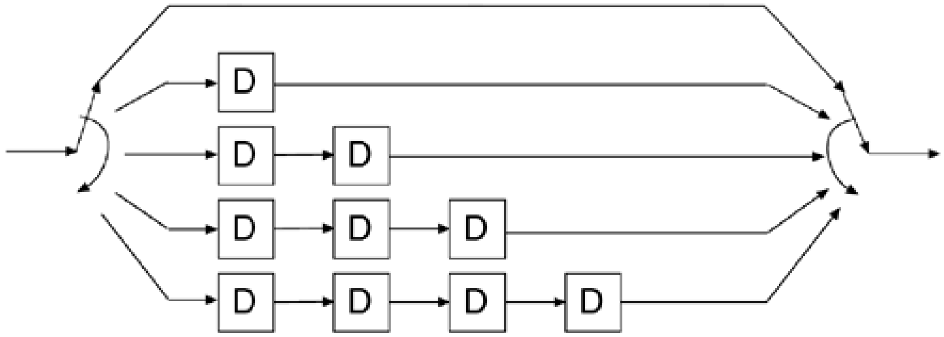


Deinterleaving is the opposite operation of interleaving; that is, the bits are put back into the original order. The deinterleaver corresponding to Figure 3.8 is simply a transpose of the original 8x6 matrix. The deinterleaver is a 6x8 matrix to which the received bits are written into columns and read from rows.

Convolutional Interleaver

A convolutional interleaver is another possible interleaving solution that is most suitable for systems that operate on continuous stream of bits. This interleaver structure was published by Ramsey [12]. Figure 3.9 shows the basic structure of a convolutional interleaver. The interleaver operates by writing the bits into the commutator on the left, and reading bits out from the commutator on the right. The delay elements *D* are clocked after each cycle of the commutators is completed; that is, after the last delay line has been written and read. The main benefit of a convolutional interleaver is that it requires approximately half of the memory required by a block interleaver to achieve the same interleaving depth. This saving can be significant for long interleaver depths.

Figure 3.9. Basic structure of a convolutional interleaver.



Deinterleaving of convolutional interleaver is achieved by flipping the interleaver along its horizontal axis. The structure is otherwise identical except the longest delay line is at the top, and the no delay line is last. Otherwise the deinterleaving operation is identical to interleaving.

Interleaving in IEEE 802.11a

Interleaving is a very important component of the IEEE 802.11a standard. Interleaving depth has been selected to be equal to one OFDM symbol. Therefore it is naturally a block interleaver. The performance effect of interleaving in IEEE 802.11a is a consequence of frequency diversity. IEEE 802.11a is a wideband communications system and experiences a flat fading channel very rarely. This is an essential requirement to be able to exploit frequency diversity. The combined effect of interleaving and convolutional channel coding takes advantage of the frequency diversity provided by the wideband nature of the transmitted signal.

Interleaving depth is only one OFDM symbol, because the channel is assumed to be quasi-static; that is, the channel is assumed to stay essentially the same for the duration of a transmitted packet. Therefore no additional diversity gain can be achieved by interleaving in time. Additionally increasing the interleaving depth would increase the delay of baseband processing; the maximum delay possible is constrained by the IEEE 802.11 MAC level protocol's acknowledgement packet short interframe spacing (SIFS) timing requirements. The SIFS time is equal to $16\mu s$, hence after the packet ends, processing it has to be completed in less time. This is one of the most demanding implementation requirements of the IEEE 802.11a standard.

The interleaving depth measured in bits changes according the used modulation: BPSK, QPSK, 16-QAM, and 64-QAM have interleaving depths of 48, 96, 192, and 288 bits, respectively. The interleaving depth for each modulation is calculated by multiplying the number of data subcarriers by the number of bits per symbol.

Figures 3.10 and 3.11 show BER and PER curves for IEEE 802.11a 12 Mbits/s mode in a Rayleigh fading channel that has 75ns root mean square (rms) delay spread. The effect of the interleaver is most striking for PER, at 1% the gain from interleaver is approximately 5 dB. Figure 3.11 also shows how the slope of the curve is steeper with interleaving; this is a consequence of the diversity gain achieved.

Figure 3.10. Bit error rate of IEEE 802.11a 12 Mbits mode with and without interleaving in 75ns rms delay spread Rayleigh fading channel.

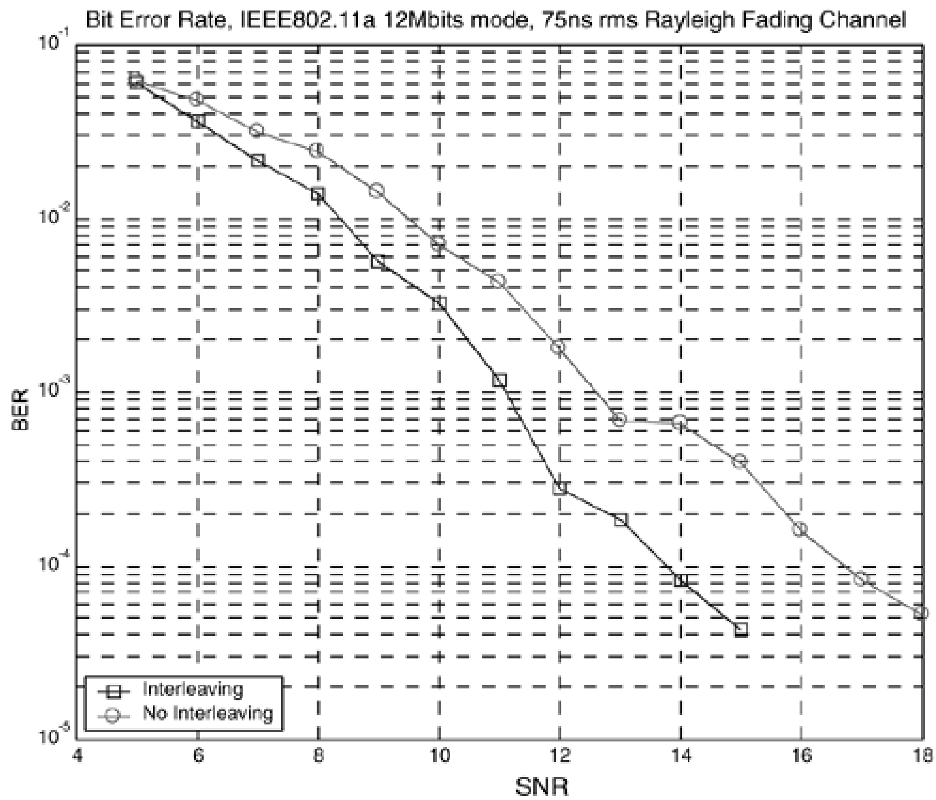
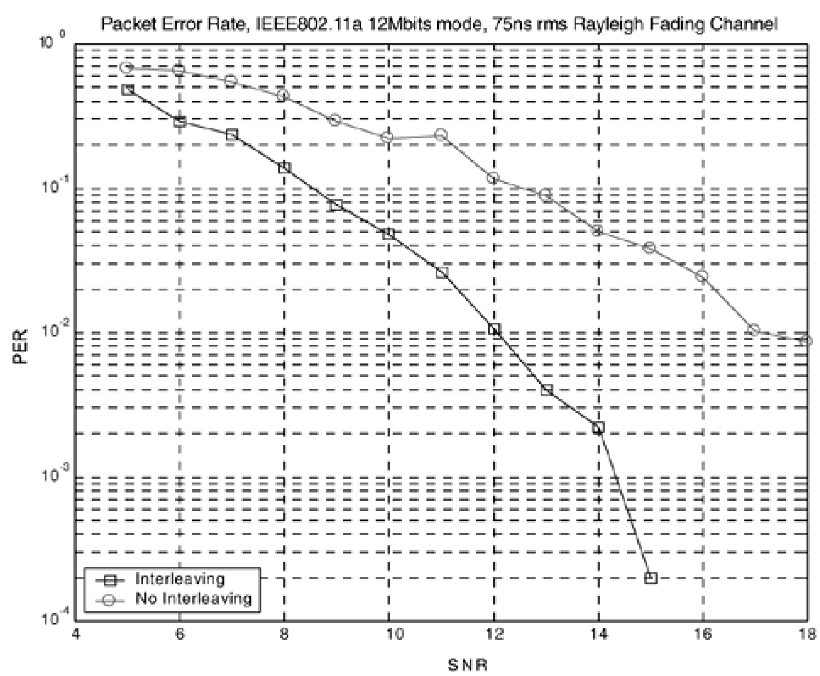


