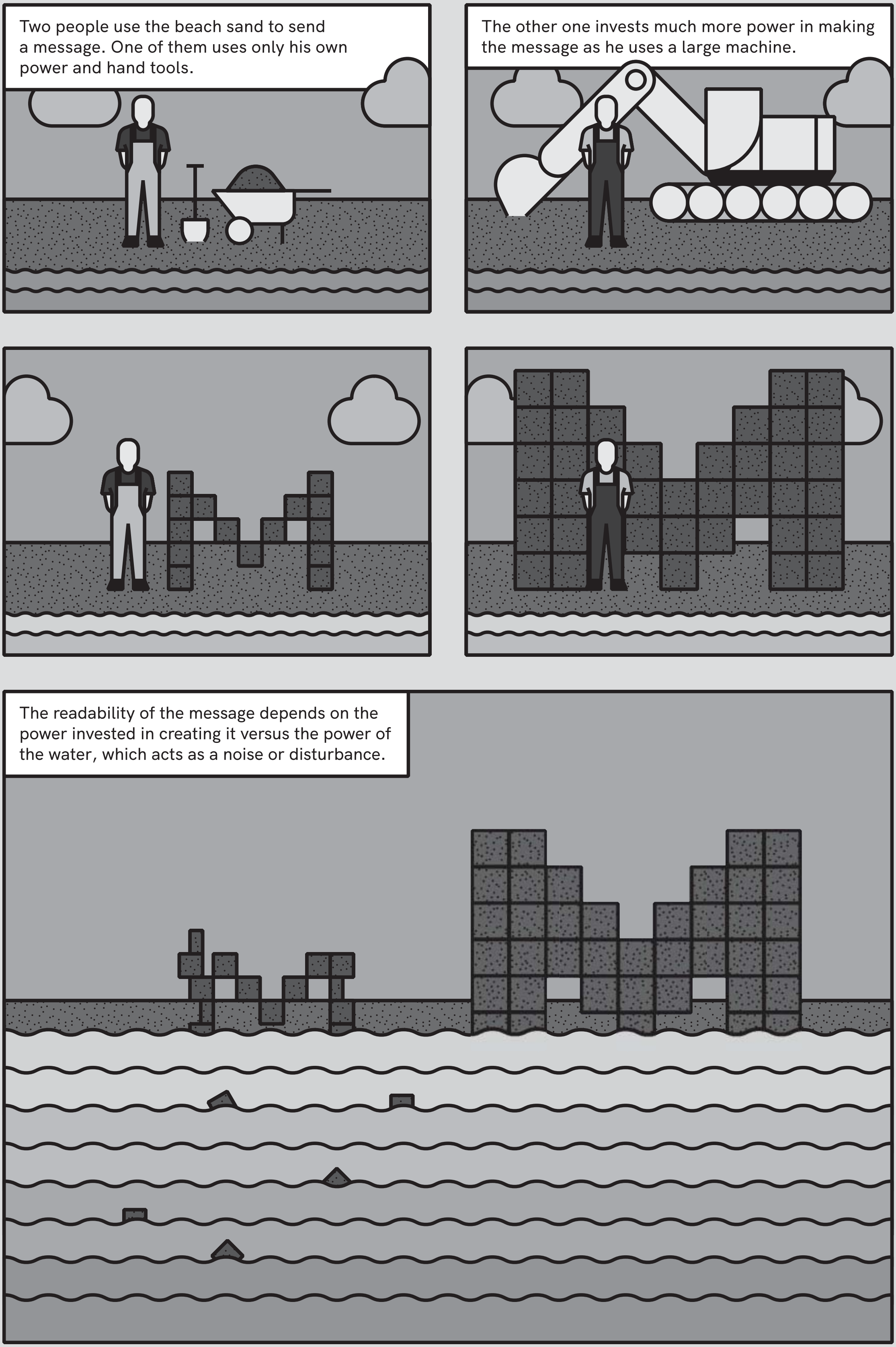

Story by Petar Popovski / Art by Peter Gregson

5

Packets Under the Looking Glass: Symbols and Noise

The information content of a packet can always be described by 0s and 1s. On the other hand, the representation of a data packet may change depending on the layer at which the packet is observed. For example, the packet can be represented by 0/1 at the link layer, while it can be represented via specific voltage levels at the physical layer or as a specific form of a radio wave when sent wirelessly over the air. We use the term *modulation* to denote the way in which information bits are represented in a particular (sub-)module of the communication system.

We refer again to Figure 4.1, describing the modules and layers for a simple one-way link. Once the 1/0 data packet is passed on to the TXmodule, it can undergo several physical transformations. In other words, the layer that contains the TXmodule and the RXmodule consists of several sub-layers and each sub-layer may use a different physical representation of the data. Our objective in this chapter is not to describe all those different physical representations, but rather to discuss the first sub-layer, termed the *baseband layer*. In the baseband layer, the data represented by 0/1 is mapped into complex numbers, which serve as carriers of information. A special case of a baseband layer is the one that uses only real numbers.

In the previous chapters we have dealt with packets and sometimes with packet fractions, but in this chapter we take the looking glass and examine the internal structure of a packet. What we will find there are baseband signals, which are sequences of complex (or real) numbers that are used to represent the information contained in a data packet. So far, we have used the notions of bit, data, and information in a rather informal manner. However, now that we need to describe how information is modulated to symbols, we need to look at the fundamentals of information theory and make these notions more precise.

5.1 Compression, Entropy, and Bit

Using the layering model, it can be stated that the useful data, contained into a *message*, is created at the highest, application layers. This message is then passed on to the lower layers and, finally, it is sent as a wireless, most often radio, transmission. In this context, an important question is the one related to the initial digitalization: how does the system obtain the data bits that constitute a message?

Wireless Connectivity: An Intuitive and Fundamental Guide, First Edition. Petar Popovski.

5.1.1 Obtaining Digital Messages by Compression

While any information can be represented in a binary form using 0s and 1s, the binary representation is not the one that is naturally associated with many information sources, such as picture or audio signal. Information can be understood, for example, as a description of the state of an object. It is intuitively clear that the more states the object can have the more symbols (characters, bits, signs) we need to have in order to describe the state unambiguously. In fact, if Zoya can have 2^B different states, then using B bits she can represent all her possible states in an unambiguous, *lossless* manner. Hence, Zoya can describe her current state to Yoshi by providing him with the B bit values that uniquely describe that state.

However, there is an omnipresent idea in nature and many aspects of human life: using shortcuts for things that are used more commonly or states that occur more frequently. Indeed, much before information theory was conceived, Morse code was built by using the idea that one does not need to always use an equal number of symbols to represent all the letters. Intuitively, fewer bits should be used to describe the states that occur more frequently and, vice versa, more bits are used to describe the states that occur rarely. The motivation is that, when we observe a long sequence of states, the average number of symbols sent will decrease compared to the case in which each state is encoded with a constant number of symbols.

To make this clearer, let us consider a simple example, in which Zoya can observe a system with three possible states, S_1, S_2, S_3 that occur with probabilities $p_1 = 0.5, p_2 = 0.25$, $p_3 = 0.25$, respectively. In order to communicate the state to Yoshi, Zoya can encode it into bits using the following mapping:

$$S_1 \rightarrow 0 \quad S_2 \rightarrow 10 \quad S_3 \rightarrow 11. \tag{5.1}$$

The *average* number of bits used to describe the three states of the system is:

$$\overline{B} = 0.5 \cdot 1 + 0.25 \cdot 2 + 0.25 \cdot 2 = 1.5 \quad \text{(bits/state)}. \tag{5.2}$$

This essentially puts forward *information as a probabilistic concept* or, more precisely, as the average number of bits required to describe the current state of a system, where the state occurrence follows a certain probability distribution.

As a more general case, let the system have S different states that occur with probabilities $p_1, p_2, \dots p_S$. Shannon proved that the minimal average number of bits $\overline{B}$ required to describe this system cannot be lower than the *entropy* of the distribution $p_1, p_2, \dots p_S$, defined as follows:

$$H(p_1, p_2, \cdots p_S) = -\sum_{s=1}^{S} p_s \log_2 p_s \tag{5.3}$$

and is always a non-negative quantity. The entropy for the example used above is:

$$H(0.5, 0.25, 0.25) = 1.5$$

such that the encoding that we have devised in (5.1) is optimal.

5.1.2 A Bit of Information

There is another important feature of the optimally compressed sequences, such as the one that can be produced using (5.1). Let us consider the following array of bits that Zoya sends to Yoshi in order to describe a sequence of states:

$$0010110101001010010 0100 \ldots . \tag{5.4}$$

Encoding is done in such a way that Yoshi can uniquely determine the state sequence as $S_1S_1S_2S_3S_1$ …. Let us pick a random position in the sequence and ask the following: what is the probability that the bit at that position has the value 0/1? The selected position must be one of the following three: (1) the single bit representing S_1, (2) the first bit representing S_2 or S_3, (3) the second bit representing S_2 or S_3. Assume that the observed sequence of states is of length L, where L is a very large number. Then the total number of bits used to represent the sequence L is approximately $B = 1.5L$. Within the sequence of L states, S_1 occurs approximately $0.5L$ times, while S_2 and S_3 occur, each of them, $0.25L$ times. The total number of zeros within the B bits representing the sequence is a sum of:

- $0.5L$ zero bits corresponding to the single bits that represent occurrences of S_1
- $0.25L$ zero bits corresponding to the second bit that represents the occurrences of S_2.

The probability that a randomly picked position b from (5.4) carries a zero bit value is

$$P(b = 0) = \frac{0.5L + 0.25L}{L} = \frac{0.75L}{1.5L} = 0.5. \tag{5.5}$$

In fact, this is a general feature, stating that if the sequence of bits represents an optimally compressed sequence of states, then the bit values 0 and 1 are equiprobable. Furthermore, it can be shown (which we are not doing here) that the binary digits from the sequence (5.4) are independent in the sense that the sequence looks as if it has been generated by flipping a fair coin that results in 0 or 1 with probability 0.5.

This brings us to a meaningful *definition of a bit* as a binary variable that gets value 0 or 1 with probability 0.5. Indeed, one can calculate that the entropy of such a binary random variable is 1. The operation of encoding the source information into a compressed sequence of bits is referred to as *source coding*. In the layered system in Figure 4.1, the source coding operation takes part within the DataSender. Since the focus in this book is what happens with the data bits after the DataSender delivers them to the lower layers, our discussion always assumes that the information sources are compressed optimally and the data bits are equiprobable, such that one data bit carries one bit of information.

It can thus be stated that one bit of information corresponds to the level of uncertainty that one has about the outcome of the random experiment with two possible outcomes, both equally probable. As stated above, when compression is ideal, the messages produced by the source encoder contain "pure" bits, such that each bit carries new information that is in no way correlated to the information contained in another bit of the optimally compressed message.

5.2 Baseband Modules of the Communication System

Figure 5.1 describes the decomposition of the TXmodule of Zoya. In this chapter we will focus on the functionality of the TXbaseband and RXbaseband. Zoya's TXbaseband accepts bits and outputs a complex number, denoted z and called *baseband symbol* or simply *symbol*, which is passed on to the TXRFmodule. After various physical transformations and

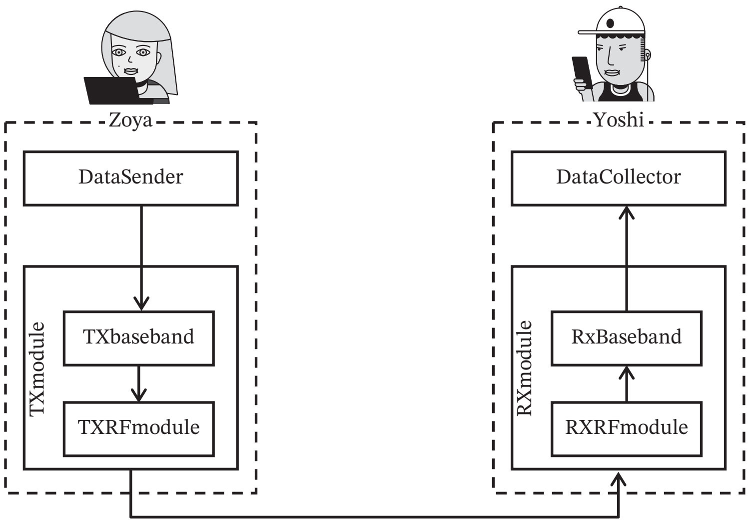

Figure 5.1 A look inside the black boxes of the TXmodule and RXmodule. TXbaseband receives bits from the sender and outputs complex numbers to TXRFmodule, transforms them and send them to the RXRFmodule, which outputs complex numbers to the RXbaseband.

external impacts, the symbol sent from the TXRFmodule arrives at the RXRFmodule of Yoshi. The RXRFmodule passes on a transformed version of z to the RXbaseband module. Yoshi received a complex number, denoted by y, and given by

$$y = hz + n \tag{5.6}$$

where h is the *channel coefficient* and n is the *noise*. Both h and z represent the external factors that affect the communication system. The expression (5.6) is the simplest representative of the *baseband model* or *symbol-level model*[1] of a wireless communication system. More precisely, (5.6) models a single-user system, as there is only one transmitter, but we will see that interference among multiple users is naturally integrated in the model. The TXmodule can send one symbol z each T_s s. It should be noted that the choice of T_s is due to factors that are outside the scope of the baseband model and will be discussed in Chapter 9.

5.2.1 Mapping Bits to Baseband Symbols under Simplifying Assumptions

Let us assume the setup in which $T_s = 1$ (μs, microsecond) and $h = 1$. In addition, let us adopt for a moment the assumption, highly unrealistic, that there is no noise and $n = 0$. In the simplest case, z can have only two complex values, say -1 and $1 + j$. We use -1 to represent the bit value 0 and $1 + j$ to represent the bit value 1. Under these assumptions, Yoshi receives the symbols of Zoya perfectly $y = z$, such that we can say that the baseband layer offers data transfer at a rate of 1 (Mbps, megabits per second).

We can do better than this, by picking four different complex values, say $1, j, -1, -j$ and associate each of them with a 2 bit value. A possible mapping is the one where the bits 00

1 We will use the terms baseband and symbol-level interchangeably, unless explicitly stated otherwise.

are represented by 1, written in short as 00 → 1; the remaining three bit combinations are mapped as 01 → j, 10 → −1, and 11 → j. By using this quaternary mapping, each symbol transmission carries two bits, thereby increasing the data rate to 2 (Mbps).

Following this line of reasoning, one can pick M different complex values, such that each baseband symbol carries $\log_2 M$ bits, leading to a data rate of $\log_2 M$ (Mbps). The condition to get this data rate is that each single symbol is received correctly, thereby implying that *all* $\log_2 M$ bits that are mapped to that symbol are received correctly. However, this involves several strong assumptions:

1. In order to receive useful data bits, Zoya and Yoshi must have agreed beforehand that the communication is about to take place.
2. Yoshi knows the time instants, spaced T_s s apart, at which Yoshi should receive the complex symbols sent by Zoya. In a realistic setting, the symbol sent by Zoya does not instantaneously appear at Yoshi's side, but rather after some delay ΔT.
3. The values of the channel coefficient h and the noise n are perfectly known *in advance* for each received symbol. Note that we have assumed that $n = 0$, but assuming that n is known in advance is equally good, as Yoshi can perfectly recover Zoya's transmitted symbol as $z = \frac{y-n}{h}$.

