9. Physical Layer Modeling

A. de Baynast (European Microsoft Innovation Center)
M. Bohge, D. Willkomm (Technische Universität Berlin)
J. Gross (RWTH Aachen University)

The Physical Layer (PHY) is serving as the interface between the Data Link Layer (DLL) and the environment. Accordingly, it defines the relation between the device and the physical medium. In wireless systems, the general task of the PHY is to convert bit streams into radio waves and vice versa. Though the transmitter and the receiver are dual, they are comprised of different components in the physical layer. The transmitter takes digital input in form of (payload) bits and converts them into an analog signal, generally around a given carrier frequency which is then radiated via the antenna. At the receiver, this analog signal which has been distorted during the propagation is then converted back into a (payload) bit stream. The general goal of the PHY is to ensure that the bit stream at the transmitter and at the receiver are identical. This is a very challenging task as the wireless channel (see Section 11) can distort and corrupt the analog signal in many different, random ways. In the following, we first provide an overview of the different functionalities of the PHY at the transmitter and receiver as applied in most common standards today like for cellular networks (such as Global System for Mobile Communications (GSM), Universal Mobile Telecommunications System (UMTS), Long Term Evolution (LTE)) or local/metropolitan area networks (e.g. IEEE 802.11, IEEE 802.16) as well as for broadcast networks (Digital Audio Broadcasting (DAB), Digital Video Broadcasting (DVB)). Then we discuss common simulation approaches used in the PHY and their shortcomings for network simulation. Finally, we comment on ways to include selected aspects of the PHY in network simulation models.

9.1 Overview of the PHY Layer

In wireless systems the PHY layer can be subdivided into four domains as depicted in Figure 9.1: the **bit domain**, the **symbol domain**, the **sample domain** and the **waveform/analog domain**. Note that any layer above the PHY can be considered to be part of a fifth domain referred to as the "packet domain", as indicated for the DLL in Figure 9.1. Data through the PHY layer is sequentially represented in each of these domains, i.e., by packets, bits, symbols, samples and finally by waveforms.

There are historical reasons behind this decomposition. Until the emergence of powerful processors in embedded communication devices during the last two decades, the bit stream was directly transformed into waveforms, representing the information either by amplitude levels referred to as Amplitude Modulation (AM) [375, Chap. 4, p.169], or by frequencies referred to as Frequency Modulation (FM) [395]. The corresponding waveform was directly modulated by an oscillator set to a specific carrier frequency. At the receiver, the analog signal was filtered by a matched filter in order to reduce the noise power outside the bandwidth of interest. In the particular case of FM receiver [395], a simple Phase-Locked Loop (PLL) could be used to lock to the current frequency of the signal. Whereas the hardware implementation of such a receiver was extremely cheap, the spectral efficiency (i.e. the ratio of throughput over the required system bandwidth) remained low.

As the demand for higher data rates increased dramatically over the last two decades, designs of transmission systems with higher spectral efficiency over broadband channels were required. Higher spectral efficiency has been achieved by using more sophisticated transmission schemes, which in turn require more complex algorithms especially at the receiver. In order to support complex algorithms, the devices nowadays comprise of one or several digital signal processors[442] that support a significant number of the operations represented in Figure 9.1. Besides the historical aspect, this decomposition into different domains is also fundamental in the comprehension of the results and the limitations of any wireless network simulator since some of these domains correspond to the different abstraction levels used by simulation tools as shown later. In this subsection, we first give a functional overview of the four different domains of the PHY before we discuss single, functional elements in the next subsection as second step.

Bit Domain. At the transmitter, a packet coming from the DLL enters the *bit domain* of the PHY. Three main functions are performed here: Cyclic Redundancy Check (CRC) coding, Forward Error Correction (FEC) coding and interleaving. Firstly, CRC bits are added to the packet bits. The main purpose of CRC code is to detect at the receiver if an error occurred during a transmission. Note that a CRC code only allows usually the detection of errors, it does not allow the correction. In order to correct eventual transmission errors, FEC coding schemes are used. Many different codes with different characteristics are known today: for correcting only few errors (less than 1 erroneous bit for 1000 bits transmitted) without penalizing the transmission rate too much, Reed-Solomon (RS) codes are good candidates. If the transmission channel introduces more errors (like typical GSM channels do), convolutional codes can be used. Even more efficient error correction codes are Turbo-codes [58] or Low-Density-Parity-Check (LDPC) codes [170, 300]. While they are more efficient than convolutional codes, their decoding algorithms are more complex as they require several decoding iterations. However, most of the current communication standards support them at least as option