Figure 3.11. Packet error rate of IEEE 802.11a 12 Mbits mode with and without interleaving in 75ns rms delay spread Rayleigh fading channel.

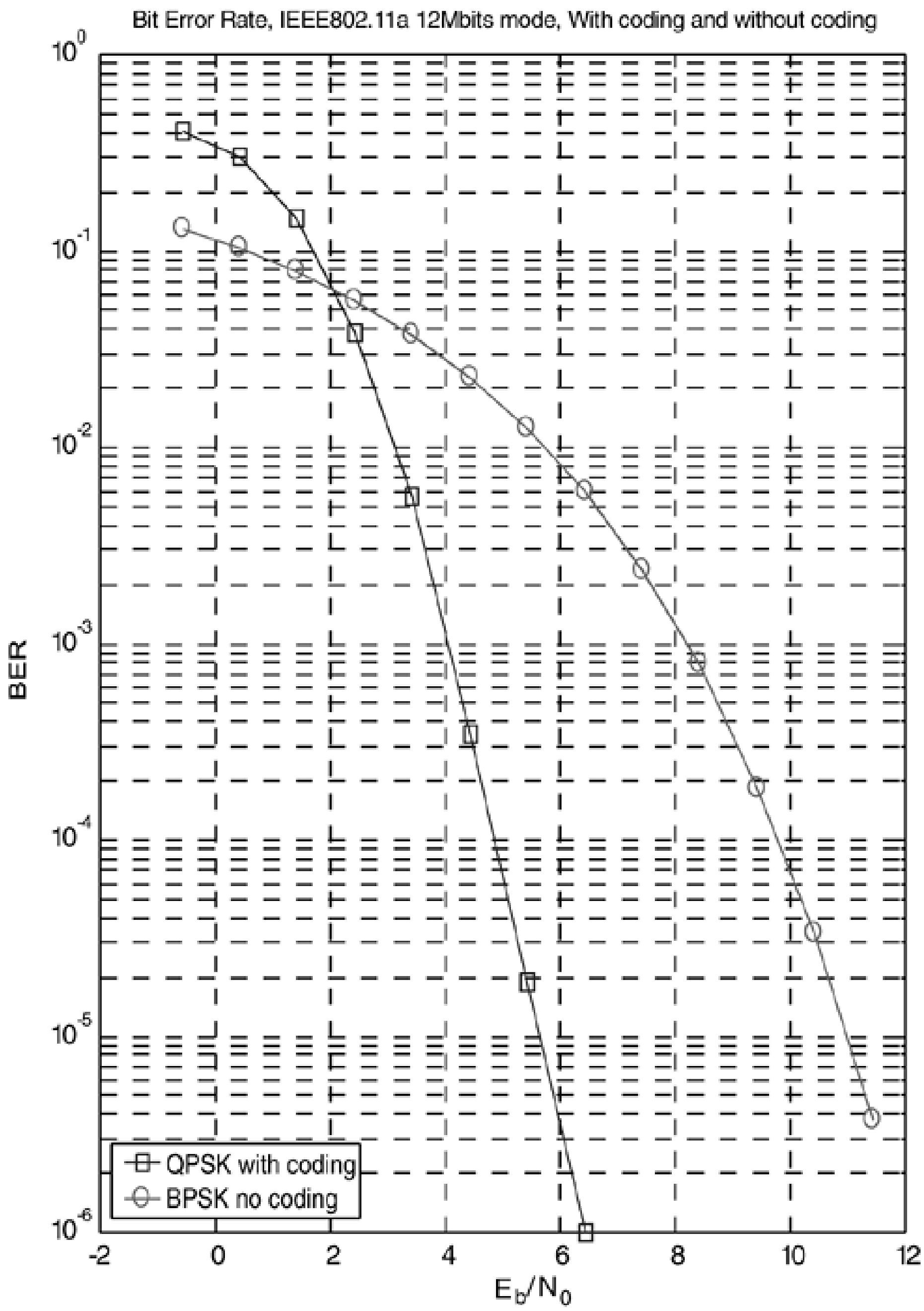


Channel Codes

Channel codes are a very important component of any modern digital communication system, and they make today's effective and reliable wireless communications possible. There are several books that concentrate on different channel coding approaches [7, 9, 10, 17, 20]. The basic measure of channel coding performance is *coding gain,* which is usually measured in dBs as the reduction of required $\frac{E_b}{N_o}$ to achieve a certain bit error rate in AWGN channel. As an example, IEEE 802.11a could use two methods to achieve 12 Mbits/s data rate. The simplest way would be to use uncoded BPSK modulation on each subcarrier, therefore each OFDM symbol would carry 48 bits worth of information. The symbol time is $4\mu s$ or 250000 symbols per second, hence the overall data is 250000x48 = 12Mbits/s.