5.2.2 Challenging the Simplifying Assumptions about the Baseband

These assumptions are too optimistic for any practical setting. The first assumption may look trivial at first glance, but it becomes less so when one asks: how it is possible to have the very initial communication, the one used to exchange the control information or *metadata* and establish the channel? To understand the significance of this problem, consider a binary sequence that Zoya sends e.g. 00101101010010100100100 …. It is legitimate to ask: what had Zoya been sending before the first 0 of this sequence? Obviously there has to be a way to tell Yoshi where the data sequence begins. The problem becomes intricate if all that Zoya can send to Yoshi is 0s and 1s, but there is no special start symbol that is different from 0 and 1. We will return to this problem in Chapter 6, which treats the problem of how a communication channel is defined.

By far the strongest of the three assumptions is the one that the noise n is known. In fact, in the role of n in (5.6) is to model the ignorance about all the disturbances that can affect the reception of symbols at Yoshi's RXmodule. Such an ignorance can be modeled by *random noise*, such that for each symbol the value that is added to hz is a random complex number. Another common assumption is that the random noise for different symbols is *uncorrelated*: the random complex number that represents the noise that affects the ith received symbol is independent of the random complex numbers that represent noise elements added to the other received symbols. The assumption that the noise samples are not correlated is, in a way, the worst-case assumption for the receiver. This is because in the case in which the noise samples for different symbols are correlated the receiver can, in principle, use the knowledge derived from the noise that affects one symbol in order to remove at least part of the noise in another symbol.

An important observation is that, in the process of receiving the desired data from Zoya, Yoshi needs to receive other, auxiliary information that is a pre-condition for correctly receiving the desired data. For example, Yoshi needs to synchronize to the transmissions of Zoya, such that he knows the time instants at which the symbols sent by Zoya arrive. Furthermore, Yoshi needs to perform channel estimation in order to learn h. In practice there is even more auxiliary information that needs to be acquired, such as frequency estimation, but since we have not treated the concept of frequency in detail, we assume that frequency estimation is done perfectly. Recalling the discussion in Section 1.2 on the initial contact and link establishment, it can be stated that such type of initial communication should, in principle, be established by assuming no synchronization and no shared knowledge between Zoya and Yoshi. Hence, the *invite* packet should fulfill this role and be used to gain synchronization at the receiver; let Yoshi learn the channel coefficient and prepare for the subsequent communication.

The communication is further challenged by the fact that Yoshi should occasionally repeat the operation that enabled him to gather auxiliary information. For example, the delay ΔT at which the symbols sent from Zoya arrive at Yoshi may change over time and, if these changes are significant with respect to the value of T_s, Yoshi needs to account for them and perform a re-synchronization. Unless stated otherwise, in our discussion we will always assume that the synchronization is perfect.

The channel coefficient may also change in time due to, for example, movements by Yoshi and/or Zoya, such that Yoshi needs to repeatedly invest resources to estimate the channel. In the quest to receive the desired information, there is always a cost paid in terms of resources that need to be invested to acquire auxiliary information. A common way to acquire the value of h is for Zoya to use several symbols as *pilots* and send dummy data, known to Yoshi in advance, such that Yoshi can extract the value of h.

It is usually considered that the resources spent on pilots are negligible with respect to the resources used for the desired information. However, this is not always the case as the following idealized counter-example shows. Let us assume that h can take only two possible values, $h = -1$ or $h = 1$ and assume that this value changes randomly after each second symbol. If the random change is such that $h = -1$ and $h = 1$ occur with equal probability, then this means that Yoshi needs to receive one bit of auxiliary information every second symbol. On the other hand, if Zoya uses modulation that sends one data bit per symbol, then this means that the auxiliary information is 50% of the useful information that should be received by Yoshi, which is substantial. This example is rather extreme, but puts forward a caution that one cannot always ignore the resources that need to be used to acquire auxiliary information. As we will see later on, the knowledge of h has a decisive impact on how the communication channel is defined and the data rates that can be achieved when communicating over that channel.

5.3 Signal Constellations and Noise

5.3.1 Constellation Points and Noise Clouds

We start with a simple non-trivial case in which $h = 1$ and it is assumed that this fact is known to the transmitter Zoya and the receiver Yoshi. We further assume that the receiver perfectly knows the time instants at which the received symbols arrive. A packet that consists of B data bits is represented at the baseband layer through a sequence of

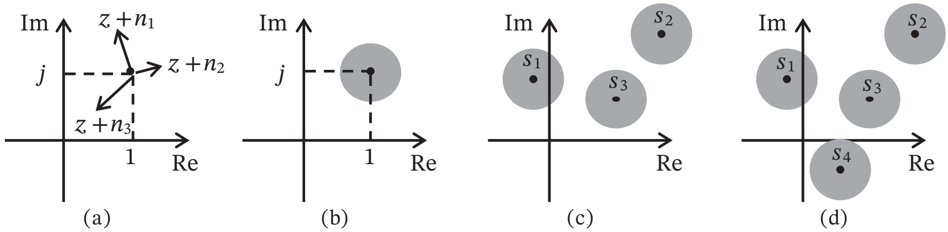

Figure 5.2 Effect of the additive noise complex baseband symbols. (a) Three different received symbols for the same transmitter symbol z. (b) The noise cloud is represented by the circle around z in which noise lies with very high probability p_c, e.g. 0.999. (c) Three example constellation points A, B, C; if binary modulation is used, then A and B should be chosen to represent 0 and 1 (or vice versa). (d) Constellation with $M = 4$ points.

L symbols. The relation between B and L will become clear further in the text. We will use the notation

$$y_i = hz_i + n_i = z_i + n_i \tag{5.7}$$

where $h = 1, i = 1, 2, \dots L$ and z_i is the ith baseband symbol that constitutes the packet sent by Zoya. The noise n_i is additive and it is independent of the value of the transmitted symbol z_i. We will assume that the noise can be described by a stochastic process that is *stationary*, such that for each received symbol the noise sample is drawn from the same probability distribution. It is further assumed that the noise has a mean value 0. This assumption is not limiting, since if the mean value of the noise is some non-zero value μ, then noise would have a constant bias of μ which, over a long period, could be estimated quite correctly and subtracted, leading again to the equivalent case of noise with zero mean value.

We take $1 + j$ as one complex value that is used to represent a specific sequence of bit values. Consider the case in which $z_i = 1 + j$ is sent three times $i = 1, 2, 3$. Since each n_i is a complex number, it is added to the signal, as shown in Figure 5.2(a), resulting in three non-identical received points. Intuitively, noise introduces the highest uncertainty if it is equally likely to move the received point y_i away from z_i in any direction on the complex plane. In other words, the noise introduces the highest uncertainty when it is *circularly symmetric*. Another commonly assumed property of the noise is that occurrence of noise with low magnitude is more likely than the occurrence of noise with large magnitude. This is certainly true for the widely used model of Gaussian noise. Therefore, the impact of the noise can be conveniently represented by a circular *noise cloud* around the correct signal z_i. Given that z_i has been sent, such a cloud represents the area in which the received signal y_i lies with very high probability, for example, $p_c = 0.999$; this is depicted in Figure 5.2(b). The higher the probability p_c, the larger the cloud size.

One of the key issues related to the baseband module is the design of the *constellation* of complex points that are transmitted from the TXbaseband. Each constellation point represents a certain combination of data bits, such that the important question is how to design the bit-to-symbol mapping, which tells us which bit combination should be represented by each particular constellation point. In the simplest case of *binary modulation*, the constellation consists of two points, representing 0 and 1, respectively. Figure 5.2(c) shows three candidate constellation points, s_1, s_2, and s_3. The best pair is s_1 and s_2, since they are the two points that have the largest *Euclidean distance* between them. The reason is the following: since low-magnitude values of the noise are more likely, it follows that a large Euclidean distance decreases the probability that the received symbol will come close to s_1 when s_2 is sent, and vice versa.

If each symbol can have only two complex values, then this symbol can be used to represent a single bit of information. However, there are many, uncountably infinite, complex values out there, so one may opt to choose four constellation points and use them to represent two bits of information, since the number of different states that can be represented by two bits is $2^2 = 4$. We can carry on and say that when we have a packet of B bits we can select a complex constellation that consists of 2^B complex points such that the whole packet can be transmitted with a single baseband symbol.

Where is the catch in this line of thinking? In order to explain this, we need to specify the exact way in which the receiver operates. With a slight abuse of the terminology, let us use the term *transmitted noise cloud* for the noise cloud around the constellation point that has been transmitted. This corresponds to the received noise cloud with channel coefficient $h = 1$. For each received noisy symbol, the receiver determines whether it belongs to any of the noise clouds such that the point that represents the received symbol lies in one of the shaded areas in Figure 5.2. If it does, then the receiver decides that the transmitted symbol that had been sent is the one that is at the center of that noise cloud. If the received symbol does not belong to any noise cloud, the receiver treats that reception as an error.

Let us fix the size of the noise cloud based on the desired probability p_c of the event that the received baseband symbol lies inside the transmitted noise cloud. When adding a new constellation point, we want its noise cloud not to overlap with the noise cloud of another constellation point. If the noise clouds are separated, then one can guarantee that the probability of successful reception of the symbol will correspond to the probability p_c. This is clearly not the case when the noise clouds of two or more constellation points are overlapping. As the constellation points become more separated, the performance of the communication link can be improved in two ways:

1. If the noise clouds are kept fixed and therefore the probability of success p_c is constant, then larger separation implies that more points can be added to the constellation.
2. If the noise clouds are increased and the number of constellation points is kept constant, then the probability of success p_c increases.

The only way to keep the newly added constellation points whose noise clouds do not overlap is to choose complex points z for which the magnitudes $|z|^2$ are large. This increases the average magnitude of the transmitted symbols, which is directly related to the *transmission cost*, something that we have not yet introduced in our communication model. The most common way to introduce transmission cost is to put a limit on the *transmission power* that can be used by the TXmodule. The physical power of a transmitted complex symbol z_i is proportional to $|z_i|^2$. The limit on the physical power can always be converted into a mathematical power limit imposed on the baseband complex symbols, such that we can say that $|z_i|^2$ represents the power of the symbol z_i. Once we take into account the transmission cost, it becomes clear that one cannot use constellation points with arbitrary magnitudes, which puts a limit on the number of different constellation points whose noise clouds are not overlapping.

5.3.2 Constellations with Limited Average Power

For analytical tractability and pedagogical approach to the constellation design, it is usually assumed that the *average power* is limited. However, in practice, the *maximal* power of the transmitter is always limited.

In order to illustrate the constraint imposed by the average power, let us assume that the TXmodule uses four constellation points, which is known as quaternary modulation. Furthermore, let us assume that the constellation be chosen to be as it is shown in Figure 5.2(d). Since each symbol represents two bits of data, the probability of sending any of the symbols s_1, s_2, s_3, s_4 is equal to $\frac{1}{4}$. Then the average power used by the TXbaseband is

$$P_\mathrm{T} = \sum_{i=1}^{4} \frac{1}{4} |s_i|^2. \tag{5.8}$$

Before we proceed to show why the quaternary constellation in Figure 5.2(d) is not particularly good, we make a slight digression about the properties of the noise. An important noise parameter is its average power P_N, which is assumed to be limited. For fixed P_N, it turns out that the noise distribution that leads to the highest uncertainty at the receiver is the Gaussian distribution. Specifically, each noise sample n_i is a complex number drawn independently from a Gaussian distribution with mean value 0 and variance P_N. The most common baseband communication channel is the one in which the input is z_i, the coefficient h is fixed throughout the whole transmission and known by both communicating parties, and the noise n_i has a Gaussian distribution. This channel plays the central role in designing and analyzing communication systems and is termed the *additive white Gaussian noise (AWGN)* channel. We postpone the explanation of the attribute "white" to Chapter 9, in which we discuss the frequency characteristics of the signals.

The complex Gaussian distribution is circularly symmetric and it can move the received symbol away from the transmitted symbol uniformly in any direction. This resonates well with the model of a noise cloud, in which no direction is preferred. The noise cloud is only an approximation of the Gaussian noise and its radius is proportional to $\sqrt{P_\mathrm{N}}$. Let us fix the probability $p_\mathrm{e} = 1 - p_\mathrm{c}$ that the received symbol falls out of the transmitted constellation point. A larger noise power P_N results in a larger noise cloud. For example, if the radius of the noise cloud is taken to be $3\sqrt{P_\mathrm{N}}$, then the probability that the noise falls in the cloud is around 0.95.[2] The Gaussian distribution can potentially produce infinitely high values and there will be always a non-zero probability that the received symbol falls outside of the transmitted noise cloud in Figure 5.2(b). Further, with the Gaussian noise it is always possible that the received symbol falls out of the transmitted noise cloud and lies in the noise cloud of another constellation points such that the receiver can even not detect that an error has occurred.

Based on this, the objective of the constellation design should be: *for given transmit power P_T, noise power P_N, and a given number of constellation points, pick the points in the complex plane such that they are separated as much as possible, thereby minimizing the probability that the noise causes confusion of one symbol with another*. In that sense, the constellation

2 Note that this is due to the complex, two-dimensional Gaussian random variable. For a real Gaussian random variable, the probability that the noise falls within the interval $[-3\sqrt{P_\mathrm{N}}, 3\sqrt{P_\mathrm{N}}]$ is 0.997, the well known "three sigma" number.

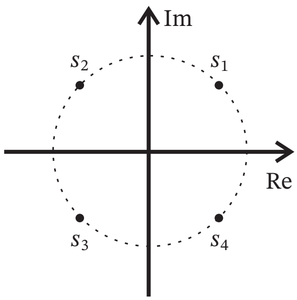

Figure 5.3 The QPSK constellation. All constellation points lie on a circle such that each of them has the same power. The decision region A_i for $i = 1, 2, 3, 4$ is the quadrant that contains the point s_i.

in Figure 5.2(d) is not optimal, as the four points can be separated in a better way, while keeping P_T unchanged.

The best way to separate the four points under a given power constraint is by using the QPSK (quadrature phase shift keying) constellation, depicted in Figure 5.3. Note that any rotation of this constellation works equally well. The decision criterion that has been created by using the noise clouds includes error detection, that is, it announces an error if the received symbol does not fall in any of the possible noise cloud.

It is also possible to make a decision criterion without error detection, which we illustrate as follows. We partition the complex plane into *decision regions* $A_1, A_2, \dots A_M$, where the number of decision regions M correspond to the number of different constellation points. For QPSK we have $M = 4$. The decision regions can be defined by specifying a function $g(\cdot)$:

$$\hat{z}_i = g(y_i) \tag{5.9}$$

that for a given received symbol y_i outputs the decided symbol $\hat{z}_i$, which is not necessarily equal to the transmitted symbol z_i. In this way, each possible received point y_i is mapped to a transmitted constellation point.