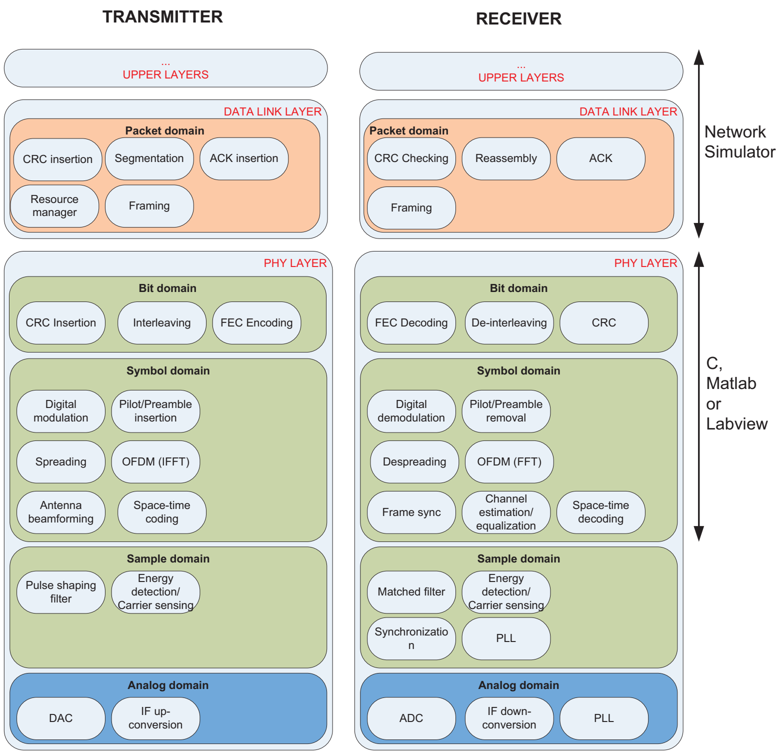

Fig. 9.1: Decomposition of the MAC and PHY layer into 5 domains for wireless transmission systems: the packet domain (MAC), the bit domain, the symbol domain, the sample domain and the analog domain. A tentative list (non exhaustive) of operations that enable reliable transmission between the transmitter and the receiver in current transmission systems is also represented in the figure. The operations are listed by domain which corresponds to the abstraction level in network simulators.

(like in UMTS, LTE, DVB or IEEE 802.11n). As the transmission errors often happen in bursts [38], the coded bit stream is interleaved before transmission so that the corresponding de-interleaver at the receiver will spread the bit errors uniformly within the received coded bit stream. All these operations are realized on the bit stream (see the bit domain in Figure 9.1). However, this binary description of the transmit information is not sufficient for direct mapping to an analog waveform in systems with higher spectral efficiency.