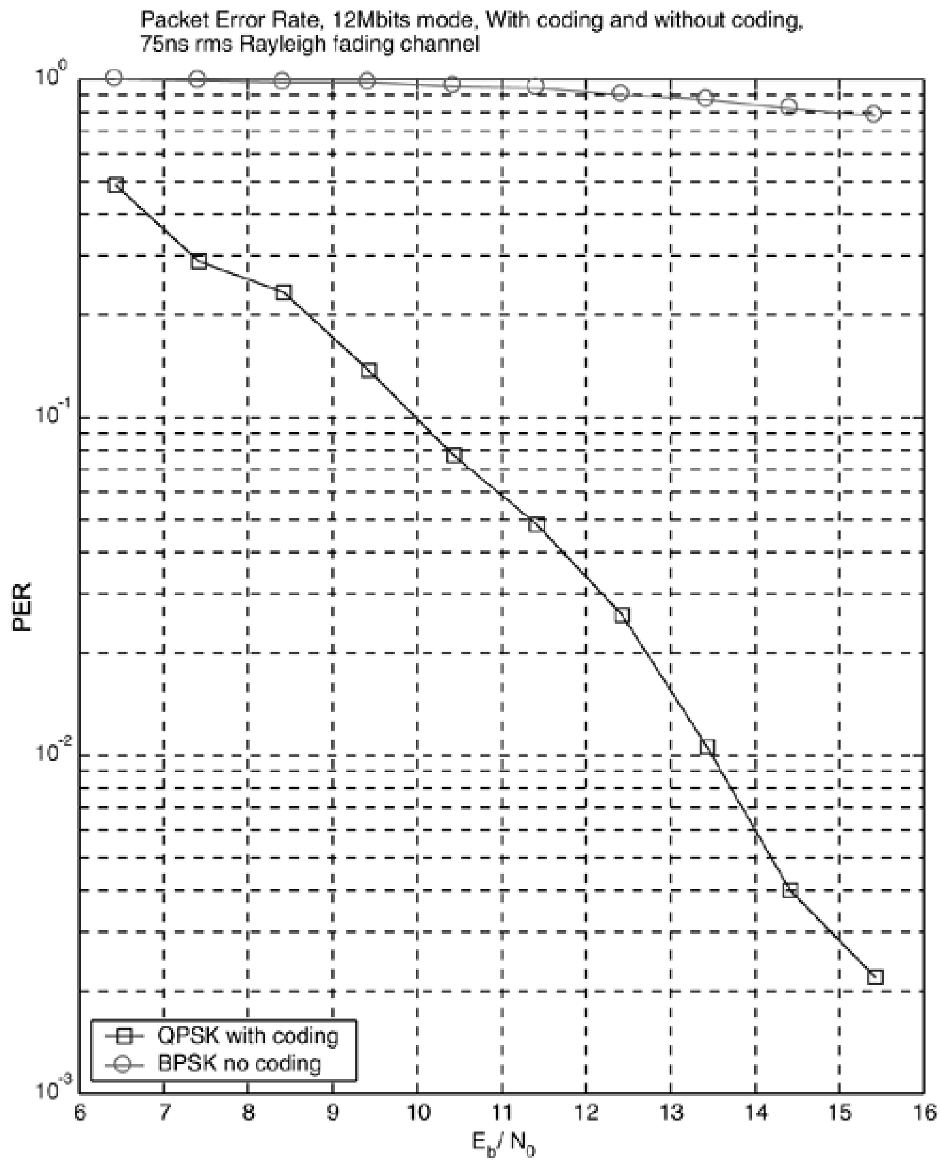
Another way to achieve the same data rate is to use channel coding. The IEEE 802.11a 12Mbits/s mode uses QPSK and rate 1/2 convolutional code. This results in a significantly lower $\frac{E_b}{N_o}$ or *SNR* to achieve a good BER performance. This is illustrated in Figure 3.12, which shows a coding gain of approximately 5.5dB at bit error rate of. 10^{-5}. This means that to achieve the same performance, the system that does not use channel coding has to spend 5.5dB more energy for each transmitted bit than the system that uses channel coding. Figure 3.12 shows another typical behavior of channel codes, the uncoded and coded BER curves cross over at some point. However this happens at so high BER that channel coding practically always provides an improvement.

Figure 3.12. Coding gain in BER of 64 state rate 1/2 convolutional code in AWGN channel.



To make an even more striking demonstration of channel coding in action, Figure 3.13 shows packet error rate curves in a fading channel with the same parameters as the previous example. IEEE 802.11a is a packet-oriented system, hence PER is the most commonly used measure of its performance. In the figure, the curve without channel coding does *not* produce usable PER at all, hence the price paid for implementing channel coding is handsomely paid back. Over the 9dB increase in $\frac{E_b}{N_o}$, the performance of the coded system improves by two orders of magnitude, whereas the performance of the uncoded system only improves from 100% PER to about 80% PER. Flatness of the uncoded PER curves is explained by noting that even one bit error causes a packet error. The system is operating in a fading channel, hence typically some subcarriers experience very low SNR, and bit errors on these bad subcarriers cause packet errors.

Figure 3.13. Coding gain in PER of 64 state rate 1/2 convolutional code in fading channel.



The previous examples of channel coding options show another important parameter of channel coding, namely the *rate* of the code. Code rate is the ratio of bits input, called the *message word,* to the encoder to the bits output, called the *code word,* from the encoder. This ratio is always less than one; in the previous example, rate 1/2 means that twice as many bits are output from the encoder than were input. Therefore the coded system had to use QPSK modulation, instead of BPSK to achieve the same data rate. Channel coding always forces the system to use a larger constellation to keep the same data rate as an uncoded system. Going to a larger constellation reduces d_{min}; this implies higher BER at the output of the demodulator. However, at the output of the channel code decoder, the bit error rate is significantly reduced.

The performance of channel codes is ultimately limited by the channel capacity formula, Equation 3.1. After about 50 years of research, *Turbo-codes* [7] have finally emerged as a class of codes that can approach the ultimate limit in performance. Another innovation, and a very active research area, are Low Density Parity Check (LDPC) codes, which also have performance very close to the capacity.

Convolutional codes have been the most widely used channel code in wireless systems for the past decades, hence this section concentrates on them and the performance of IEEE 802.11a system. Additional channel codes are Trellis-Coded Modulation (TCM) that is closely related to convolutional codes and algebraic Reed-Solomon (RS) codes. These coding schemes are described in sections "Trellis Coded Modulation" and "Block Codes," respectively.