A *symbol error* occurs whenever the decided symbol $\hat{z}_i$ is not equal to the symbol z_i that had been originally transmitted. The decision regions are defined in such a way that the probability for symbol error is minimized. According to the common noise property, the received symbol y_i will stay, with high probability, close to the transmitted point z_i. It therefore makes sense to define $g(\cdot)$ such that for a received symbol y_i it is decided that the transmitted symbol $\hat{z}_i$ is the one that has a minimal Euclidean distance to y_i. For a general value of h, $g(y_i) = \hat{z}_i$ such that the distance between y_i and $h\hat{z}_i$ is minimized. For the AWGN channel, assuming that a decision is made for each symbol and that each of the possible symbols occurs with equal probability of $\frac{1}{4}$, the described per symbol decision rule is optimal. For example, for each of the four decision regions for the QPSK constellation, Figure 5.3 corresponds to one quadrant of the complex plane: A_1 is the quadrant that contains s_1, A_2 is the quadrant that contains s_2, etc.

5.3.3 Beyond the Simple Setup for Symbol Detection

Three observations are in order to supplement our simplistic treatment of constellation design, noise, and symbol detection.

Let us first take the case in which the transmitted symbols are not equiprobable. This can happen if the source of information is not encoded into an optimal, compressed sequence of data bits, since in that case each bit would get value 0 or 1 with equal probability. Specifically, let the information source be encoded in such a way that the bit sequence 00, represented by s_1, occurs with probability 0.7. Furthermore, assume that each of the remaining bit sequences 01, 10, 11 and its respective symbol s_2, s_3, s_4 occurs with probability 0.1. Then the constellation should be designed so that s_1 is well separated from the whole group s_2, s_3, s_4, thereby decreasing the probability of error when s_1 is sent. On the other hand, s_2, s_3, s_4 are clustered together such that the probability of error when either of them is sent is higher compared to the probability of error when s_1 is sent. This is an example of *unequal error protection (UEP)*, in which the symbol s_1 is better protected compared to the other symbols. Despite the high probability of error when one of the symbols s_2, s_3, s_4 is sent, the average probability of error can be kept low, as the symbol s_1 occurs more frequently.

The second observation is related to the optimality of per symbol decision. If all the transmitted symbols are independent of each other, then knowing z_i cannot tell us anything about another symbol z_j and vice versa. Since the correspondent noise samples n_i and n_j are independent, then it follows that y_i and y_j are independent, such that it is optimal to make an individual per symbol decision: decide on z_i from y_i and on z_j from y_j. However, this is not true when z_i and z_j are dependent. Let us take the simple case in which each symbol is repeated twice such that $z_2 = z_1, z_4 = z_3, \cdots z_{2i} = z_{2i-1}$, and this is in advance agreed by the transmitter Zoya and the receiver Yoshi. Then the receiver should make a decision by *jointly* considering y_{2i-1}, y_{2i} instead of deciding separately on y_{2i-1} and y_{2i}. Assuming that $z_1 = z_2 = z$ and $h = 1$, it is optimal to combine the received symbols in the following way:

$$y' = y_1 + y_2 = z + n_1 + z + n_2 = 2z + n_1 + n_2. \tag{5.10}$$

Note that the desired symbols have added up coherently: z_1 and z_2 can be seen as two-dimensional vectors, both pointing in the same direction, i.e. the direction of z. On the other hand, the noise samples n_1 and n_2 are uncorrelated and point in random directions with respect to each other. Following the properties of the Gaussian distribution, the resulting noise $n_1 + n_2$ is again Gaussian noise with power $2P_{\mathrm{N}}$, where P_{N} is the power of n_1 and n_2. The new decision function made for y' should output the closest point from a modified constellation: since $2z$ belongs to the set $\{2s_1, 2s_2, 2s_3, 2s_4\}$, the decision function should select one of those four points. Hence the distance between the constellation points doubled, while the radius of the noise cloud increased only $\sqrt{2}$ times to $\sqrt{2P_{\mathrm{N}}}$. Therefore, the decision based on y' will have a lower probability of error compared to any of the per symbol decision for y_1 or y_2 and is the essence of the widely used techniques of *maximum ratio combining (MRC)* and *Chase combining*.

However, one may object to this statement as follows. It may happen that, for a particular instance of y_1 and y_2, the receiver does not make an error if it decides purely to use y_1, but it does make an error if it makes a decision to use y'. This could happen, for example, if the noise sample n_1 is close to zero, while n_2 has a very high magnitude in a wrong direction, i.e. bringing the received sample close to an incorrect received symbol. However, if we consider many received symbols, then the *average* number of errors when the receiver makes a decision by using y' is lower than the number of errors made when the receiver uses only one of the outputs y_1 or y_2.

Finally, the third observation is related to the issue of correlated noise. At first assume that n_1 and n_2 are perfectly correlated $n_1 = n_2 = n$. Then the noise in y' adds up coherently and thus has a power of $4P_N$, i.e. the noise cloud has the power of $2\sqrt{P_N}$. Seemingly, the correlation of the noise does not help in any way. However, if the sender Zoya knows in advance that the noise is correlated, she could send $z_1 = z$ as the first symbol and $z_2 = -z$ as the second symbol such that Yoshi could create the value $y_1 - y_2 = 2z$ and decode a perfect noiseless version of the transmitted symbol. In general, if the transmitter knows how the noise at the receiver is correlated, it can apply communication strategies that utilize this correlation and achieve a lower error probability.

5.3.4 Signal-to-Noise Ratio (SNR)

In the previous examples we have used $h = 1$ for simplicity. Assuming a fixed probability of error, which corresponds to a fixed probability that the received signal falls outside the transmitted noise cloud, then the following conclusions could be derived.

For a fixed noise power P_N, and therefore fixed size of the noise cloud with radius $\sqrt{P_N}$, the link can accommodate more constellation points as the transmit power P_T increases. Alternatively, for fixed P_T, the link can accommodate more constellation points as the power P_N of the noise, and thereby the radius of the noise clouds decreases. Therefore, the number of separated constellation points depends on how large P_T is relative to P_N, rather than the value of P_T in an absolute sense. This leads us to define the *signal-to-noise ratio (SNR)*, a quantity measured at the receiver that serves as an indicator of how many constellation points can be used and is defined as

$$\gamma = \frac{\text{received signal power}}{\text{received noise power}} = \frac{P_T}{P_N}. \tag{5.11}$$

The SNR is a dimensionless, scalar quantity, but it is convenient to express it in *decibels (dB)*. If the scalar value of the SNR is γ, then the dB value is:

$$\gamma_{dB} = 10\log_{10}\gamma \quad \text{(dB)}. \tag{5.12}$$

In general, the received signal is given by (5.6), with a channel coefficient h. Assuming that the receiver Yoshi knows h, he can use the complex conjugate value of the channel coefficient, denoted by h^* and create the quantity

$$y' = h^* y = |h|^2 z + h^* n. \tag{5.13}$$

This is an elementary example of *matched filtering* applied by the receiver. The useful signal $|h|^2 z$ is multiplied by the real quantity (phase-0 complex number) $|h|^2$ and therefore has identical structure as the originally transmitted signal z, except for an amplification factor $|h|^2$. The noise n is not correlated with h, such that h can be treated as a constant when finding the average power of $h^* n$, which in (5.13) is equal to $|h|^2 P_N$. The SNR of the received signal is calculated through:

$$\gamma = \frac{\text{received signal power}}{\text{received noise power}} = \frac{|h|^4 P_T}{|h|^2 P_N} = \frac{|h|^2 P_T}{P_N}. \tag{5.14}$$

Note that in the case of a scalar channel (5.6), one can create the following quantity:

$$y' = \frac{y}{h} = z + \frac{n}{h} = z + n' \tag{5.15}$$

and work with the equivalent channel that has $h = 1$ and noise power $\frac{P_N}{|h|^2}$, which is the variance of the Gaussian random variable n'. In this way the SNR remains as $\frac{|h|^2 P_T}{P_N}$.

5.4 From Bits to Symbols

5.4.1 Binary Phase Shift Keying (BPSK)

BPSK is a modulation method in which one bit is mapped to one baseband symbol. Figure 5.4(a) illustrates the BPSK constellation $\mathcal{S}_B = \{-\sqrt{2}, \sqrt{2}\}$; the average power of this constellation is 2. There are only two ways to map the bits into symbols, the one shown in Figure 5.4(a) and the opposite one, where 0 is represented by $-\sqrt{2}$. Both ways work equally well since, due to the symmetry of the Gaussian noise, they have identical probability of error, irrespective of whether 0 or 1 is sent. In the case of BPSK the probability of *symbol error* is identical to the probability of *bit error*. The probability of error is a function of the SNR, denoted by γ defined in (5.14), and is given by:

$$P_B(\gamma) = Q(\sqrt{2\gamma}) \tag{5.16}$$

where the Q-function is defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{x^2}{2}} dx \tag{5.17}$$

and it decreases with x. More precisely, the error probability $P_B(\gamma)$ goes to 0 as the SNR γ goes to infinity, since in that case the size of the noise cloud relative to the Euclidean distance between the two constellation points goes to zero.

There is one subtlety regarding how SNR is defined in the case of BPSK and its impact on the probability of error. Recall that noise is modeled as a complex random variable that is circularly symmetric and the SNR in (5.14) is defined with respect to the total power P_N. Consider the matched filter in equation (5.13). If BPSK with constellation set $\mathcal{S}_B = \{-\sqrt{2}, \sqrt{2}\}$ is used, then the information-bearing signal that is obtained after the receiver applies matched filter is either $-|h|^2\sqrt{2}$ or $|h|^2\sqrt{2}$. The SNR of the received signal is $\gamma = \frac{2|h|^2}{P_N}$. The noise n can be represented as

$$n = n_{Re} + jn_{Im}. \tag{5.18}$$

It is circularly symmetric, meaning that is has half of its power in the real dimension and half of its power in the imaginary dimension. In other words, n_{Re} is a Gaussian random variable with power $\frac{P_N}{2}$ and is independent of the Gaussian random variable n_{Im} that has also a power of $\frac{P_N}{2}$. Now let us consider h^*n that represents the noise that is relevant and directly affects the decision made by the receiver. The receiver applies a matched filter and, since h^* is a constant complex coefficient that is known by both transmitter and the receiver, it follows that the noise that affects the decision is:

$$n_1 = h^*n. \tag{5.19}$$

Since h is a constant value, n_1 is also a Gaussian random variable that is circularly symmetric, but it has a variance of $|h|^2 P_N$. Following the explanation from above, it follows that n_1

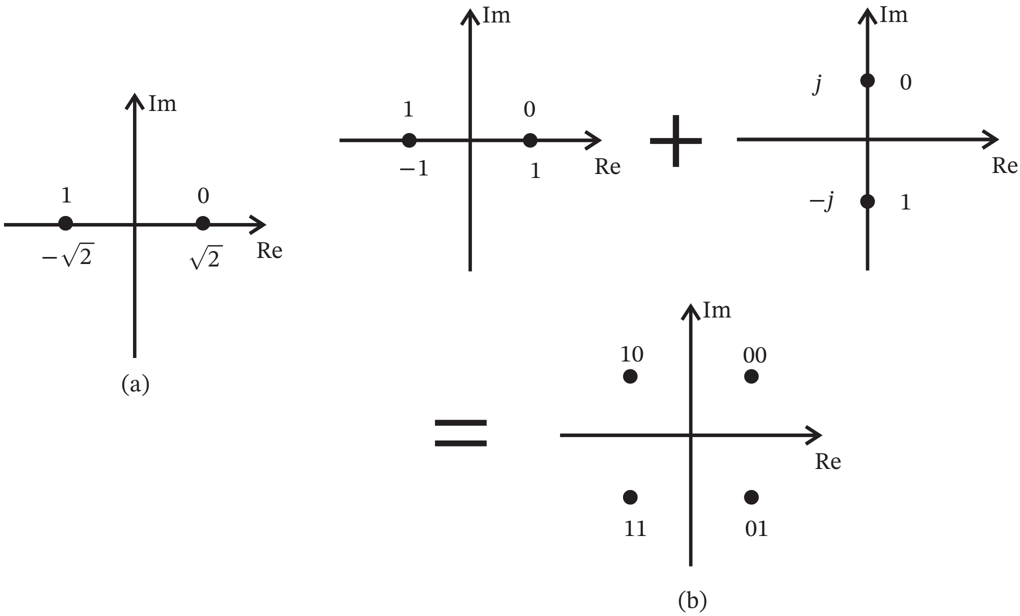

Figure 5.4 Bit-to-symbol mapping for two constellations of power 2. (a) BPSK. (b) QPSK with Gray mapping as a superposition of two orthogonal BPSK signals.

can also be decomposed into real and imaginary components:

$$n_1 = n_{1,\mathrm{Re}} + jn_{1,\mathrm{Im}} \tag{5.20}$$

where $n_{1,\mathrm{Re}}$ and $n_{1,\mathrm{Im}}$ are independent real Gaussian random variables with zero mean and each of them has variance $\frac{|h|^2 P_\mathrm{N}}{2}$. Note that for the BPSK constellation depicted in Figure 5.4(a) only the real noise component $n_{1,\mathrm{Re}}$ can cause error as it can bring the received signal close to the other constellation point. In fact, the noise cloud in this case is *one-dimensional*. The imaginary component $n_{1,\mathrm{Im}}$ is irrelevant, since it always moves the received signal within the same decision region. Therefore only half of the power of the complex noise is relevant for calculating the error performance of a BPSK signal.

5.4.2 Quaternary Phase Shift Keying (QPSK)

We have thus established the fact that the noise in the real dimension does not affect the transmission in the imaginary dimension and vice versa. Therefore, the imaginary component can be used to *multiplex* one more BPSK signal on the same complex symbol, thus leading to a QPSK modulation. Figure 5.4(b) shows how to synthesize a QPSK symbol from two independent BPSK symbols. The obtained QPSK constellation is:

$$\mathcal{S}_\mathrm{Q} = \{s_1, s_2, s_3, s_4\} = \{1+j, -1+j, -1-j, 1-j\}. \tag{5.21}$$

Therefore, the data rate of a QPSK system is equivalent to a double-speed BPSK system, which is a BPSK system whose symbol time is half of the QPSK symbol time and thus there are two times more BPSK symbols per unit time. The probability of each individual bit for given γ remains $P_B(\gamma)$, as given by (5.16). On the other hand, *symbol error* in QPSK is not identical with a bit error. A symbol is received correctly only when both constituent bits are received correctly, such that the probability of symbol error in QPSK at given SNR of γ is:

$$P_Q(\gamma) = 1 - (1 - P_B(\gamma))^2. \tag{5.22}$$

We have obtained the bit-to-symbol mapping in QPSK by synthesizing two independent BPSK symbols. Note that there are 4! = 24 ways in which two bits can be mapped to four symbols. Due to the noise symmetry all these mappings fall into two different mapping classes. The way the two bits are mapped to a QPSK symbol in Figure 5.4(b) is known as *Gray mapping*. Besides allowing the symbol to send two bits independently in QPSK, Gray mapping has another desirable property. In order to explain it, we need to make a digression and define the concept of *Hamming distance*.