Symbol Domain. An intermediate step consists of transforming a bit or several bits into a symbol. Hence, the transmit information enters now the *symbol domain*. For instance, in an Amplitude Shift Keying (ASK) radio system, the symbol value represents the amplitude of the waveform. In Frequency Shift Keying (FSK) systems, the symbol value corresponds to a specific frequency taken from a pre-defined set. Hence, bits are still represented by digital values but these are already "place-holders" for specific characteristics of waveforms. In most today's systems however, more advanced schemes are used. Typical bit-to-symbol mapping schemes are Gaussian Minimum Shift Keying (GMSK) as used in GSM, Quadrature Amplitude Modulation (QAM) in IEEE 802.11, UMTS, LTE, IEEE 802.16, or Differential Quadrature Phase Shift Keying (DQPSK) in DAB/DVB. A description and a complete analysis of all these modulation schemes can be found in [375]. In order to simplify the design of the receiver, some pilot symbols and preamble or end-preamble are inserted. For instance, several null symbols are inserted before each packet in IEEE 802.11 such that the beginning of a packet can be detected using a double sliding window algorithm. Recently, advanced transmission schemes for broadband wireless communication have been proposed and some of them have been deployed successfully. These advanced transmission schemes are spread-spectrum, multi-carrier modulation and Multiple Input Multiple Output (MIMO) systems. In the particular case of spread-spectrum (which is used either for multiple access – see Section 10.1.4 – or for combating frequency selectivity of the transmission channels) each symbol is spread by a so-called spreading sequence of length equal to the spreading factor. In UMTS systems, the spreading factor varies from 16 to 1024 such that multiple users can simultaneously transmit in the same band without interfering with each other since the spreading sequences are mutually orthogonal. Hence, the bit-to-symbol mapping leads to a few bits mapped to many modulation symbols (i.e. "spreading" the information over many symbols). This leads to a better performance in channels with inter-symbol interference. A further advanced transmission scheme are multi-carrier modulation systems; the most successful example of such schemes is Orthogonal Frequency Division Multiplex (OFDM). An OFDM system combats efficiently the inter-symbol interference occurring in frequency selective propagation channels [375, Chap.12, p.718] by applying the Digital Fourier Transform (DFT) at the receiver and its inverse at the transmitter. This technique considerably simplifies the receiver implementation in case of frequency-selective channels since it is much easier to invert the channel effect in the frequency domain, i.e. after taking the DFT operation as long as it is done at the correct time lag (and assuming that the inverse of the coherence bandwidth of the channel does not exceed the guard interval of the system). Finally, a third advanced transmission scheme consists of utilizing antenna arrays at the transmitter or at the receiver or at both sides. The main advantage of MIMO systems is to considerably increase the spectral efficiency of the system by exploiting the spatial diversity, as demonstrated in [158]. Three techniques for MIMO systems are generally used. Firstly, beamforming consists of steering the energy of the signal towards the receiving antenna(s) by adjusting the weights of the phase array of the antennas [477]. The second technique is called space-time coding [457] which in contrary consists of spreading the energy spatially into all directions in an uniform way such that all potential receivers can receive the signal. Finally, the third technique consists of spatially multiplexing the emitted signal over all transmit antennas. Whereas considerable gains in terms of throughput and reliability are achieved, the drawback of these techniques is their computational complexity since matrix inversion of the channel coefficients is often required at the receiver side.

Sample and Waveform Domain. The description of the transmit information in terms of symbols is still not enough since it corresponds to a stream of consecutive discrete values. However, there are several different ways to map the symbols into waveforms, especially regarding the transition from one symbol to the next one. Most importantly, these different ways end up in different bandwidths that the signal consumes in the frequency domain. The exact transition from one symbol to the next one is governed by the so-called pulse shaping filters which limit the signal to the required bandwidth. Most of the current systems are using a square-root raised cosine filter at the transmitter and receiver. This filter reduces the required bandwidth to a minimum while no symbol interference occurs (theory of the eye pattern [375, Chap. 9]). As digital signal processors have become more and more powerful, the pulse shaping filtering is nowadays realized digitally. We refer to this level as *sample domain* since the output of the filter are sampled at much higher rate than the incoming symbols. The samples are then converted into an analog signal by a Digital-to-Analog Converter (DAC) which is finally modulated to the respective carrier frequency (for instance 2.4 GHz for 802.11, 1.6 GHz for LTE, 900 MHz or 1800 MHz for GSM, 400-700 MHz for DVB).

It has to be mentioned that the whole process is performed by quite specialized hardware. In order to support a transmission rate of 10 or 20 megabit per second, the digital sequence representing the same information at the sample domain can easily consume a bit rate of several hundred megabit per second. These information flows are processed by digital signal processing units which realize all functions on dedicated hardware as ASIC or DSPs. This obviously also applies to the analog part (D/A conversion, mixing, amplification) which is performed by a radio.

Compared to the transmitter, the receiver is far more complex in its functionality. This is mainly due to the processing steps of the sample domain since the receiver needs to be synchronized in time and frequency (and possibly space for systems with multiple antennas) in order to accurately retrieve the symbols. In fact, a wireless receiver is usually able to detect and decode transmit signals which are only a few decibel above the noise power of the system (every electromagnetic system features some form of noise which interferes with very weak signals arriving from some transmitter – see Section 11.10). This requires quite sophisticated and specialized processing operations to be performed which relies on digital signal processors. After synchronization the received signal basically works its way through the transmitter components in an opposite way.

In the next section, we detail each operation at the transmitter and the receiver into independent paragraphs with special emphasize on the computational complexity. When the description of the operation at the receiver is dual of the operation at the transmitter, we describe the operation of the transmitter and its counterpart function at the receiver within the same paragraph. We would like to insist that the list is not exhaustive. It provides an overview of the main functions implemented in current wireless transmission standards. The purpose of the list is to give an overview of the operations commonly utilized in current wireless transmission standards.