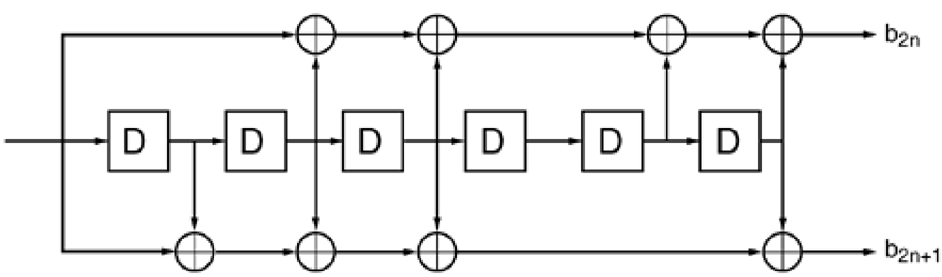
Convolutional Codes

Convolutional codes are one of the mostly widely used channel codes in today's systems; all the major cellular systems (GSM, IS-95) in use today use convolutional channel codes. IEEE 802.11a and HiperLAN/2 WLAN standards also use convolutional error correcting code, and IEEE 802.11b includes an optional mode that uses them. Convolutional codes owe their popularity to good performance and flexibility to achieve different coding rates.

A convolutional code is defined by a set of connections between stages of one or more shift registers and the output bits of the encoder. The number of shift registers k is equal to the nominator of the code rate and the number of output bits n is equal to the denominator. The rate of a convolutional code is generally expressed as $\frac{k}{n}$. The number of connections required to describe a code is equal to the product of k and n. For each output bit there are k connections that define how the value of the output bit is calculated from the state of the shift registers.

For example, Figure 3.14 shows the convolutional encoder used in IEEE 802.11a. This is a rate $\frac{1}{2}$ code with connections 133_8 and171_8. The connections are defined as octal numbers, the binary representations are 001 011 011_2 and 001 111 001_2. The octal notation is used to shorten the expressions, when connections for different convolutional codes are tabulated. From the binary notation, the structure of the encoder is easily constructed. The connections are aligned to the end of the shift register, and a value of 1 means that the shift register stage output is connected to one of the output bits of the encoder using a binary XOR operation. In Figure 3.14 the connection 133_8 that defines values the even indexed bits b_{2n} and the connection 171_8 defines the values of the odd indexed bits b_{2n+1}.

Figure 3.14. 64 state rate 1/2 convolutional encoder used in IEEE 802.11a.



The end of shift register alignment for defining connection values is not universal. Several sources report the connection values by aligning them to the start of the shift register input. Using this notation the connection for the code in Figure 3.14 are 554_8 and 744_8 or in binary 101 101 100_2 and 111 100 100_2. The length of the binary representation is extended with zeros to be a multiple of three, so that the octal values are well defined. Connection values for several different convolutional codes can be found in coding theory text books [9, 10, 20]. These codes were all found using an exhaustive computer search for all possible convolutional codes of the specific rate and number of shift register elements.

The number of shift register elements determines how large a coding gain the convolutional code can achieve. The longer the shift registers, the more powerful the code is; unfortunately, the decoding complexity of the maximum likelihood Viterbi algorithm grows exponentially with the number of shift register elements. This complexity growth limits the currently used convolutional codes to eight shift register elements, and IEEE 802.11a uses only six, due to its very high speed data rates.

The performance of a convolutional code is determined by the minimum *free distance* of the code. Free distance is defined using the *Hamming distance* that is equal to the number of position in which two code words are different. Free distance of a convolutional code is the minimum Hamming distance between two different code words. An asymptotic coding gain at high SNR for a convolutional code can be calculated from the free distance and the rate of the code, as in Equation 3.1.

Equation 3.28

$$coding\ gain = 10\log_{10}(rate \cdot free\ distance)$$

For example, the code used in IEEE 802.11a has a free distance of 10 and rate $\frac{1}{2}$ that gives an asymptotic coding gain of $10\log_{10}\left(\frac{1}{2}\cdot 10\right) = 7.0dB$. However, this is an asymptotic result; Figure 3.12 showed that at 10^{-5} BER, the coding is equal to 5.5dB, and less for lower $\frac{E_b}{N_o}$. The asymptotic result is reached at such a high SNR that practical systems usually do not operate at that SNR region.

Puncturing Convolutional Codes

Usually communications systems provide a set of possible data rates; for example, IEEE 802.11a has eight different data rates: 6, 9, 12, 18, 24, 36, 48, and 54 Mbits/s. Now if the system could only change the data rate by adjusting the constellation size, and not the code rate, a very large number of different rates would be difficult to achieve as the number of constellations and the number of points in the largest constellation would grow very quickly. Another solution would be to implement several different convolutional encoders with different rates and change both the convolutional code rate and constellation. However this approach has problems in the receiver that would have to implement several different decoders for all the codes used.

Puncturing is a very useful technique to generate additional rates from a single convolutional code. Puncturing was first discover by Cain, Clark, and Geist [3], and subsequently the technique was improved by Hagenauer [6]. The basic idea behind puncturing is not to transmit some of the bits output by the convolutional encoder, thus increasing the rate of the code. This increase in rate decreases the free distance of the code, but usually the resulting free distance is very close to the optimum one that is achieved by specifically designing a convolutional code for the punctured rate. The receiver inserts dummy bits to replace the punctured bits in the receiver, hence only one encoder/decoder pair is needed to generate several different code rates.

The bits that are not transmitted are defined by a *puncturing pattern* . Puncturing pattern is simply a set of bits that are not transmitted within a certain period of bits. Figure 3.15 shows the two different puncturing patterns of IEEE 802.11a. The pattern (a) is used to generate rate $\frac{3}{4}$ code from the rate $\frac{1}{2}$ mother convolutional code. This puncturing pattern has a period of six bits, and bits 3 and 4 are punctured (not transmitted) from each period. The *puncturing rate* is equal to $\frac{4}{6} = \frac{2}{3}$ and the overall code rate is equal to $\frac{1}{2\frac{2}{3}} = \frac{3}{4}$, since $\frac{2}{3}$ of the original encoded bits are output from the puncturer. The other punctured code rate of IEEE 802.11a is a rate $\frac{2}{3}$ code. The puncturing pattern is shown in Figure 3.15 (b) has a period of 4 bits, and the fourth bit is punctured, the puncturing rate is, $\frac{3}{4}$ hence the overall code rate is $\frac{1}{2\frac{3}{4}} = \frac{2}{3}$.