Let $\mathbf{b}_1$ and $\mathbf{b}_2$ be two arrays of bits or bit vectors, each consisting of L bits. The Hamming distance between $\mathbf{b}_1$ and $\mathbf{b}_2$ is the number of bit positions in which these two packets differ. For example, if $L = 3$ and $\mathbf{b}_1 = (010), \mathbf{b}_2 = (100)$, then the Hamming distance is

$$d_H(\mathbf{b}_1, \mathbf{b}_2) = 2$$

Hence, if the bit sequence $\mathbf{b}_1$ was sent, but the bit sequence $\mathbf{b}_2$ was received, then the Hamming distance $d_H(\mathbf{b}_1, \mathbf{b}_2)$ is equal to the number of erroneous bits in the received packet. Following the idea that the small values of the noise are more likely than the large ones, then it is more likely that the noise causes less rather than more bit errors. Under such assumption, it is understandable that the number of bits in error should be minimized, as in that case the received data is closest, in terms of Hamming distance, to the data that has been sent. Furthermore, as it will be seen in the later chapters, there are error correction codes that can be used to correct the errors in the received packets and those codes cope better with fewer errors rather than a larger number of errors.

If QPSK modulation is considered, the bit array that is mapped to a single QPSK symbol consists of two bits. In Gray mapping, the Hamming distance between the bit vectors that correspond to neighboring symbols in the constellation is 1. Recall that the probability that symbol s_j has been received when symbol s_j has been sent, under influence from Gaussian noise, is inversely proportional to the Euclidean distance between the two symbols. In the particular QPSK constellation from Figure 5.4(b), if $1 + j$ is sent, then it is more likely to receive $1 - j$ than $-1 - j$ and, if Gray mapping is used, it is thus more likely to have one rather than two bits in error.

In the case of QPSK, one can say that the Gray mapping makes the Hamming distance between the bit vectors proportional to the Euclidean distance between the corresponding symbols and minimizes the average number of bit errors upon transmission of a single QPSK symbol.

5.4.3 Constellations of Higher Order

Each complex symbol consists of two dimensions, real and imaginary, and each dimension can carry information bits independently of the other dimension. In other words, each complex symbol contains *two degrees of freedom* for modulating information. In the case of BPSK and QPSK, each dimension contains only one bit of information. Note that the noise,

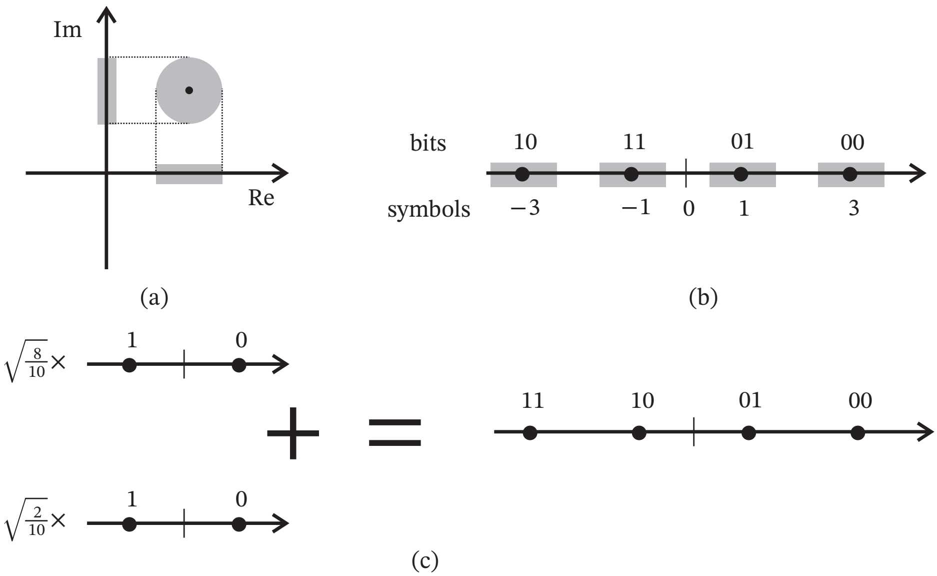

Figure 5.5 (a) The noise cloud is a circle in two dimensions and becomes an interval in one dimension. (b) Pulse amplitude modulation (PAM) with Gray mapping; the noise intervals are marked. (c) PAM as a superposition of two BPSK modulated signals with identical phase.

which is represented by a circular cloud in a two-dimensional complex plane, is projected on an *interval* when a single dimension is considered, see Figure 5.5(a). Recall also that the noise cloud in the case of BPSK is one-dimensional. This interval represents the region that will contain the noise-contaminated received signal in, say, 99.99% of the cases.

We can now use the idea to put more constellation points on a single dimension, which leads to *pulse amplitude modulation (PAM)*. If the noise intervals that are around each constellation point are non-intersecting, then it can be guaranteed that when a symbol is sent, error does not occur in 99.99% of the cases. Similar to the complex case, one cannot extend the constellation points arbitrarily over a single real dimension, as they need to be confined to a finite interval in order to satisfy the power constraint of the transmitter.

Figure 5.5(b) illustrates a 4-PAM constellation that uses Gray mapping of the bit vectors. The average power of this particular constellation is $P = \frac{1}{4}(1 + 3^2 + 1 + 3^2) = 5$. Alternatively, a 4-PAM constellation of power P can be synthesized by *superposition* of two BPSK constellations of power P in the following way:

$$z_4 = \sqrt{\alpha} z_{\mathrm{B1}} + (1 - \sqrt{\alpha}) z_{\mathrm{B2}} \tag{5.23}$$

where z_4 is the equivalent 4-PAM symbol, z_{B1} and z_{B2} are two independent BPSK symbols. The coefficient α is a *power coefficient* with $0 \leq \alpha \leq 1$. Since z_{B1} and z_{B2} are independent, the power P_4 of the 4-PAM constellation is calculated as:

$$P_4 = \alpha P + (1 - \alpha)P = P. \tag{5.24}$$

The process of constructing 4-PAM constellation by superposition is depicted in Figure 5.5(c). Note that both BPSK constellation points z_{B1} and z_{B2} need to lie on the

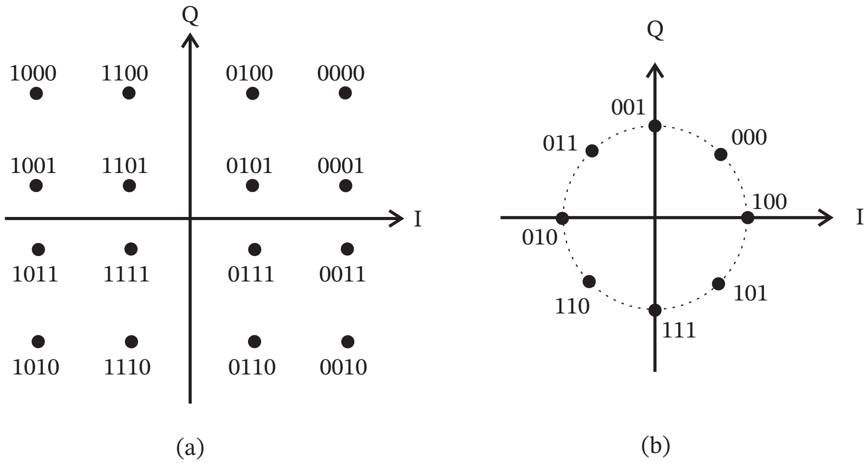

Figure 5.6 Higher-order constellations. Axis labels: I (in-phase) and Q (quadradure). (a) 16-QAM with Gray mapping; (b) 8-PSK with Gray mapping.

same line of the complex plane. In other words, they need to have the same *phase*, which is the angle of the complex constellation point when expressed in polar coordinates. In order to obtain the same constellation from Figure 5.5(b), the power coefficient should be chosen as $\alpha = \frac{8}{10}$ (or, equivalently, $\alpha = \frac{2}{10}$). It is interesting to note the following: no matter how the bits 0 and 1 are mapped on the constituent BPSK constellations, the bit-to-symbol mapping of the resulting 4-PAM constellation is not a Gray mapping. In other words, the Hamming distance between the bit vectors mapped to two neighboring constellation points is not necessarily equal to 1.

The PAM constellation can be applied in each of the complex dimensions and, similar to the way in which QPSK is synthesized from two BPSK signals, the two constellations can be superposed, leading to *quadrature amplitude modulation (QAM)*. Figure 5.6(a) depicts a 16-QAM constellation. In contrast to the superposition in (5.23), the two PAM signals that are superposed to obtain QAM do not have the same phase, but they are rotated by 90° with respect to each other. As a special case, one lies on a real and the other on the imaginary axis. This is identical to the superposition of two BPSK signals onto a QPSK signal. Note that in Figure 5.6 we have labeled the axes *I (in-phase component)* and *Q (quadrature component)*, respectively, which is the standard terminology in digital communication.

Considering Figure 5.6(a), if the bit-to-symbol mapping is made after the constellation is created, then it is possible to obtain a Gray mapping, as the one used in Figure 5.6(a). Nevertheless, it should be noted that the desirable property of Gray mapping (Hamming distance proportional to the Euclidean distance) is preserved only *locally*. For example, the bit vectors 1000 and 1010 have a Hamming distance of 1 and can be considered to be neighbors in the digital domain, while their corresponding symbols are not neighbors in terms of Euclidean distance, observed within the baseband domain. On the other hand, if the 16-QAM symbol is obtained by superposition of two constituent 4-PAM constellations, one as the I-component and one as the Q-component, then it is not possible to obtain Gray mapping; this is similar to the situation in which 4-PAM constellations are created by superposition of two BPSK signals.

There are other possible constellations, out of which we mention the *phase shift keying (PSK)* constellation. Note that not all of the 16-QAM symbols have the same amplitude, while in certain cases the underlying hardware requires that each constellation symbol has the same amplitude. PSK achieves exactly that; see Figure 5.6(b) that depicts the 8-PSK constellation. The name is due to the fact that information is carried solely in the phase of the symbol, not its amplitude.

5.4.4 Generalized Mapping to Many Symbols

A common characteristic of the modulation methods discussed so far is that each complex symbols is modulated in an identical way. For example, in 16-QAM the set of possible 16 constellation points for each symbol is identical and the 4-bit array of a particular value, e.g. 0000, is always mapped to the same constellation point.

A way to generalize this approach is to redefine the way in which the power constraint is taken into account. Namely, the constellations discussed until now satisfy the constraint of the average power calculated per symbol. Let us, instead, consider two complex symbols at a time (u, v) to represent a *hypersymbol* of size two, such that the constellation point is a *vector* of two complex numbers. The power of a specific constellation point of the hypersymbol is $|u|^2 + |v|^2$, while the *average* power per complex symbol per one transmitted constellation point is:

$$P(u, v) = \frac{1}{2}(|u|^2 + |v|^2). \tag{5.25}$$

Since each complex number has two degrees of freedom, the hypersymbol has in total four degrees of freedom. Hence, the design of a constellation that satisfies (5.25) is design of a constellation in a four-dimensional space, under a given power constraint.

For example, one can specify a set $\mathcal{S}$ of 256 possible constellation points, such that for each possible transmitted pair it holds that $(u, v) \in \mathcal{S}$. Each group of 8 bits is mapped to one constellation point and, as the 8-bit groups are equiprobable, each constellation point is transmitted with probability $\frac{1}{256}$. The constellation points are selected in such a way to satisfy the average power constraint P_T per single symbol or $2P_T$ per hypersymbol of size two:

$$\frac{1}{256} \sum_{(u,v)\in\mathcal{S}} P(u, v) = 2P_T. \tag{5.26}$$

We should think about 16-QAM as being *only a special way* to satisfy this constraint by choosing $u \in \mathcal{S}_Q$ and $v \in \mathcal{S}_Q$, where $\mathcal{S}_Q$ is the 16-QAM constellation and the set $\mathcal{S}$ is constructed as $\mathcal{S} = \mathcal{S}_Q \times \mathcal{S}_Q$. In other words, when designing the set $\mathcal{S}$ under the constraint (5.26), the values that u can take are not necessarily identical with the values that v can take. If one finds the best modulation set $\mathcal{S}$, where each constellation points is a vector of two complex symbols and the average power per symbol satisfies (5.26), then this modulation can never be worse than the choice of 16-QAM.

In fact, the best way to pack a given number of points in a four-dimensional space is to use a four-dimensional lattice, rather than a composition of two-dimensional lattices. A more general formulation of this approach can be stated as follows. The hypersymbol can consist of K symbols, such that we need to deal with a $2K$-dimensional lattice. Given that there is a power constraint P_T per symbol, then the power constraint per hypersymbol is KP_T. If one hypersymbol should contain $\log_2 M$ bits, then the size of the constellation set $\mathcal{S}$ is $|\mathcal{S}| = M$. The problem can be formulated as follows: Find M constellation points on a $2K$-dimensional lattice, such that the average power across all those points is less or equal to KP_T. The actual design is quite involved, but it illustrates an important principle: expanding the number of dimensions by spreading over multiple symbols and then designing a constellation in the space with extended dimensions. We will see in Chapters 6, 7, and 8 that this is the main principle underlying the techniques for reliable communication over unreliable channels.

5.5 Symbol-Level Interference Models

The representation of data packets through symbols significantly enriches the models discussed in Chapter 3, and that in itself offers new possibilities for design of communication schemes.

We start by considering how interference or collision is modeled at a symbol level. We refer again to Figure 3.1 from Chapter 3, where we introduced the concepts of capture effect and successive interference cancellation (SIC). For simplicity, we assume that Zoya and Xia are synchronized, such as in slotted ALOHA, and they both start transmitting towards Basil simultaneously. Each transmitted packet requires L complex symbols, which corresponds to a duration of a slot T. If the symbols are sent in a time-division manner, then each symbol has a duration of $\frac{T}{L}$. We can represent the ith symbol received by Basil as:

$$y_{B,i} = h_{ZB} \cdot z_i + h_{XB} \cdot x_i + n_{B,i} \tag{5.27}$$

- $y_{B,i}$ is the ith received symbol at Basil
- z_i and x_i are the ith transmitted symbols from Zoya and Xia, respectively
- n, i is the noise received by Basil during the ith symbol reception
- h_{ZB} and h_{XB} are the complex channel coefficients from Zoya to Basil and from Xia to Basil, respectively.