9.2 Description of the Main Components of the PHY Layer

In this section, a brief description of the aforementioned components at the transmitter and at the receiver is given. A complete list and explanation of all techniques used in the PHY layer is beyond the scope of this book. Furthermore, whereas the transmitter steps are clearly described in the standards, the implementation of the receiver is left free for manufacturers and therefore the exact operations at the receiver are generally not documented in details. The following descriptions are intended only to give an overview of the functional blocks present at a transmitter and receiver to the reader. For more details, the reader is asked to refer to the citations and references therein.

9.2.1 Components of the Bit Domain

Cyclic Redundancy Check Codes. CRC codes are hash functions designed to detect transmission errors. A CRC-enabled device calculates a short, fixed-length binary sequence, known as the CRC code, for each block of data (i.e. usually a frame, a header or a packet) and sends them both together. Generally, the CRC bits are padded at the end of the block. When a block is read or received, the device repeats the calculation. If the new CRC does not match the one calculated earlier, then the block contains a data error and the device may take some action such as discarding the block (DAB, DVB, GSM, UMTS) or requesting the block to be sent again (WLAN). The term CRC code originates from the fact that the check code is redundant (it adds zero information) and the algorithm is based on cyclic codes. CRC codes are popular because they are simple to implement, and are particularly effective at detecting common errors caused by noise in transmission channels.

All popular wireless systems are using CRC codes at the PHY (mostly for header protection) and at the MAC layer (header protection as well as payload protection). CRC codes are quite efficient at detecting the accidental alteration of data. Typically, an n-bit CRC, applied to a data block of arbitrary length, will detect any single error burst not longer than n bits and will detect a fraction $1 - 2^{-n}$ of all longer error bursts. As errors in wireless channels tend to be distributed non-randomly, i.e. they are "bursty", CRC codes' properties are more useful than any alternative schemes.

Forward Error Correction Codes. Encoding and decoding information via FEC codes is a system of error control for data transmission, whereby the sender adds redundancy to the transmitted information using a predetermined algorithm, also known as an error-correction code. Each redundant bit is invariably a complex function of many original information bits. The original information may or may not appear in the encoded output; codes that include the unmodified input in the output are called systematic, while those that do not are non-systematic. Contrary to CRC, FEC coding schemes allow the receiver to correct errors (within some upper bound). The advantages of FEC codes are that a feedback-channel is not required (as in GSM, UMTS systems and broadcasting systems) or that retransmissions of data are dramatically reduced in presence of a feedback-channel (as in IEEE 802.11 or LTE). This advantage comes at the cost of a lower throughput as redundancy is added to the bit stream. Given a FEC code of rate $1/3$ generates 2 bits of redundancy per information bit, which triples the bandwidth needed for the transmission or equivalently reduces the effective throughput of the system by a factor of three. The maximum fraction of errors that can be corrected is determined in advance by the design of the code, so different FEC codes are suitable for different transmission conditions.

There are two main categories of FEC codes: Convolutional codes and block codes [375, Chapter 8]. Convolutional codes work on bit or symbol streams of arbitrary length. They are most often decoded with the Viterbi algorithm [166]. Viterbi decoding allows asymptotically optimal decoding efficiency with increasing constraint length of the convolutional code, but at the expense of exponentially increasing complexity with respect to the constrained length. Most of the current wireless transmission standards (GSM, UMTS, LTE, IEEE 802.11, DAB, DVB) are using convolutional codes. The corresponding coding rate and constrained lengths are shown in Table 9.1.

	GSM	UMTS	LTE	IEEE 802.11	DVB-T
Coding rate	1/2	1/2-1/3	1/3-7/8	1/3-5/6	1/2-3/4
Constrained length	7	7 (9)	7-9	7	8

Table 9.1: Coding rate of convolutional codes as used in the wireless transmission standards GSM, UMTS, LTE, IEEE 802.11, and DVB-T.

Block codes work on fixed-size blocks of bits or symbols of predetermined size. Practical block codes can generally be decoded fast due to their block length. There are many types of block codes, but among the classical ones the most notable is Reed-Solomon (RS) coding. In current systems, block codes and convolutional codes are frequently combined in concatenated coding schemes. A short constraint-length convolutional code with low coding rate does most of the work and a Reed-Solomon code with larger symbol size and block length corrects the few errors left by convolutional decoder. Due to their order in the transmission chain, the Reed-Solomon and the convolutional codes are usually referred as outer and inner codes, respectively.