Figure 3.15. Puncturing patterns of IEEE 802.11a.

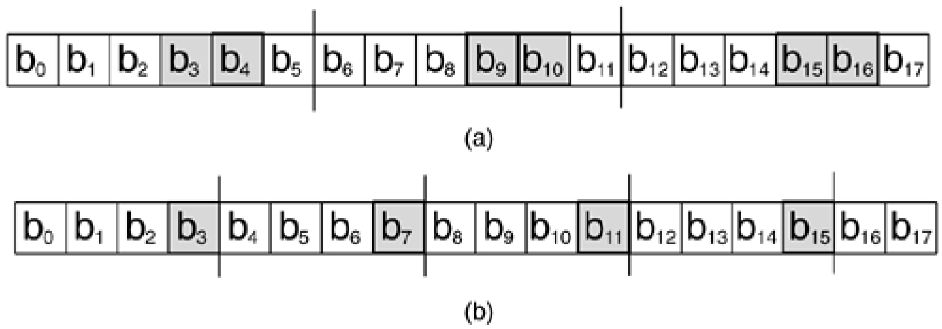


Table 3.4 shows the free distances and the asymptotic coding gains of the three code rates used in IEEE 802.11a. The table also shows the optimum rate $\frac{3}{4}$ and $\frac{2}{3}$ codes; as you can see, the performance loss due to using punctured codes, instead of the optimum ones, is very small. The rate $\frac{1}{2}$ is naturally the optimum code, because the original code is a rate $\frac{1}{2}$ code. Therefore the table does not show the punctured free distance and coding gain values for this rate.

Table 3.4. Free Ddistances of the 64 State Convolutional Codes Used in IEEE 802.11a

Code Rates	*Punctured Free Distance*	*Punctured Coding Gain*	*Optimum Free Distance*	*Optimum Coding Gain*
$\frac{1}{2}$	–	–	10	7.0 dB
$\frac{2}{3}$	6	6.0 dB	7	6.7 dB
$\frac{3}{4}$	5	5.7 dB	6	6.5 dB

Before the punctured code can be decoded, the removed bits have to be inserted back into the bit stream, because the decoder of the original code expects them to be there. Depuncturing is done by simply inserting dummy bits into the locations that were punctured in the transmitter. The values of the dummy bits depend on whether the system uses hard or soft decisions. A hard decision system should insert randomly one and zero bits into the punctured locations. A soft decision receiver inserts a soft decision value of zero. For the usual case of decoding with the Viterbi algorithm, the zero-valued dummy bit does not have any effect on the outcome of the decoder.

Decoding Convolutional Codes

There are several algorithms that can be used to decode convolutional codes. The algorithms vary in complexity and performance, and more complexity means better performance. During recent years, the Viterbi algorithm has reached a dominant position as the method to decode convolutional codes, especially in wireless applications; other methods are practically nonexistent. The reason is that the Viterbi algorithm is a maximum-likelihood code word estimator; it provides the best possible estimate of the transmitted code word. The Viterbi algorithm was introduced in Chapter 1, and it is discussed in detail in any of the coding theory text books mentioned in the start of this section, so we assume you are familiar with it.

The Viterbi algorithm can easily be implemented using either hard or soft decision demodulation. Figures 3.16 and 3.17 show the performance improvement in BER and PER gained by using soft decisions. The gain is approximately 2dB at 10^{-3} BER and 10^{-1} PER. The gain increases to a little bit more than 2dB for higher SNRs. Therefore soft decision decoding is the recommended method to use with Viterbi decoding because the performance improvement it provides does not cost any communications resources.

Figure 3.16. Bit error rate of IEEE 802.11a 12Mbits/s mode in AWGN channel using soft and hard decision decoding.