When there is no danger of confusion, we will drop the index i and write z to denote a symbol sent by Zoya. Note that the situation in which Xia is not transmitting, corresponds to having $x_i = 0$ for all i. In that case, the system model from (5.27) falls back to the single transmitter–single receiver system, described as

$$y'_B = h_{ZB} \cdot z + n_B. \tag{5.28}$$

Another assumption is that h_{ZB} and h_{XB} are constant during all L symbols and, furthermore, these coefficients are known by Basil. Let Basil receive

$$y_B = h_{ZB} \cdot z + h_{XB} \cdot x + n_B \tag{5.29}$$

and let us assume for a moment that he somehow knows the data bits of Xia (we will see in the following two subsections how that could happen). Then Basil can recreate the baseband signal x locally and, since he knows h_{XB}, he can create $h_{XB} \cdot x$, cancel it from y_B and obtain:

$$y'_B = y_B - h_{XB} \cdot x = h_{ZB} \cdot z + n_B. \tag{5.30}$$

In other words, if Basil knows the signal of Xia in advance, then the situation the Basil faces in decoding Zoya's signal is equivalent to the case in which Xia does not transmit/interfere at all. This leads to a seemingly trivial, but often overlooked fact: *the situation in which the information content of an interfering signal is known is equivalent to the situation in which there is no interfering signal at all.*

In the following two subsections we will represent the four scenarios depicted in Figure 3.1 and provide new insights brought by the symbol-level interference models.

5.5.1 Advanced Treatment of Collisions based on a Baseband Model

We first treat the cases depicted in Figures 3.1(c) and (d), where Zoya is closer to Basil compared to Xia, such that Basil receives a stronger signal from Zoya. In contrast to the packet-level model, having the symbol-level interference model (5.27) enables us to represent the stronger signal of Zoya by putting the following condition on the channel coefficients:

$$|h_{\mathrm{ZB}}| > |h_{\mathrm{XB}}|. \tag{5.31}$$

Let us for a moment ignore the received noise and, then given $h_{\mathrm{ZB}}, h_{\mathrm{XB}}$, as well as the constellations used to send z and x, we can speak about a *received constellation*, produced by the superposition:

$$h_{\mathrm{ZB}} \cdot z + h_{\mathrm{XB}} \cdot x. \tag{5.32}$$

Figure 5.7(a) shows the QPSK constellation of power P_{T}, used by both transmitters. Figure 5.7(b) depicts the channel coefficient h_{ZB} and the scaled version of the constellation $h_{\mathrm{ZB}} \cdot z$, while Figure 5.7(c) depicts the scaled version of the constellation $h_{\mathrm{XB}} \cdot x$. Note that the latter constellation is rotated as h_{XB} is a complex number with a phase that is different from 0.

The received constellation has 16 different points, as depicted in Figure 5.7(d). Each of the 16 points in the received constellation can be obtained from a *unique pair* of transmitted symbols $(s_{\mathrm{Z}}, s_{\mathrm{X}})$, where s_{Z} is sent by Zoya and s_{X} is sent by Xia. This means that, if Basil is able to correctly decode the point of the received constellation (5.32), then Basil can correctly decode s_{Z} and s_{X}, both of them simultaneously. This is termed *joint decoding* and it represents a new insight provided by the symbol-level interference model. Recall that the packet-level models treated in Chapter 3 are not capable of representing joint decoding in a similar way. Figure 5.7(d) shows the situation in which the SNR is very high, such that the noise clouds are not overlapping. Hence, all 16 constellation points can be reliably distinguished at the receiver. From each received symbol, Basil can jointly decode the symbols of Zoya and Xia, and these symbols will be correct with high probability[3].

A similar outcome can be achieved by intra-collision SIC. In contrast to joint decoding, which arises as a possibility when we consider a symbol-level interference model, this type of decoding is also possible within the packet-level model, described in Figure 3.1(d), where it is referred to simply as SIC. At first Basil tries to decode the packet of Zoya, since her signal is stronger. For our example, this means that, at first, Basil tries to determine in which quadrant each of the received symbols belong. Determining the quadrant corresponds to

3 We will sometimes use the abbreviation w.h.p. to denote "with high probability".

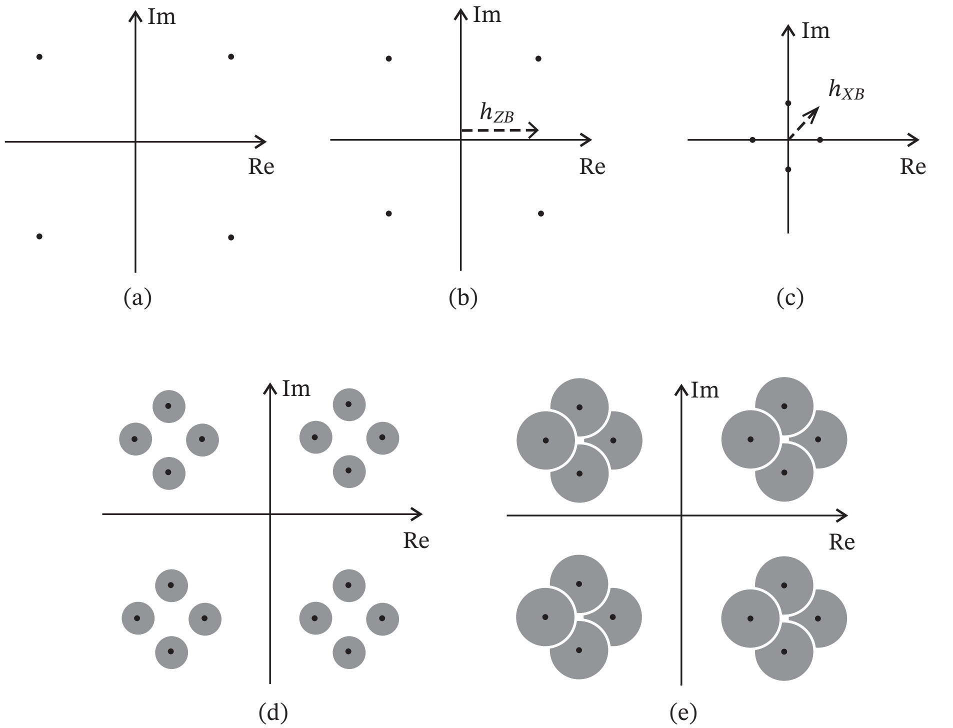

Figure 5.7 Received constellation from two interfering transmitters, each using QPSK signaling. (a) The original QPSK constellation; the power of this constellation is not plotted in proportion to the received constellations and in practice it is much larger than them. (b) The channel coefficient h_{ZB} (dashed) and the scaled constellation $h_{ZB} \cdot z$. (c) The channel coefficient h_{XB} (dashed) and the scaled constellation $h_{XB} \cdot x$. (d) The received constellation under high SNR. (e) The received constellation under low SNR, where it can be seen that $h_{XB} \cdot x$ contributes to an expanded noise cloud.

determining the symbol s_Z sent by Zoya. After all L symbols s_Z are decoded, then Basil verifies if the packet is correct by using error detection (CRC check). If the CRC check is negative, then the process stops and neither the packet of Zoya nor the packet of Xia is decoded correctly. On the other hand, if the CRC check is positive and no error is detected in Zoya's packet, then Basil can recreate all symbols z and, with the knowledge of h_{ZB}, Basil can perfectly cancel the contribution of Zoya in the received signal (5.27). With that, Basil obtains a signal whose ith symbol is given by:

$$y'_{B,i} = y_{B,i} - h_{ZB} \cdot z = h_{XB} \cdot x + n_{B,i}. \tag{5.33}$$

Since this equation also describes a single-user, point-to-point channel, for which we have already described how demodulation/decoding is done, it follows that Basil can decode the packet sent by Xia.

It is important to note that, when SIC[4] is used, the receiver observes that the signal $h_{ZB} \cdot z$ is received with a cloud that consists of noise plus interference and is thus an expanded

4 Unless explicitly stated otherwise, when we use the term SIC, we will refer to intra-collision SIC.

version of the original noise cloud. Hence, we can treat it as an expanded noise cloud, where the interfering signal $h_{XB} \cdot x$ is treated as part of the noise, such that the total noise observed when decoding z_i is $h_{XB} \cdot x + n_B$. In this example, x_i can take four different values, while $n_{B,i}$ is Gaussian, which means that $h_{XB} \cdot x + z_B$ is *not* Gaussian. However, in practice, the mix of noise and interference is often approximated to be a Gaussian random variable. This assumption enables us to approximate the error probability by using the known expressions from the Gaussian case. In this case, the variance of the resulting, approximately Gaussian, noise is:

$$|h_{XB}|^2 P_T + P_N \tag{5.34}$$

where P_T is the power of x_X and P_N is the power of the original noise. Now we can define the *signal-to-interference-and-noise ratio (SINR)*:

$$\frac{|h_{ZB}|^2 P_T}{|h_{XB}|^2 P_T + P_N} = \frac{\gamma_{ZB}}{1 + \gamma_{XB}} \tag{5.35}$$

where the last equation has been obtained by dividing the numerator and the denominator by P_N, while the remaining SNR variables are:

- $\gamma_{ZB} = \frac{|h_{ZB}|^2 P_T}{P_N}$ is the SNR of the received signal from Zoya when Xia is silent $x = 0$
- $\gamma_{XB} = \frac{|h_{XB}|^2 P_T}{P_N}$ is the SNR of the received signal from Xia when Zoya is silent $z = 0$.

Figure 5.7(e) shows a situation of a lower SNR: noise clouds are selected such that the quadrant of the received symbol can be decoded reliably; however, the received point within the quadrant cannot be reliably determined. If the receiver applies SIC, then this means that the data of Zoya will be correctly decoded, and Zoya's signal is subtracted from the received signal in order to obtain (5.33). However, the SNR of the obtained single-user channel is not sufficient to decode the signal of Xia correctly. This corresponds to the *capture effect*, discussed in Chapter 3, where only the packet of Zoya is decoded correctly.

The reader should note that the results of joint decoding and intra-collision SIC are not completely identical. With intra-collision SIC it can happen that the packet of Zoya is not decoded correctly in the first step, such that if the CRC check for the decoded packet of Zoya does not work, Basil does not proceed to decode the signal of Xia. In joint decoding, both packets are simultaneously decoded, while CRC check is run for each of the packets separately; therefore, it may happen that Xia's packet is received correctly even if Zoya's is not, which is never the case with intra-collision SIC.

As a final remark, we have presented the ideas of SIC by assuming perfect interference cancellation. In practice, it can happen that the signal of Zoya cannot be completely removed from the received signal, even if Zoya's data is decoded correctly and her signal is perfectly recreated locally by Basil. One reason for non-ideal interference cancellation can be the non-ideal knowledge of the channel coefficient h_{ZB}.

5.6 Weak and Strong Signals: New Protocol Possibilities

In this section we address the two remaining cases from Figure 3.1. The case in Figure 3.1(b) corresponds to *weak received signals*, where the SNRs γ_{ZB} and γ_{XB} are low, such that even in the absence of interference, the individual signal of Zoya or Xia is very unlikely to be decoded correctly. Using the example with QPSK from Figure 5.7, it follows that the constellation for each of the signals will be received with very weak power. In this case, the corresponding diagram of the joint received constellation, depicted in Figures 5.7(d) and (e), will be completely drowned in noise, such that neither joint decoding nor SIC is likely to produce at least one correct packet.

Figure 3.1(a) depicts the case in which each of the signals is received by Basil as a strong signal, such that the SNRs γ_{ZB} and γ_{XB} of Zoya and Xia, respectively, are high. In other words, the SNR of each signal is sufficient to perform correct decoding when the signal of the other user is absent. However, in the presence of the interfering signal, the SINR for both Zoya and Xia is low, such that intra-collision SIC cannot be initiated. Indeed, the SINR of Zoya and Xia, respectively, is:

$$\mathrm{SINR}_Z = \frac{\gamma_{ZB}}{1+\gamma_{XB}} \tag{5.36}$$

$$\mathrm{SINR}_X = \frac{\gamma_{XB}}{1+\gamma_{ZB}}. \tag{5.37}$$

As an example, if $\gamma_{ZB} = \gamma_{XB}$ is very large, then

$$\mathrm{SINR}_Z = \mathrm{SINR}_X \to 1 = 0 \quad \text{(dB)}. \tag{5.38}$$

In this case the reception of the signals from Zoya and Xia is said to be *interference limited* rather than *noise limited*.

This point is further illustrated by an extreme example in which Basil can have ambiguity about the signals sent by Zoya and Xia. Assume that $h_{ZB} = h_{XB} = 1$ and there is no noise $n_B = 0$, while both Zoya and Xia use QPSK symbols from the constellation $s_Z, s_X \in \{1+j, 1-j, -1+j, -1-j\}$. It can be easily checked that in such a setting, the received constellation has 9 points, rather than 16. If Basil receives $y_B = 0$, then there are four possible symbol pairs (s_Z, s_X) that could have produced that outcome:

$$(s_Z, s_X) \in \{(1+j, -1-j), (-1+j, 1-j), (-1-j, 1+j), (1-j, -1+j)\}.$$

In this case Basil has an ambiguity about the exact pair (s_Z, s_X) that produces $y_B = 0$, such that error-free decoding is not possible. The reader can note that there are other four received constellation points $y_B \in \{2, 2j, -2, -2j\}$ for which Basil has an ambiguity about the symbols transmitted by Zoya and Xia. For completeness, we note that there are four received constellation points for which Basil can recover the transmitted symbols of Zoya and Xia without ambiguity: $y_B \in \{2+2j, 2-2j, -2+2j, -2-2j\}$. However, since both Zoya and Xia are selecting their transmitted symbol in an independent and uniform random manner, there is a high probability of obtaining y_B, which leads to ambiguity and thus a decoding error.

5.6.1 Randomization of Power

Here we look at the implications that our observations about SNR and SINR can have on the design of random access protocols.

In the simple collision model, when two users collide, none of them can increase their chances for correct transmission by deterministically and persistently retransmitting their packet, as identical action from two or more users will continue to cause collisions. For this type of model, the reception is all-or-nothing, such that the only option is that the accessing devices use some ALOHA-like or probing mechanism: apply randomization in selecting the time for retransmission and hope that next time there will be no collision.

Having a symbol-level perspective on the interference, one can think of ALOHA as a protocol that applies randomization in the power domain rather than in the time domain. To see this, consider the example in which Zoya sends her packet to Basil in slot 1 using power P, experiences collision, randomly selected to wait during slots 2 and 3, and retransmit the packet in slot 4. This protocol is originally specified as a randomization in time.

Nevertheless, one can think about it as follows: Zoya has two possible levels for the transmitted power, P and 0. She sends the packet using power P, and gets feedback that the packet has not been received correctly. In slot 2 she flips a coin in order to decide whether to use power P or 0. She selects power 0, which trivially implies that her packet will not be received correctly. As a side note, when Zoya uses power 0 and does not transmit, Basil does not need to send feedback to her and tell her about the unsuccessful reception. She flips a coin in slot 3 and selects power 0 again and finally the coin flip in slot 4 results in a transmission decision, which means that she applies a transmit power of P.

The described power randomization seems to be an unnecessary complication of the ALOHA protocol. However, this impression can be improved by considering that the power can be controlled in a more fine-tuned manner, using multiple rather than only two levels, which leads to new opportunities for protocol design that rely on the symbol-level model. An equivalent of collision in the collision model is a wasted, undecodable interference in the complex baseband model, where each of the two users has a high SNR, but low SINR. Instead of randomizing the time instant of retransmission, both Zoya and Xia can decide to transmit in the next slot, but make a randomized choice of the power level that is used for transmission. This will again result in a collision in the next slot, but the new randomized instances of the transmit power of Zoya and Xia may result in favorable conditions for initiating an intra-collision SIC, such that the collision will eventually not be wasted.