Whereas the convolutional codes provide a good trade off between the computational complexity and error-correction capability, there are other FEC coding schemes whose performance is within few tenths of decibels of the maximal theoretical rate. The maximal theoretical rate is referred as the *Shannon limit* in information theory [106]. The best known of these codes are the Turbo-codes [58] and the LDPC codes [170]. Although the complexity of the decoding is higher than for convolutional codes, their performance is such that they are proposed in most of the current standards as optional schemes (for example turbo-code of coding 1/3 in UMTS and LDPC in IEEE 802.11n). The Turbo-codes and LDPC codes share the same decoding algorithm referred to as belief propagation, sum-product or sigma-pi [111, 268]. This is an iterative algorithm for performing inference on graphical models, such as factor graphs that can be used to represent Turbo-codes and LDPC codes. Several iterations are required before convergence as illustrated in Figure 9.2.

FEC codes have often an all-or-nothing tendency, i.e. they can perfectly extract the transmitted message if the Signal-to-Noise-Ratio (SNR) of the transmission channel is large enough and cannot correct any error if the SNR of the transmission channel is too small. Therefore, digital communication systems that use FEC coding tend to work well above a certain minimum SNR and not at all below. Typical values are 8 decibels in GSM and 10 decibels for UMTS. This all-or-nothing tendency becomes more pronounced the more efficient the code works, i.e. the closer the FEC code approaches the theoretical limit imposed by the Shannon capacity [106]. This is particularly true for the Turbo-codes, the LDPC codes and the concatenated schemes with convolutional code as inner code and RS code as outer code.

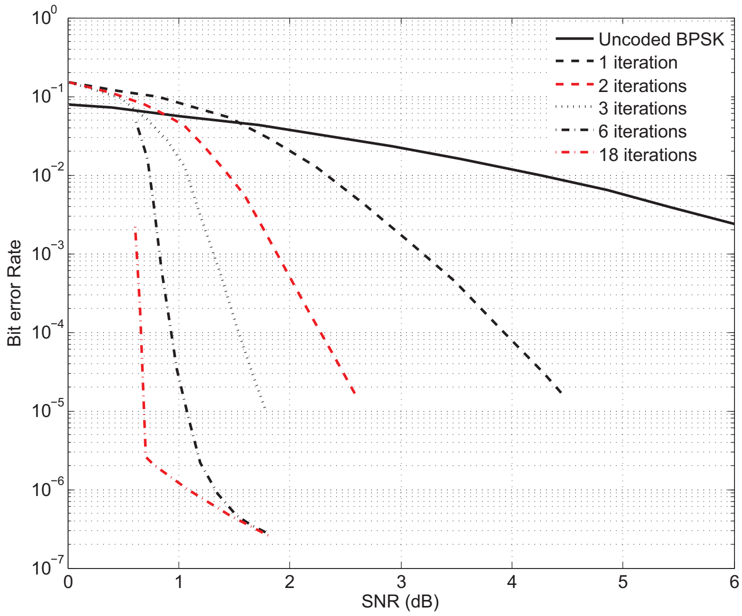

Fig. 9.2: Probability of bit error as a function of the signal-to-noise ratio of the transmission channel and the number of iterations of the decoding algorithm. For Turbo-codes, 5 iterations are usually sufficient. The computational complexity of the decoding algorithm is linear with respect to the number of iterations (data taken from [289]).

The performance gain of coded transmission compared to an uncoded transmission is illustrated in Figure 9.3 and is referred as the *coding gain*.

Finally, FEC coding schemes are often combined with puncturing (IEEE 802.11, LTE, DVB). Puncturing is a technique used to make the rate of a code slightly higher than the basic rate of the code [194]. It is reached by deletion of some bits in the encoder output. Bits are deleted according to a puncturing matrix. Punctured convolutional codes are also called "perforated".

Bit Interleaving/De-Interleaving. Data is often transmitted with FEC coding that enables the receiver to correct a certain number of errors which occur during transmission. If a burst of bit-error occurs, the number of bit-errors within the same code word may exceed the threshold under which a correction would have been possible. In order to reduce the effect of such error bursts, the bits of a number of consecutive codewords are interleaved before being transmitted [390]. This way, the wrong bits of an error burst are spread over several code words, which makes the error correction of the overall bit stream easier to be accomplished.