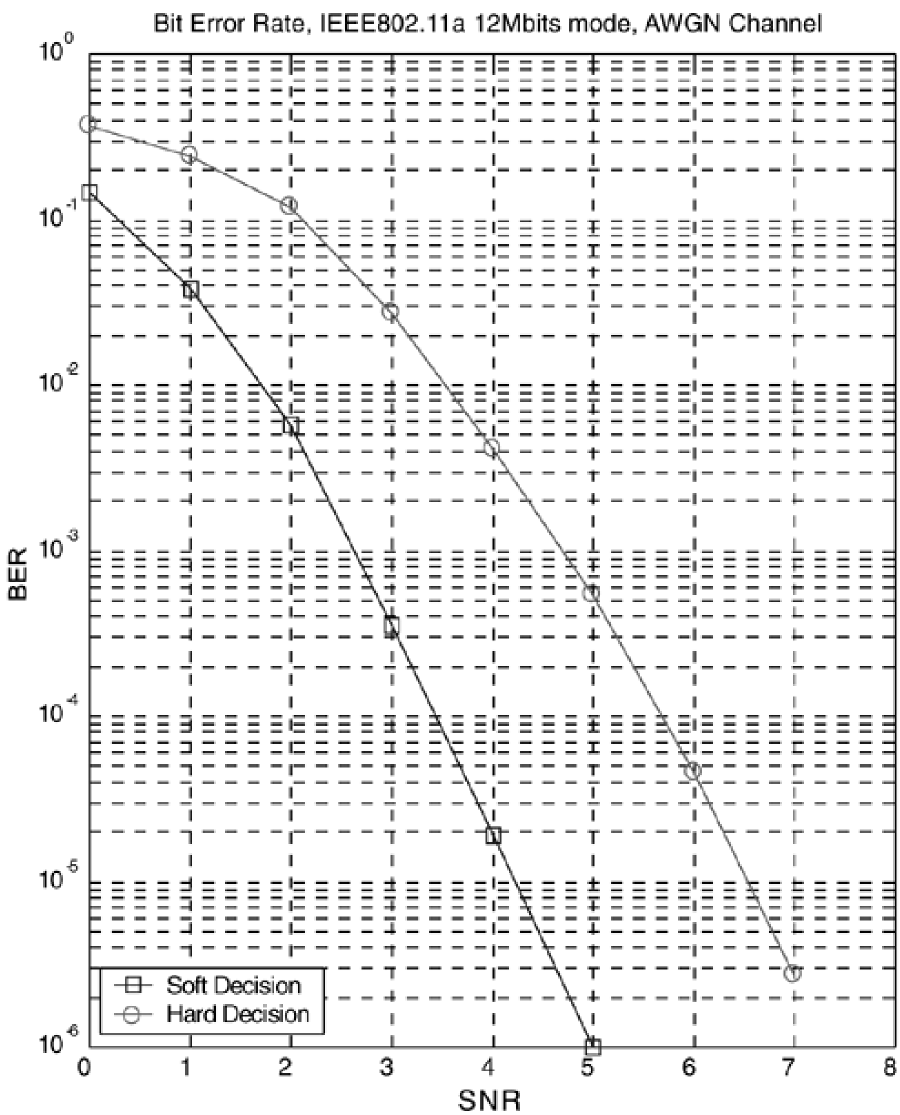
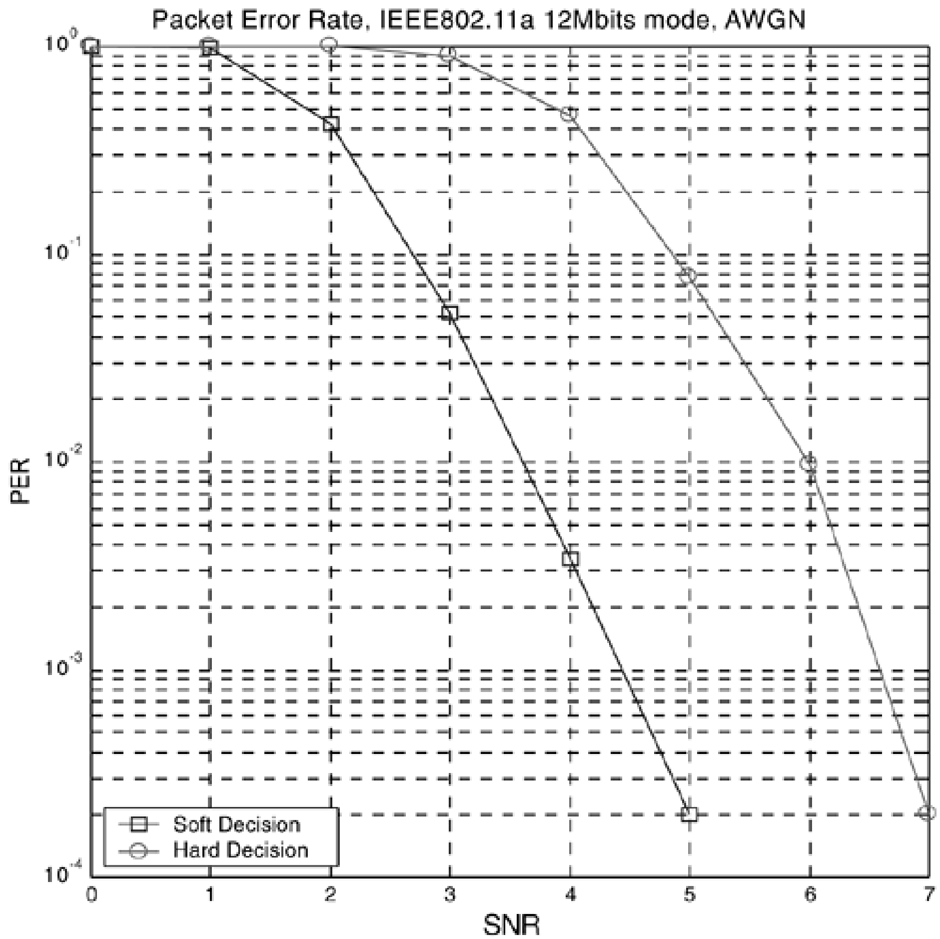


Figure 3.17. Packet error rate of IEEE 802.11a 12Mbits/s mode in AWGN channel using soft and hard decision decoding.



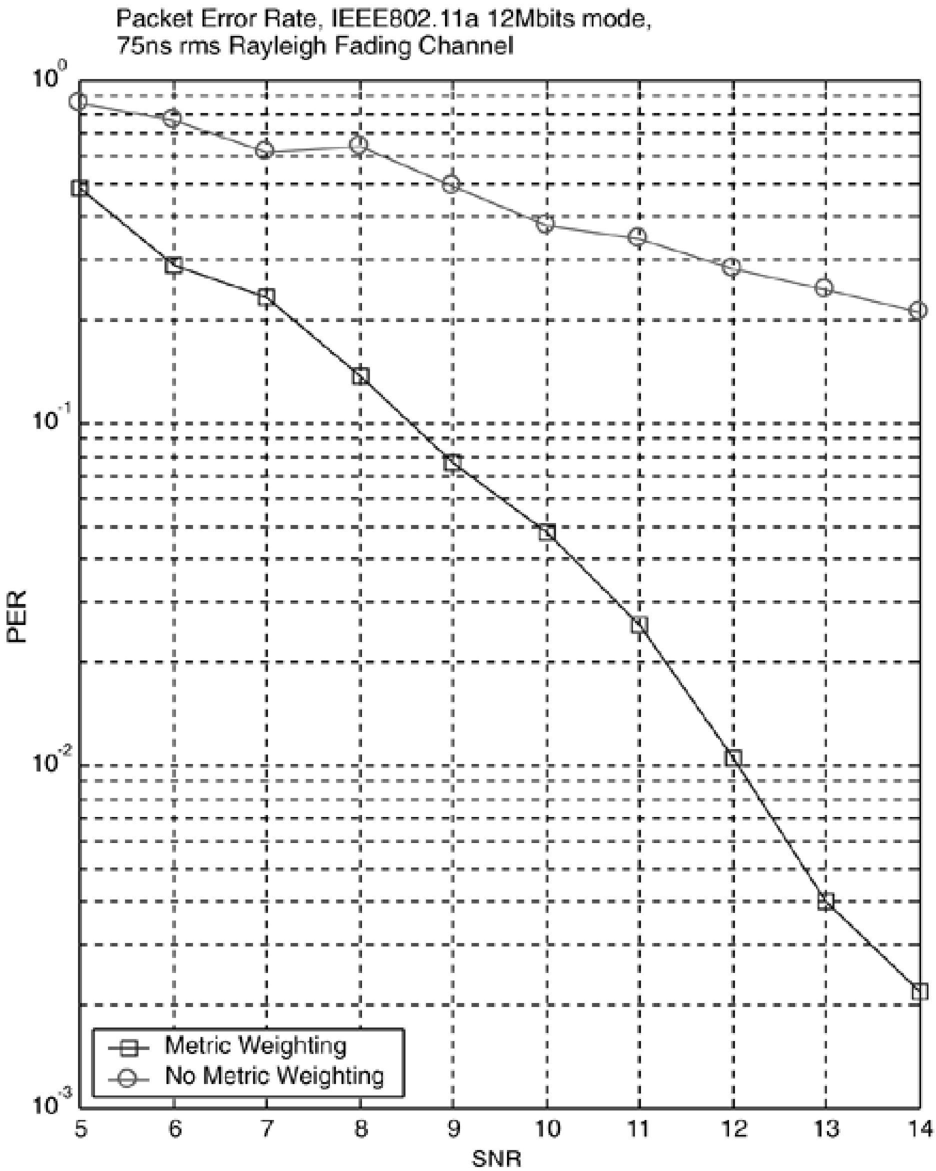
A coherent OFDM system has to estimate the frequency response of the channel. The information about the amplitudes of the individual subcarriers can be incorporated into the Viterbi algorithm to provide performance improvement. Equation 3.29 shows how the path metrics p_n in the Viterbi algorithm are calculated by weighting the squared Euclidian distance between the soft decision $\hat{b}_n$ and the reference value b_n by the squared amplitude of the subcarrier k on which the bit was transmitted.