More concretely, assume that for the retransmission Zoya keeps the power at the same level and γ_{ZB} stays the same, while Xia decreases her power and thus attains $\gamma_{XB,1} < \gamma_{XB}$ during the retransmission. Let us assume that $\gamma_{XB,1}$ is chosen in a way that leads to the following properties. $\gamma_{XB,1}$ should be sufficiently low, such that SINR_Z is sufficiently high so that Zoya's packet can be decoded under the interference from Xia. After Zoya's packet is decoded and canceled, then Xia's packet is ready for decoding without any interference. This can be carried out successfully if the choice of Xia's transmit power is such that $\gamma_{XB,1}$ is still sufficiently high.

The previous example is clearly illustrating the fact that, by enriching the collision model into a baseband interference model, the design space for random access protocols expands by allowing fine-tuned randomization in the power domain. This strategy is, in principle, also applicable to the case of weak signals. One can think of a scenario where Zoya and Xia starts sending with minimal power. If no feedback is received from Basil, then one reason is that the signals are too weak for Basil to detect anything. As a next action, Zoya can increase her power to a randomized higher layer, and Xia can do the same, such that the received SNR and SINR can come to a level that can result in capture and, possibly, intra-collision SIC.

5.6.2 Other Goodies from the Baseband Model

The representation at the level of baseband symbols has also implications on the techniques based on inter-collision SIC, described in relation to coded random access in Section 3.4.2. From our discussion in Section 3.4.2 related to the example in Figure 3.4, the reader can get some intuition into why it is possible to have inter-collision SIC. However, the idea of inter-collision SIC becomes much clearer when it is described in the context of an interference model with complex symbols. For the example in Figure 3.4(a), the collision received in slot 1 is represented as an interference between two uplink signals. The received signal in slot 1 is given by:

$$y_{\mathrm{B},i,1} = h_{\mathrm{ZB}} \cdot z_i + h_{\mathrm{XB}} \cdot x_i + n_{\mathrm{B},i,1}. \tag{5.39}$$

The ith received symbol in the collision-free slot 2 is described by

$$y_{\mathrm{B},i,2} = h_{\mathrm{ZB}} \cdot z_i + n_{\mathrm{B},i,2}. \tag{5.40}$$

It is important to note the noise instance $n_{\mathrm{B},i,2}$ in slot 2 is independent and, generally, different from $n_{\mathrm{B},i,1}$. From this slot, Basil decodes packet **Z**, then reconstructs Zoya's baseband symbol z_i and subtracts $h_{\mathrm{ZB}} \cdot z_i$ from the signal buffered from slot 1. Now Basil obtains:

$$y'_{\mathrm{B},i,1} = h_{\mathrm{XB}} \cdot x_i + n_{\mathrm{B},i,1} \tag{5.41}$$

and based on this attempts to decode Xia's packet **X**.

An important advantage of the symbol-level interference model is that it can easily scale to more than two interfering users. If there are K transmitters sending simultaneously to Basil, the ith received symbol is:

$$y_{\mathrm{B},i} = \sum_{k=1}^{K} h_{k\mathrm{B}} x_{k,i} + n_{\mathrm{B},i} \tag{5.42}$$

where $x_{k,i}$ is the ith transmitted symbol from transmitter k, while $h_{k\mathrm{B}}$ is the channel coefficient from transmitter k to Basil. There is still a single noise sample per symbol, $n_{\mathrm{B},i}$.

The concepts of joint decoding, SIC, capture can be generalized to the case of K transmitters and the associated probabilities can be systematically derived starting from (5.42). Without going into detail, we note that when the interfering signals contains contribution from multiple independent transmitters, then one can invoke the central limit theorem and thus justify the treatment of the interference as Gaussian noise. For example, when user 1 is about to be decoded from (5.42), the interference from $K-1$ users where K is large looks like a Gaussian noise, unlike the case in which there is a single interferer.

Finally, we briefly discuss how the model is changed when the packets sent by different transmitters do not necessarily fully overlap. This is another aspect that is very difficult to account for in a packet-level model, while it can naturally be represented in the symbol-level baseband model. Recall that when the model (5.27) was introduced, we assumed that the packets of Zoya and Xia have an identical number of L symbols and both packets start at the same time. Instead of presenting a general model, we provide an example to show what is changed when the previous assumptions do not hold. Let us assume that Zoya's packet consists of $L_{\mathrm{Z}} = 100$ symbols, while Xia's packet consists of $L_{\mathrm{X}} = 40$ symbols. Let us also assume that Xia starts her transmission when Zoya transmits the 11th symbol. Using the Gaussian approximation from above we can state the following:

- The first 10 and the last 50 symbols of Zoya are received with SNR of γ_{ZB}.
- The symbols 11–50 sent by Zoya are received with SINR of $\frac{\gamma_{ZB}}{1+\gamma_{XB}}$.
- All the symbols sent by Xia are received with SINR of $\frac{\gamma_{XB}}{1+\gamma_{ZB}}$.

If the SINR of Xia is high, then Xia's signal is decoded first, subtracted, and then Zoya's signal is decoded free of interference. With our current assumptions about the symbol generation process, all the symbols sent by Zoya are independent, i.e. knowing something about symbol 1 of Zoya tells us nothing about symbol 31 of Zoya. This is changed when *error control coding* is introduced, which is described in the upcoming chapters, where the symbols sent within the same packet do have certain dependency. In this case, a viable approach would be to decode at first the interference-free symbols of Zoya and then use that information, along with the received symbols 11–50, to decode the whole packet of Zoya. If Zoya's packet is decoded correctly, then SIC is started and Xia's packet is decoded in the absence of interference.

5.7 How to Select the Data Rate

The definition of a wireless link given under the collision model, illustrated in Figure 1.1, is very much idealized and the value of the data rate R is not selected based on the physical properties of the model. Furthermore, once R is fixed, it is received perfectly when the receiver Yoshi is within a given distance d from the transmitter Zoya, provided that there is no collision. On the other hand, no data is received correctly when Zoya and Yoshi are separated for a distance larger than d. In Chapter 3 we have softened the link model by introducing probabilistic elements: the data rate R is still fixed, but the probability of receiving the packet decreases as the distance to the receiver increases, which is a better approximation of the physical reality. The symbol-level model allows for a more sophisticated modeling of the distance effects by relating it to the SNR and the probability of error.

5.7.1 A Simple Relation between Packet Errors and Distance

Before examining how the error probability depends on the distance, we need to look closer into the definition of a data rate. Let us consider a packet of L symbols. If we choose to use BPSK, then $D = L$ bits are sent during time T; however, if 16-QAM is chosen, then $D = 4L$ bits are transmitted during the same time. The respective data rates are given by:

$$R_{\mathtt{BPSK}} = \frac{L}{T} \qquad R_{16-\mathtt{QAM}} = \frac{4L}{T} = 4R_{\mathtt{BPSK}}.$$

It should be noted that the data rates exemplified above are *nominal rates*, which reflect what the receiver gets when we assume that there are no errors. The actual data rate depends on the *packet error probability (PEP)*, which is determined according to the modulation applied by the transmitter and the SNR at the receiver. In our current model for packet transmission, the data bits mapped to one symbol are independent of the data bits mapped to the other symbols. Hence, errors occur independently for each symbol reception and in order to receive the packet correctly, the receiver needs to receive correctly *all* symbols belonging to that packet. Note that, if only part of the symbols belonging to the packet are received correctly, then they are of no use, since the packet has a single integrity check that tells whether it has been received correctly or not. In other words, for a packet that is not received correctly there is no some "correctness localization" property that would indicate which bits are received correctly with certainty.

Let P_s denote the probability of symbol error. For a packet of L symbols, PEP is given by the probability that there is at least one symbol error, which is given by:

$$\text{PEP} = 1 - (1 - P_s)^L. \tag{5.43}$$

It is important to note that, for fixed modulation used by the transmitter Basil, P_s increases as the SNR of Zoya decreases.

The baseband model can bring novel insights about how the data rate is affected by the received power of the useful signal. In order to do that, at first we need to identify how the received power depends on the distance. The basic physical law implies that electromagnetic power decreases as a square of the distance from the power emitting source. Nevertheless, as Chapter 10 shows, this simplified law is valid under idealized assumptions: empty space around the emitter, no obstacles or scatterers, etc.

Yet, we can use this propagation law, as well as the speech analogy, to make the following simple approximation: if Zoya is closer to Basil compared to Xia, then a signal sent by Basil will be received by Zoya with a higher power compared to the signal of Basil that is received by Xia. In contrast to the received power of the useful signal, the noise power is a phenomenon related to the receiver and it is independent of the transmitted signal. Consider the situation in which Basil transmits to Zoya and Xia. Assuming that Zoya and Xia have identical receivers, then they have identical noise power, but since Zoya receives a stronger signal from Basil, it follows that the SNR that Zoya has at her disposal to decode the signal from Basil is higher compared to the SNR available to Xia.

Let us assume that Basil fixes the modulation and uses a certain modulation constellation to transmit. This also results in a fixed nominal data rate R. As the distance between Basil and Zoya increases, the power at which Zoya receives Basil's signal decreases. This means that the constellation that is observed by the receiver Zoya, without considering the noise, shrinks towards $(0, 0)$ in the complex plane. In other words, the distances among the constellation points decrease, while the noise power and therefore the noise cloud size stays the same. This results in a decreased SNR. Note that the described, distance-dependent decrease in SNR, is different from the mechanism illustrated in Figure 5.7(d) and Figure 5.7(e), where the received constellation remains constant, while the noise cloud increases. The decreased SNR explains the higher error rate experience by Zoya as she moves to a larger distance from the sender Basil, which worsens the overall data reception.

At this point the reader may raise an objection to the consistency of the uncoded transmission and the packet integrity check, used to obtain equation (5.43). In order to have an integrity check, the data (information) bits should be appended with check bits that are used for error detection. The check bits are calculated based on a certain code, say CRC, and each check bit is a deterministic function of the information-carrying bits. Therefore, a symbol carrying check bits is dependent on the symbols that carry information bits, which could potentially be used to increase the overall probability of successful packet reception.

In this chapter we ignore this possibility and we assume that each symbol is decoded individually at the decoder and mapped onto bits. These bits are not altered further based on the side information that can be received from the dependent symbols, but instead only an error detection check is made and, if it fails, the packet is deemed incorrect. Clearly, this type of operation offers a sub-optimal performance, but the expression for PEP in (5.43) is correct. Otherwise, the dependency among the bits provides the basis for error correction, which is discussed in Chapter 7.

5.7.2 Adaptive Modulation

Having introduced the PEP, the next question is how to select the nominal data rate. Let Zoya transmit to Yoshi using a fixed transmission power and let Yoshi receive the signal of Zoya at an SNR denoted by γ. Zoya has an option to use BPSK, QPSK or 16-QAM and the respective symbol error probabilities are denoted by $P_{\mathrm{B}}(\gamma), P_{\mathrm{Q}}(\gamma)$, and $P_{16}(\gamma)$. Recall that the probability of symbol error is inversely proportional to the Euclidean distance between the constellation points. Therefore, the following relation for the symbol error probabilities holds *for any SNR* γ:

$$P_{\mathrm{B}}(\gamma) < P_{\mathrm{Q}}(\gamma) < P_{16}(\gamma). \tag{5.44}$$

If the number of symbols in a packet is fixed to L, then the previous relation immediately implies that the following holds for any SNR:

$$\mathrm{PEP}_{\mathrm{B}}(\gamma) < \mathrm{PEP}_{\mathrm{Q}}(\gamma) < \mathrm{PEP}_{16}(\gamma) \tag{5.45}$$

for the respective PEPs that can be calculated using (5.43). Thus, at any SNR and for a fixed number of symbols L, the packet sent by using BPSK is the most reliable one and the reliability decreases as the modulation order increases.

Nevertheless, this does not imply that Zoya should always choose BPSK. If Zoya has, for example, a large file of B_{F} bits to send to Yoshi, then Zoya is interested in minimizing the total time required to send the file. The transmission of the large file takes long time, such that the minimization of the total transmission time T_{F} corresponds to the maximization of the average goodput $\mathrm{G} = \frac{B_{\mathrm{F}}}{T_{\mathrm{F}}}$. Observed over many slots, this becomes statistically equivalent to the *expected number of bits* that Yoshi receives correctly in a single time slot of duration T, written as follows:

$$\mathrm{G}_M(\gamma) = \frac{B_M}{T}(1 - \mathrm{PEP}_M(\gamma)) \tag{5.46}$$

where M denotes the modulation used and can have value B, Q or 16, such that $B_B = L$, $B_Q = 2L$, and $B_{16} = 4L$. If Zoya knows γ and can calculate $\mathrm{PEP}_M(\gamma)$, then Zoya compares $\mathrm{G}_B, \mathrm{G}_Q$, and G_{16}, and selects the modulation that provides the highest goodput.

The properties of the PEP are such that there is no single modulation that offers the highest throughput at all SNRs. Figure 5.8 shows a typical relationship of the goodputs offered by different modulation schemes at different SNRs, assuming that a packet has the same length in terms of number of baseband symbols. The main idea behind *adaptive modulation* is that Zoya should adapt her selection of modulation to the current SNR. For the example depicted in Figure 5.8, at low SNR and until SNR $= \gamma_1$ BPSK should be used. For SNR between γ_1 and γ_2, QPSK should be used. Finally, when the SNR is above SNR $= \gamma_2$, 16-QAM should be used. A condition to apply the described adaptation mechanism is that

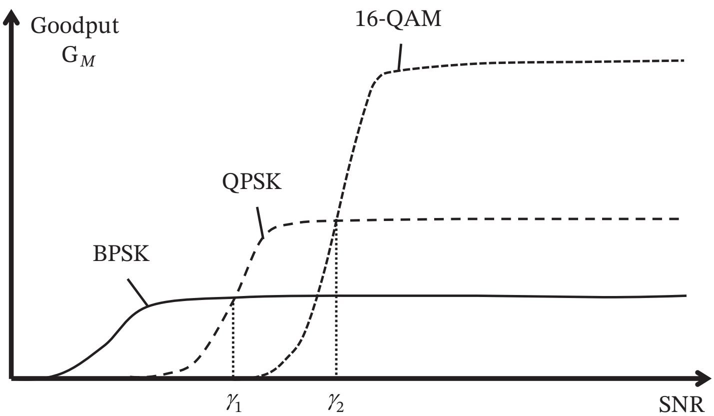

Figure 5.8 Curves of the goodput G_M for three different modulations BPSK, QPSK, and 16-QAM versus the SNR. The SNRs at which the modulation should be changed are denoted by γ_1 and γ_2.

Zoya knows the SNR at which her signal is received by Yoshi. Since, for fixed transmission and noise power, the SNR changes only due to the channel coefficient, we can say that Zoya can apply adaptive modulation only if she has *channel state information (CSI)*, which is in fact the knowledge of γ.