Equation 3.29

$$p_n = |H_k|^2 \left|\hat{b}_n - b_n\right|^2$$

The performance effect of this weighting is significant in fading channel. Figure 3.18 shows the PER of IEEE 802.11a 12Mbits/s mode when the convolutional code decoding is done with using weighting in Viterbi or by using equal weight for all the path metrics. The impact of weighting on performance is very large. The reason is diversity; when the subcarrier is faded, its amplitude is small and the path metric will be scaled down to almost zero. This means that the bits that were transmitted on bad subcarriers have very little impact on the decision the decoder makes, hence improved performance.

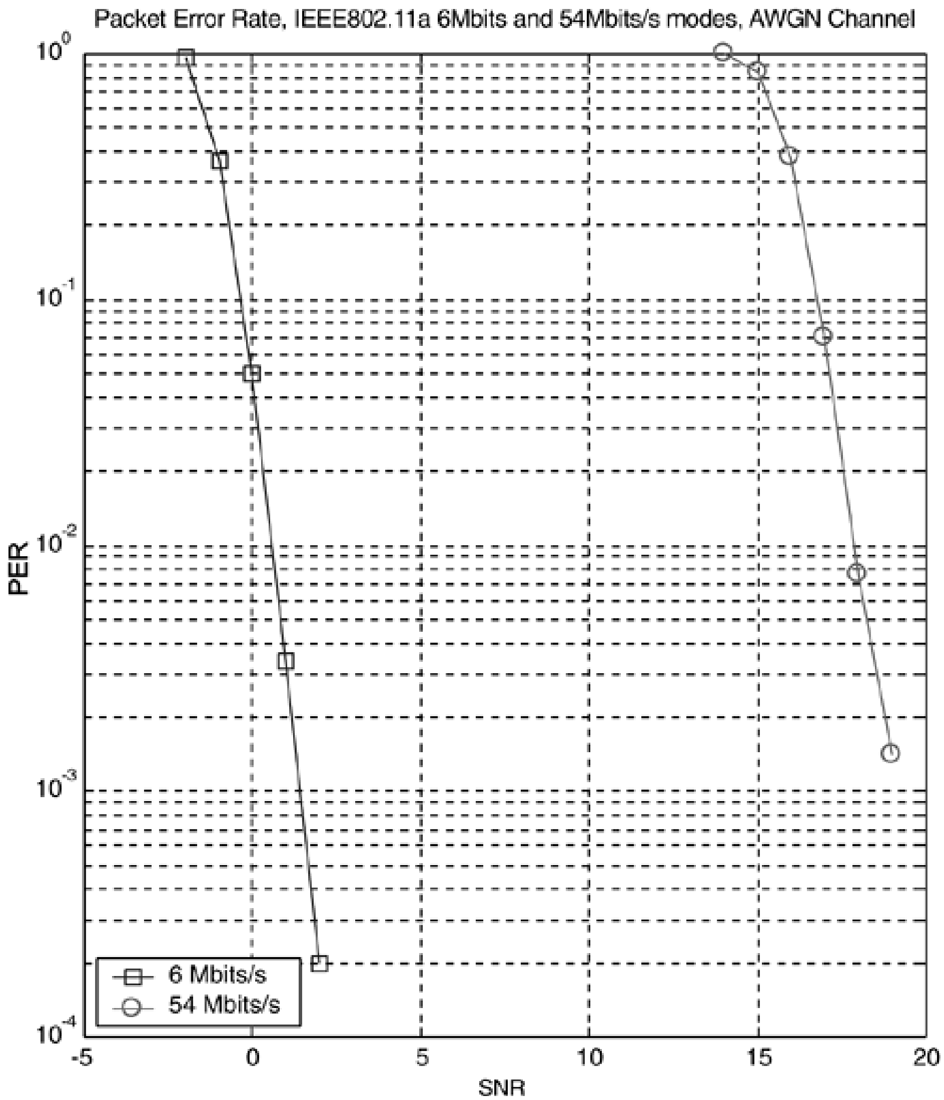
Figure 3.18. Packet error rate effect of metric weighting in Viterbi decoding of convolutional code for 12Mbits/s mode.



Performance of IEEE 802.11a

IEEE 802.11a offers eight different data rates from 6 Mbits/s to 54 Mbits/s with increasingly higher SNR required as the data rate increases. Figure 3.19 shows the SNR required for the 6 Mbits/s and 54 Mbits/s modes in AWGN channel. The total increase in SNR at 1% PER is about 17dB or about fifty fold increase in transmitted power. Another way to interpret the SNR increase is to look at the change of range of the system at different data rates. The range depends on the path loss the transmitted signal experiences, before it reaches the receiver antenna. The overall path loss L_p depends on many factors [18]. The range dependent factor is proportional to the distance d between the transmitter and receiver. The impact of d on path loss depends on path loss coefficient c, that determines how fast the signal power attenuates as it travels. This relationship is shown in Equation 3.30.

Figure 3.19. Packet error rate of IEEE 802.11a 6Mbits/s and 54 Mbits/s modes.



Equation 3.30

$$L_p \sim \frac{1}{d^c}$$

The value of the path loss coefficient is dependent on the environment and a large number of studies have been performed to determine its value in various cases [13]. The basic case is free space where *c* = *2;* in an indoor environment, a value *c* = *3* can be used. We can use these two values to get an estimate of the range difference between the lowest and highest data rate of IEEE 802.11a. When *c* = *2,* every doubling of range decreases the signal power to $\frac{1}{4}$ or by 6 dB, hence the 17 dB SNR range means approximately 7.2 factor in range. Therefore the 6 Mbits/s mode can function for a distance 7.2 times larger than 54 Mbits/s. The absolute range depends on many things, like transmitter power and receiver design. When we assume the indoor value, $d = 3$ the transmitted power decreases to $\frac{1}{8}$ or by 9 dB for every doubling of range. This implies that range difference from 6 to 54 Mbits is about 3.7.