We note that the baseband model works with the assumption that the time period T_s between two symbols is fixed, such that the number of symbols sent over a time interval of duration T is also fixed to $L = \frac{T}{T_s}$. The choice of the symbol time period T_s cannot be described within the baseband model and is based on the physical constraints of the communication signals. This reveals the limitations of the baseband model, which hides the underlying physical phenomena and the associated analog signals that are used for communication.

Another interesting factor related to adaptive modulation is the packet length. When defining the rate in (5.43), we have assumed that the packet length is fixed in terms of number of symbols L, rather than number of data bits. If the data packet is fixed in terms of number of data bits, then different modulations will result in different number of baseband symbols. For example, if the packet contains D bits, then this corresponds to $L_B = D$ symbols with BPSK, $L_Q = \frac{D}{2} = \frac{L_B}{2}$ symbols with QPSK and $L_{16} = \frac{L_B}{4}$ symbols with 16-QAM. Using again (5.43) to calculate the PEP, it is seen that now not only the probability of symbol error P_s changes, but also L changes: the higher the modulation order, the higher the P_s, but the lower the L. While a higher P_s increases PEP, a lower L decreases it. This will lead to different thresholds for adaptive modulation, $\gamma_1, \gamma_2, \gamma_3$ from Figure 5.8, compared to the case when the thresholds are determined by assuming a fixed number of baseband symbols per packet.

These observations can be used to optimize the packet length along with the modulation in a particular setting. The advantage of using a shorter packet when the modulation order is higher should be applied cautiously, since there is a lower limit to the packet length and that is determined by the amount of *overhead* in the packet. As discussed in the previous chapters, not all the bits in a packet are actual information carrying bits, but some of them are used to carry signaling information, such as source/destination address, CRC bits, etc.

As the packet length decreases, the percentage of the overhead increases, thus contributing to the decrease in the goodput.

5.8 Superposition of Baseband Symbols

In the way we have used modulation constellation until now, a group of bits is mapped to a constellation point and the total power of the baseband symbol is used to carry this constellation point. However, the baseband model, unlike the packet-level models, offers the possibility to split the total power and allocate it to multiple independent streams of data bits. This leads to the idea of *superposition coding*, a baseband procedure in which the total available power is allocated to multiple independent data packets and these packets are transmitted simultaneously.

In order to illustrate the basic idea, we can start from equation (5.23), which involves superposition of two simplest modulated signals (BPSK) and generalize to the superposition of S independent data packets. Each of those packets consists of L baseband symbols. The ith symbol transmitted by Zoya can be represented as:

$$z_i = \sum_{s=1}^{S} \sqrt{\alpha_s} z_{s,i} \tag{5.47}$$

where $\sqrt{\alpha_s} z_{s,i}$ is the contribution to the transmitted symbol from the ith symbol of the packet belonging to the sth data stream. The coefficient α_s stands for the fraction of total power that is allocated to the sth packet. Since the transmission power of Zoya is limited, the following must be satisfied:

$$\sum_{s=1}^{S} \alpha_s = 1. \tag{5.48}$$

More generally, the sum of all alphas in the previous equation needs to be less or equal to 1. However, it is intuitively clear that we should be using all the available power in order to get the highest throughput. It should be noted that the selection of the modulation for the sth packet z_s can be independent from the modulation used for the other packets.

Figure 5.9 illustrates a simultaneous transmission of three packets from Zoya to Yoshi by using superposition coding. Assume, for simplicity, that the channel coefficient between

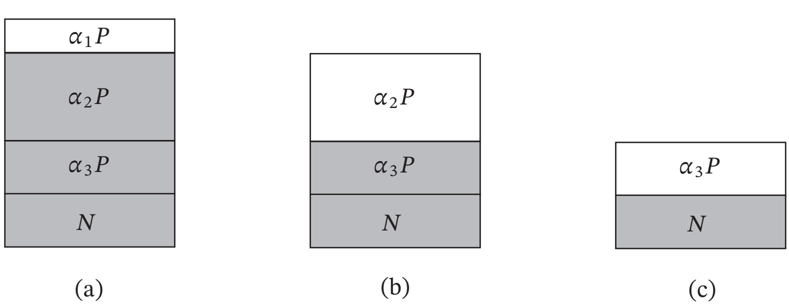

Figure 5.9 Three superposed packets with powers $\alpha_1 P$, $\alpha_2 P$, and $\alpha_3 P$, while N is the noise power. For each decoding step in SIC the shaded boxes represent the power of the noise. (a) Decoding of packet 1. (b) Decoding of packet 2. (c) Decoding of packet (3).

Zoya and Yoshi is 1, such that the symbols received by Yoshi have the same power as the symbols sent by Zoya. Decoding is done by SIC. Yoshi decodes the first signal z_1, with power $\alpha_1 P$, by treating z_2 and z_3 (powers $\alpha_2 P$ and $\alpha_3 P$, respectively) as additional contributions to the noise. The total noise and interference that is perceived during the decoding of z_1 is a sum of the powers of z_2 and z_3, as well as the power of the true noise n. The SINR available for decoding z_1 is:

$$\mathrm{SINR}_1 = \frac{\alpha_1 P}{\alpha_2 P + \alpha_3 P + N}. \tag{5.49}$$

After z_1 is decoded, it is cancelled and the receiver proceeds to decode z_2, but now the SINR is:

$$\mathrm{SINR}_2 = \frac{\alpha_2 P}{\alpha_3 P + N}. \tag{5.50}$$

Finally, after z_1 and z_2 are decoded and canceled, z_3 is decoded in the presence of noise only, such that the SINR is in fact the SNR and is equal to

$$\mathrm{SINR}_3 = \mathrm{SNR}_3 = \frac{\alpha_3 P}{N}. \tag{5.51}$$

As the modulation of each superposed packet can be chosen independently, it follows that the data rate of each of those packet can be different. Recalling the principles of adaptive modulation explained in the previous section, the data rate R_s of the sth data packet is selected based on the power allocated to that packet as well as the power of the noise/interference that affects its decoding. In this case, the additional noise is the sum of the powers of $z_{s+1}, z_{s+2}, \dots z_S$, and n. If all the data packets are decoded correctly, the achieved data rate is

$$R = \sum_{l=1}^{S} R_s. \tag{5.52}$$

However, if during the decoding procedure the first $K-1$ packets are decoded correctly, but the Kth is not, then the decoding is stopped and the achieved data rate in that particular transmission is:

$$R' = \sum_{s=1}^{K-1} R_s. \tag{5.53}$$

Superposition coding is creating *self-interference* among multiple packets, such that different packets are disturbing each other's reception. Although it seems that such an operation is sub-optimal, we present two problems for which the idea of superposition coding offers elegant solutions.

5.8.1 Broadcast and Non-Orthogonal Access

The term "broadcast" is used in communication engineering in different ways, not always consistent with each other[5]. For example "broadcast traffic" refers to, for example, a transmission of a TV tower where the same data is intended to be received by all terminals that are in range of the transmitter. In information theory a "broadcast channel" is a

5 This is also addressed in Chapter 1 when multicast is introduced.

communication channel in which Basil's transmission is received by two or more receivers (Zoya, Xia, etc.), but the transmitted signal may contain separate data for each of the receivers, unlike the usual use of a "TV broadcast".

An example of a broadcast channel is a base station (Basil) whose signal is received by more than one terminal in its range. A single transmission made by Basil, consisting of L baseband symbols, may simultaneously carry *common data*, which corresponds to the "broadcast traffic" as in TV broadcast, as well as *individual data* for Zoya and/or Xia. Common data is intended to be received by both Zoya and Xia, and in that sense it represents broadcast traffic. As already indicated in Chapter 1, we will also use the term *multicast traffic* to explicitly denote the case in which there are more than one intended recipients of the message. The individual data is part of the *unicast traffic*. The individual data for Zoya *should* be decoded by Zoya, while it *may* be decoded by Xia, provided that it helps Xia to decode her individual data or the common data. Clearly, the identical statement is valid when the roles of Zoya and Xia are reversed.

Let us consider the case of a broadcast channel in which there is no common data (multicast traffic), but only individual data intended for Zoya and Xia. A straightforward way to do it would be to use time sharing, as in the downlink transmissions described in Chapter 1, such that in one time slot only one user is served. We can present such time sharing, although in a convoluted way, through superposition coding: in the slot in which Zoya is served, there is superposition coding that allocated all the power to the packet of Zoya and zero power to the packet of Xia. In the next slot the situation is the opposite one. This representation implies that the time-shared downlink transmission is a degenerate form of superposition coding, where the superposition coefficients can be varied from one transmission to another.

The previous interpretation motivates us to try to improve the throughput by exploring the design space of superposition coding. In each transmission there are $S = 2$ data packets, for Zoya and Xia, with power fractions α_Z and α_X, respectively. In contrast to the degenerate case, we can set both α_Z and α_X to have non-zero values. Furthermore, note that the values of α_Z and α_X can be changed in each slot and this should be either known by the receivers in advance or communicated through the transmission. For example, the transmitted superposed packet can be preceded by a common preamble, modulated without superposition coding, which carries information about the coefficients α_Z and α_X and is a common data, intended to be decoded by both Zoya and Xia. The transmission methods in which both α_Z and α_X are not zero represent a class of methods, sometimes referred to as *non-orthogonal multiple access (NOMA).*

Now let us assume that Zoya receives a signal that is stronger compared to the signal received by Xia. Furthermore, let us assume that both data packets are modulated with QPSK and the coefficients are chosen, for example, $\alpha_Z = 0.1$ and $\alpha_X = 0.9$. The overall baseband signal transmitted by Basil, denoted by b, is given by:

$$b = \sqrt{\alpha_Z}z + \sqrt{\alpha_X}x \tag{5.54}$$

where z is the part of the symbol sent by Basil to Zoya, while x is the part of the symbol sent by Basil to Xia. The nominal data rate of each signal z and x is set to $R_Z = R_X = 2$ bits/symbol. The noise power is assumed to be identical for both receivers. In this setup, the first signal to be decoded is x and, after canceling it, z is decoded.

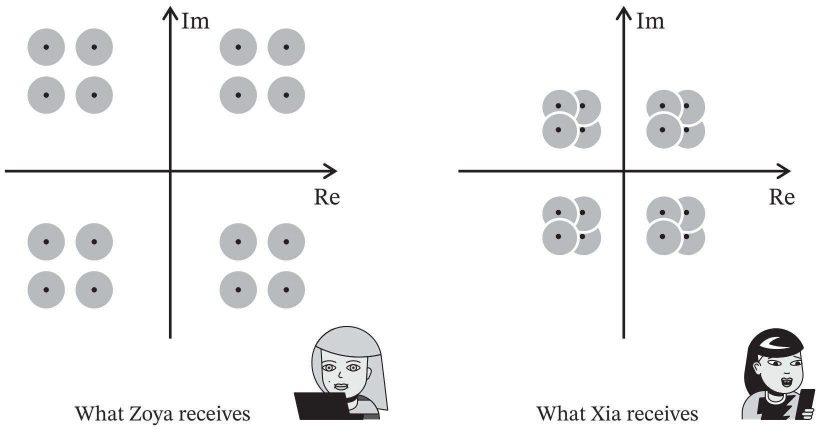

Figure 5.10 Received constellations of Zoya and Xia when Basil broadcasts two superposed QPSK signals. The signal received by Xia is weaker than the signal received by Zoya.

Figure 5.10 depicts the received signals for Zoya and Xia, respectively. By choosing $\alpha_X > \alpha_Z$, as in our example, we are ensuring the following type of operation. Both Zoya and Xia can decode x by determining in which quadrant the received signal lies; however, only Zoya can reliably decode z. If x contains the data for Xia and z contains the data for Zoya, then the goal of the broadcast is achieved, as each receiver manages to receive the desired data. In addition, Zoya is also decoding the data of Xia, since that is part of the process for getting the desired data.

The performance of non-orthogonal access, implemented as a broadcast with superposition coding, can be tuned by choosing several parameters: $\alpha_Z, \alpha_X, R_Z, R_X$. On the other hand, the definition of a suitable performance objective can be more involved. If we want to optimize the overall throughput, then the solution is to set $R_X = 0$ and $\alpha_X = 0$, such that the weaker receiver is ignored. Another performance objective could be to maximize the overall throughput by guaranteeing minimal throughput for both Zoya and Xia; in this case we must allocate some power to Xia by setting $\alpha_C > 0$ as her data rate cannot be $R_X = 0$. In any case, the performance that can be obtained through the optimization process will be always at least as good as the reference scheme that uses time sharing and serves one user at a time.

The decoding of someone else's data may raise the issue of data security. Indeed, Zoya is able to decode the data of Xia, but she may not be able to *interpret* the data, since the data bits obtained by decoding the baseband signals can be encrypted with a secret key known only by the transmitter Basil and the receiver Xia. Yet, once Zoya has the data for Xia, she may try to break the cipher and decrypt Xia's data. The situation is completely opposite with the data of Zoya: to start with, Xia cannot decode it, such that one can argue that Zoya's data is protected even more than what is offered by the encryption key.

5.8.2 Unequal Error Protection (UEP)

When the transmitted data represents multimedia content, such as an image or video, then not all the parts of the data have the same importance. As an example, consider a video encoded in the following way: basic data D_B that enables the video to be reconstructed at a lower quality and enhancement data D_E that improves the resolution. It is essential to receive the basic data D_B, which makes it a must-have. The reception of the enhancement data improves the quality, such that it is nice-to-have. This is different from the common way of treating all bits equally at the reception and can be reflected in a suitable utility function $u(D_B, D_E)$, which puts higher value on the bits belonging to the basic data compared to the enhancement data. As a simple example, the utility function u gets the following values for different outcomes:

- $u = 0$ if no basic data D_B is received, regardless of whether D_E is received
- $u = 1$ if D_B is received, but D_E not
- $u = 2$ if both D_B and D_E are received.

Let q_B be the probability that D_B is received, but not D_E; while let q_{BE} be the probability that both D_B and D_E are received. Then the expected utility at the receiver is:

$$E[u] = q_B \cdot 1 + q_{BE} \cdot 2. \tag{5.55}$$

Zoya sends a video to Yoshi over a wireless link and the objective is to maximize the expected utility. Zoya uses superposition coding, such that the basic data is modulated on z_B with α_B, and the enhancement data z_E is superposed to it with α_E. By choosing α_B and α_E, Zoya can tune the probabilities q_B and q_{BE} in order to maximize the expected utility, which supposedly reflect the overall video experience for Yoshi. Increasing α_B increases q_B, but at the same time α decreases, thus decreasing q_{BE}. Note that decoding with SIC is particularly suitable in this case, as Yoshi never attempts to decode D_E without having decoded D_B.

The idea of tuning the superposition coding to optimize the expected utility is particularly suitable for multicasting multimedia traffic to multiple receivers. This can be related to our broadcast discussion from the previous section as follows. The common message that is multicast contains the low-quality video that should be received by the worst receiver (Xia). The message unicast only to the stronger receiver (Zoya) contains the quality enhancement data. This represents an example of cross-layer optimization, as it both involves determination of the type of data (common or individual) jointly with the optimization of its transmission. The example can be generalized by applying superposition of multiple packets, which would correspond to multiple levels of enhancement. Instead of adjusting the transmission to the worst receiver, superposition coding enables graceful degradation of the multimedia quality across the receivers, such that better receivers get better quality, but worse receivers get at least some low-quality content.

5.9 Communication with Unknown Channel Coefficients

The central assumption in using the AWGN channel model $y = hx + n$ between the transmitter Xia and the receiver Yoshi is that Yoshi knows the channel coefficient h. This means that the channel h should be constant for a reasonably long time in order to allow both Yoshi and Xia to learn the value of h and then use this knowledge during the actual data communication.

The time during which the channel can be treated as constant is referred to as *coherence time* and let us assume that this corresponds to the duration of L_C symbols. Let L_P be the required number of symbols to be used as pilots, such that Yoshi can get a satisfactory estimate of h. Then the AWGN communication model is feasible to be used if $L_P \ll L_C$ and thus the channel estimation overhead is negligible. However, in some cases this assumption is not applicable. For example, this is the case when Yoshi moves relatively quickly with respect to Xia, such that the channel changes rapidly and L_C is low. In that case the value of L_P may be close to L_C, leading an inefficient operation: most of the resources (symbols) are spent on learning h, which is only an auxiliary step and not a goal per se, while few resources are left for the real goal, which is the communication of data from Xia to Yoshi.

In order to cope with this situation, Xia and Yoshi should resort to *non-coherent* communication, where the communication takes place under the assumption that h is unknown to Yoshi. Working with unknown h means that the receiver Yoshi treats h as a part of the generalized noise, which is a term that denotes the set of all factors that introduce random disturbance to Yoshi's reception. Note that h plays the role of a multiplicative noise instead of an additive one.

Let us start from the extreme case $L_C = 1$, where h changes independently from one symbol to another. The ith received symbol is given by

$$y_i = h_i x_i + n_i \tag{5.56}$$

but here both h_i and n_i are random unknown disturbances. The simplest communication scheme that Xia can use is *ON-OFF* keying, also known as binary *amplitude shift leying (ASK)*: Xia sends the bit value 0 by setting $x_i = 0$ and 1 with $x_i = \sqrt{P}$, where P is the maximal allowed power. Yoshi calculates the amplitude of the received signal:

$$r = |y_i| = |h_i x_i + n_i|. \tag{5.57}$$

The received signal needs to be tested with respect to a fixed threshold ρ, chosen in a way to minimize the probability of error. Recall that in the BPSK over AWGN channel, error occurs only if the noise n_i has an excessive value, regardless of whether the bit value 0 or 1 is sent. In the case of ASK, when 0 is sent, again the additive noise is the only factor that contributes to error, as $r = |n_i|$ and an excessive value of the noise, either positive or negative, can bring the value of r over the threshold ρ, leading to an erroneous decision by Yoshi. However, when the point 1 of the ASK constellation is sent, then $r = |h_i\sqrt{P} + n_i|$. In this case, even if there is no noise $n_i = 0$, an error can occur if the value of $|h_i|$ is very low, resulting in $r < \rho$. We note that binary ASK is the basic option; there can be Mary ASK schemes that use M different power levels, thus transmitting $\log_2 M$ bits per symbol.

If the coherence time $L_C > 1$, then other non-coherent communication schemes are possible, taking advantage of the fact that h is unknown, but constant. One idea is to use *differential modulation*, which works as follows. Let us assume that Xia sends the same symbols as in BPSK, that is $x = 1$ or $x = -1$, where for simplicity we have assumed that the average (and in this case the maximal as well) transmit power of Xia is also one. Let the first symbol x_1 be arbitrary and it does not contain any information for Yoshi; but let us fix $x_1 = 1$ and assume that this is known both to Xia and Yoshi. Then Xia modulates the first bit when sending x_2. The information is not in x_2 itself, but in the *difference* between x_2 and x_1: Xia sends the bit value $b_1 = 0$ by setting $x_2 = x_1$ and $b_1 = 1$ by setting $x_2 = -x_1$. Yoshi creates the decision variable:

$$r = |y_2 - y_1| = |hx_2 + n_2 - hx_1 - n_1| = |h(x_2 - x_1) + n_2 - n_1|. \tag{5.58}$$

When $b_1 = 0$ then $|x_2 - x_1| = 0$, while when $b_1 = 1$, then $|x_2 - x_1| = 1$ and, similar to the case with ASK, Yoshi should compare the decision variable with a threshold ρ. The second bit b_2 is modulated in identical way, by choosing x_3 relative to x_2. This type of modulation is called *differential BPSK (DBPSK)*. Continuing in the same fashion, Xia can send L differentially modulated bit values by sending $L + 1$ symbols.

Differential modulation requires h to be constant throughout the whole packet transmission, since it should be $h_2 = h_1 = h$, then $h_3 = h_2 = h_1 = h$, etc. In order to break this chain of equality with the previous symbol, let us consider the case with coherence time $L_C = 2$. We show how to construct a scheme that takes advantage of the fact that h is constant over two symbols only. Let us use the term *vector symbol* to denote a vector of two symbols $\mathbf{x} = (x_1, x_2)$. Using Figure 5.1 from the beginning of this chapter and the layering terminology, the use of vector symbol can be seen as creating an extra layer that resides above the layer TXbaseband–RXbaseband and DataSender–DataCollector.

In the simplest case, let the vector $\mathbf{x}$ sent by Xia take only two possible values, $\mathbf{x} = (1, 0) = \mathbf{s}_1$ and $\mathbf{x} = (0, 1) = \mathbf{s}_2$. Furthermore, in order to facilitate two-dimensional illustrations, let us simplify and assume for the following discussion that the values of x, h and n are real. We can now represent the two possible symbols in a two-dimensional coordinate system, as in Figure 5.11, where the horizontal axis represents the value of hx_1 and the vertical axis the value of hx_2. The gray circle represents the cloud of the additive noise n; note that here the cloud is two-dimensional, not because n is complex, but because the figure represents the two real noise samples of two consecutive symbols. Figure 5.11(a) illustrates the case $h = 1$ and the dashed line is used to mark the decision region, which is half-plane. Clearly, if $h = 1$ and Yoshi knows it, then the choice of the two constellation points in Figure 5.11(a) is not optimal, as they could be separated by a larger distance. This is not valid in the non-coherent regime, as Yoshi does not know h and the value of h can change to a negative one, as in Figure 5.11(b). However, regardless of the actual (real) value of h, what remains invariant is that $h\mathbf{s}_1$ lies always on the horizontal axis, while $h\mathbf{s}_2$ is always on the vertical axis. In other words, the one-dimensional linear subspace in which $h\mathbf{s}_i$ resides remains invariant. This means that the decision criterion, determined by the dashed line, stays identical in both Figure 5.11(a) and Figure 5.11(b). Once the

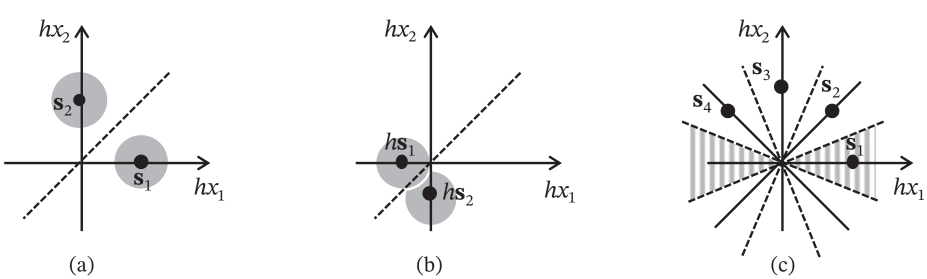

Figure 5.11 Illustration of non-coherent communication with linear sub-spaces. The values of the symbols, channel coefficients and noise are assumed to be real. (a) Case $h = 1$ and two constellation points. (b) When $h \neq 1$ each of the constellation points remains on the same axis, unless disturbed by noise. (c) Quaternary non-coherent modulation; $h = 1$ is assumed.

decision criterion is determined, then the probability of error can be determined as the joint probability that h and n lead to the event in which the noise cloud crosses the decision region.

In the previous scheme Xia sends one bit by using two symbols, such that the nominal data rate is halved compared to, say, BPSK in which one bit is modulated in each symbol. This can be compensated by using a quaternary non-coherent modulation, as illustrated in Figure 5.11(c), where it is assumed $h = 1$, since two bits are sent over two symbols. The decision region for each possible symbol spans 90°, as is illustrated for the decision region related to the symbol $\mathbf{s}_1$. The fact that the linear subspace does not change can be used to design modulation schemes in multi-dimensional spaces, with complex h, x, z and using coherence time $L_C > 2$.

We end this section by discussing briefly the possibilities for designing hybrid schemes. When the coherence time is very long, $L_C \gg 1$, then it is acceptable to waste some symbols as pilots that carry only dummy data in order to apply coherent communication afterwards. On the other hand, if L_C is very close to 1, then it is not feasible to explicitly invest resources in pilots. However, the following opportunity should be noted: if Xia sends several symbols to Yoshi to be received non-coherently and Yoshi buffers those symbols, then after he decodes the data correctly, he can reuse the known data in order to estimate the channel. Hence, when L_C is in the intermediate region, not too large and not too close to one, then one can think of a communication protocol in which the part of the packet that should contain pilots also contains data that is detected non-coherently. The non-coherent data can have its own integrity (CRC) check, such that Yoshi can verify its correctness and use it to estimate the channel h. The non-coherent part of the packet uses $L_P < L_C$ symbols, where L_P is known in advance by Xia and Yoshi. After that, for the remaining $L_C - L_P$ symbols Xia uses coherent modulation. This is a simple example of a scheme with *blind channel estimation*, where the estimation is not done by dummy data, previously agreed between Xia and Yoshi.

5.10 Chapter Summary

This chapter has established the connection between, on the one hand, *networking*, where the basic building blocks are packets, and, on the other hand, the *digital communication theory*, where the main themes are bits, symbols, modulation, and noise. We have worked with the baseband model of the system, where the basic units are complex symbols sent by the transmitter and their noisy versions at the receiver. The baseband model qualitatively expands the models from the previous sections in multiple ways: *(i)* it captures the central role that the SNR has in digital communications; *(ii)* it allows the error performance to be related to the received power and thus introduce effects that a physical distance has on the signal reception; *(iii)* it reveals new possibilities for design of protocols and transmission schemes based on control of the transmit power or creating multiple streams through superposition coding. Finally, the chapter has introduced the important idea of non-coherent communication that occurs when the receiver cannot learn the channel before decoding the useful data.

5.11 Further Reading

The material discussed in this chapter can be further explored in Gallager [2008], which discusses the principles of digital communication as well as Proakis and Salehi [2008], which provides a comprehensive discussion on signals and modulation. The ideas of superposition coding for non-orthogonal access in a broadcast channel have been introduced in Cover [1972], while it has been popularized under the term NOMA in Saito et al. [2013]. Unequal error protection in broadcast channels has become a part of the Digital Video Broadcasting (DVB) standard, see [2004–11].

5.12 Problems and Reflections

1. *Collision outcomes with actual modulation*. The introduction of a symbol level model allows us to assume that the packets contain symbols with a particular modulation type and then derive the probabilities of successful decoding and capture. Assume that Zoya and Xia are transmitting simultaneously in the uplink to Basil. The signal received by Basil is:

$$y_\mathrm{B} = h_\mathrm{ZB} z + h_\mathrm{XB} x + n \tag{5.59}$$

   where n is the noise with variance P_N. Both Zoya and Xia transmit with a fixed power P_T and send packets consisting of L symbols. The communication channel is slotted and one slot contains L symbols. A packet transmission can start only at the start of a slot, such that the transmissions of Zoya and Xia are packet synchronous. Each symbol contains QPSK modulated data and it is assumed that a packet is decoded successfully if all its L symbols are decoded successfully.
   The SNRs of Zoya and Xia are denoted by γ_ZB and γ_XB, respectively, and are given by:

$$\gamma_\mathrm{ZB} = \frac{|h_\mathrm{ZB}|^2 P_\mathrm{T}}{P_\mathrm{N}} \qquad \gamma_\mathrm{XB} = \frac{|h_\mathrm{XB}|^2 P_\mathrm{T}}{P_\mathrm{N}}. \tag{5.60}$$

   There are three possible outcomes of this collision between Zoya and Xia, as Basil will be able to decode 0 packets correctly, 1 packet correctly (through capture), and 2 packets correctly (capture and successive interference cancellation). Investigate how these probabilities are affected by changing γ_ZB, γ_XB, and L.
   Hint: Fix at first L. Then find a value of γ_ZB that offers a high probability of successful decoding of Zoya's packet for that L under the assumption that the interference from Xia is absent. Then consider the interference from Xia and vary γ_XB. In deriving the probabilities make the simplifying assumption that any interference can be treated as a Gaussian noise.
2. *Flexible transmit power*. Assume that the same model as in the previous assignment is used, but now Zoya and Xia are allowed to change the transmit power at each transmission. The transmit power can be at most P_T. Investigate the strategies that Zoya and Xia can use, while assuming that there is no feedback from Basil.

3. *Flexible transmit power and feedback*. Repeat the previous assignment, but now assume that there is a feedback from Basil after each slot. The feedback tells us whether Basil decoded no packet, one packet (also telling us whether this decoded packet was from Zoya or Xia), or both packets.
4. *Adaptive modulation over a broadcast channel*. Basil transmits downlink packets to Zoya and Yoshi. Assume that the available modulation constellations are BPSK, QPSK, and 16-QAM. A packet consists of 8 control bits and D bits of data, where D can be 32, 64 or 128 bits. The control bits contain, besides the other information, an information about the:
   - Packet destination, which can be Zoya, Yoshi, or both of them (broadcast packet)
   - The actual amount of D data bits following the control bits
   - The type of modulation (BPSK, QPSK, and 16-QAM) used to send the data bits.

   As in the previous assignments, the SNRs of Zoya and Yoshi are denoted by γ_{ZB} and γ_{XB}, but now they refer to the downlink transmission.
   (a) Which modulation should be chosen for the control bits? Discuss the pros and cons.
   (b) Following the choice in (a), find the SNR thresholds for using adaptive modulation for this downlink transmission. Keep in mind that some of the packets are broadcast, intended for both Zoya and Yoshi, such that for those packets the adaptive modulation should take into account both γ_{ZB} and γ_{XB}.
5. *Scheduling of NOMA downlink*. The objective of this assignment is to investigate the scheduling strategies for downlink transmission by using NOMA, discussed in Section 5.8.1. Consider a setup in which Basil needs to transmit downlink data to K users. Some of this data is intended for all K users, while some of the data is intended for each of the individual K users. Hence, there are $K + 1$ data packets that Basil needs to send. Assume that Basil knows the SNR of each user. You are free to make assumptions about the packet sizes as well as the available modulation constellations.

   Discuss the trade-offs that need to be considered when Basil decides how to schedule the users. Should he, for example, pair two users with strong SNRs and transmit to them using superposition coding? How should the broadcast packet be sent? What about the fairness among the users?