Exercise 3 *The performance of an OFDM system depends on the length of the impulse response of the channel. Simulate different rms delay spread values and note the PER performance. What is the explanation for the results?*

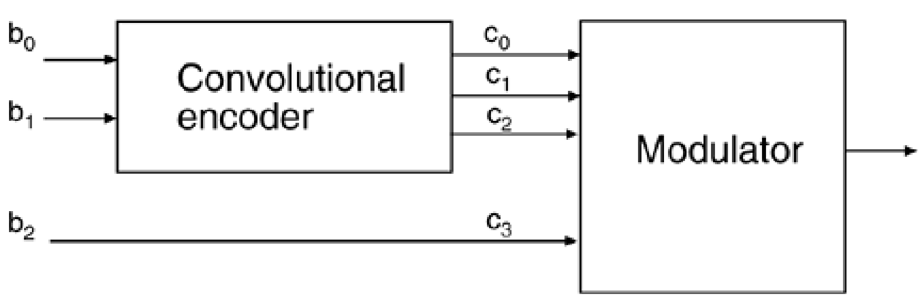
Exercise 4 *Simulate the 9 Mbits/s and 12 Mbits/s modes in both AWGN and fading channel. Compare the results and explain them using the convolutional code free distance and minimum distance of the constellations.*

Trellis Coded Modulation

Trellis coded modulation (TCM) was discovered by Ungerboeck [19]. Trellis Coded Modulation is the main subject of the book by Biglieri et al. [2]. The main idea of TCM is to merge channel coding and modulation into a single integrated component. The benefit of this approach is that the code design is optimized for the used constellation. The most significant benefits of this approach are reached in AWGN channel and with high spectral efficiencies; in other words, with large constellations. One of the first commercial applications of TCM was high speed telephone modems that use very large constellation sizes to reach data rates up 33 kbits/s over the telephone line.

A Trellis Coded Modulation encoder consists of two parts: a convolutional encoder and a modulator. Figure 3.20 shows a basic structure of a TCM encoder. The input bits to encoder are b_0, b_1 ,and b_2.The convolutional encoder in the figure has rate $\frac{2}{3}$, it encodes input bits b_0 and b_1. In general the convolutional codes used by TCM are of rate $\frac{k}{k+1}$. The three bits, c_0, c_1, and c_2, output from the convolutional encoder enter the modulator with one additional uncoded bit, c_3 therefore the overall rate of the TCM code is $\frac{3}{4}$. In general the number of uncoded bits can vary from zero to several bits. These uncoded bits are TCM's Achilles' heel for using it in fading channels with packet radio system. Figure 3.13 showed how the PER really suffers in a fading channel when there are bits, like c_3 in Figure 3.20, that are not protected with channel coding. An error in a single uncoded bit causes a packet error, and the loss of the whole packet. For a practical TCM system to achieve the 54 Mbits/s data rate, it would have to include some uncoded bits in the encoder, thus TCM was not selected as the coding method for IEEE 802.11a. The detrimental effect of the uncoded bits can be reduced by increasing the diversity level of the system. This is discussed in more detail in Chapter 4.

Figure 3.20. Trellis coded modulation encoder.



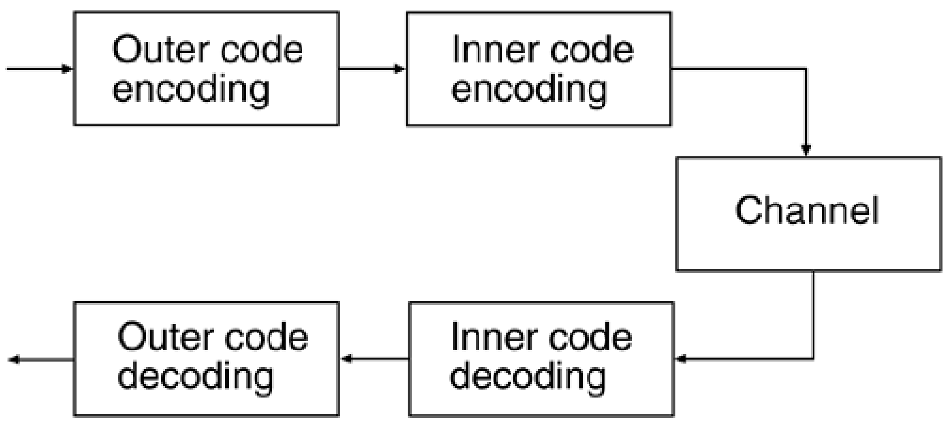
Block Codes

The first error correcting code, discovered by Hamming [5], was a block code. Block codes are different from convolutional codes in the sense that the code has a definite code word length n, instead of variable code word length like convolutional codes. The most popular class of block codes are Reed-Solomon (RS) codes discovered by Reed and Solomon [15]. Another important difference between block codes and convolutional codes is the block codes are designed using algebraic properties of polynomials or curves over finite fields, whereas convolutional codes are designed using exhaustive computer searches. Block codes have not gained wide acceptance in wireless systems, their performance does not reach the level of convolutional codes. The most famous application of Reed-Solomon codes is probably for compact discs that use RS codes to combat burst errors.

Concatenated Codes

Concatenated codes used to be the best performing error correcting codes, before the introduction of turbo codes that are discussed in next section. Concatenated codes are built by combining an outer code and an inner code, shown in Figure 3.21. The outer code is usually a Reed-Solomon block code and the inner code a convolutional code. Concatenated codes have reached performance that is only 2.2 dB from the channel capacity limit.

Figure 3.21. Basic structure of a concatenated channel coding system.



Turbo Codes

Turbo codes were a major discovery in 1993, when Berrou, Glavieux and Thitimajshima [1] released their results of a code that had performance only 0.6 dB from the channel capacity. Turbo codes are a combination of recursive systematic convolutional (RSC) codes, interleaving, and iterative decoding. Heegard and Wicker [7] provide a thorough treatment of these high performance codes. Figure 3.22 shows a basic structure of a rate $\frac{1}{3}$ turbo code encoder. The encoder outputs systematic bits b_{3n} that are identical to input bits, and two sets of encoded bits b_{3n+1} and b_{3n+2} that are encoded by different RSC encoders RSC1 and RSC2. The input to the RSC2 is an interleaved version of the input to the RSC1. The interleaver is an important component of a turbo code, and the code performance improves as the size of the interleaver is increased.