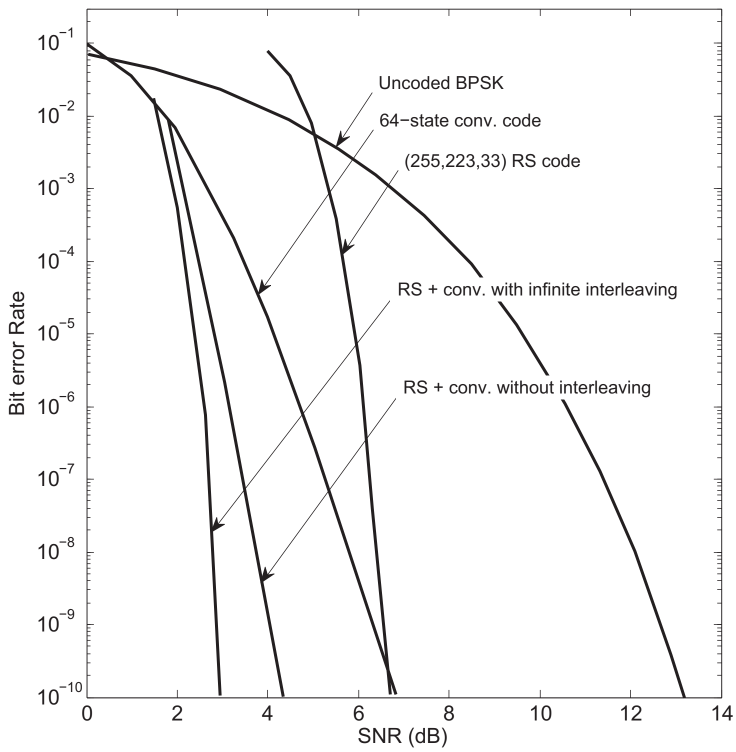

Fig. 9.3: Probability of bit error as a function of the signal-to-noise ratio of the transmission channel for several schemes: uncoded antipodal signal (uncoded BPSK), stand-alone convolutional code, stand-alone RS, concatenated scheme (convolutional code + RS) with and without interleaving. For the same spectral efficiency, the performance of concatenated schemes is superior to the performance of uncoded scheme or stand-alone FEC schemes especially for lower probability of bit error (data taken from [375]).

9.2.2 Components of the Symbol Domain

Digital Modulation/Demodulation. Digital modulation and demodulation is the process of transforming a chunk of consecutive bits into one or several parameters of a sine wave (at the transmitter) and back (at the receiver). Note that digital modulation does not generate the waveform yet but it produces a stream of waveform parameters which are turned at a later stage to an analog waveform. Modulating information onto a waveform usually involves varying one sine waveform in relation to another sine waveform. The three key parameters of a sine wave that can therefore carry information are its amplitude, its phase and its frequency. Digital modulation is sometimes referred to as constellation mapping. The most common digital modulation techniques are listed in Table. 9.2.

GSM	**UMTS**	**LTE (IEEE 802.16)**	**IEEE 802.11**
GMSK 4 bits/symb	QPSK 2 bits/symb	4/16/64-QAM 2-8 bits/symb	BPSK-4/16/64-QAM 1-8 bits/symb

Table 9.2: Types of modulation techniques used in wireless transmission standards GSM, UMTS, LTE, IEEE 802.16, and IEEE 802.11.

In GSM systems, the GMSK modulation technique is employed. GMSK is a continuous-phase frequency-shift keying modulation scheme. It is similar to standard Minimum Shift Keying (MSK) modulation [375, Chapter 5], however the digital data stream is first shaped with a Gaussian filter before being applied to a frequency modulator. This has the advantage of reducing sideband power, which in turn reduces out-of-band interference between signal carriers in adjacent frequency channels. However, the Gaussian filter increases the modulation memory in the system and causes inter-symbol interference, making it more difficult to discriminate between different transmitted data values and requiring more complex channel equalization algorithms such as an adaptive equalizer at the receiver.

In UMTS, LTE, IEEE 802.16 and IEEE 802.11 systems, QAM is utilized. It conveys two digital bit streams by modulating the amplitudes of two carrier waves, using the ASK modulation scheme. These two waves, usually sinusoids, are out of phase with each other by 90 degrees and are thus called quadrature carriers or quadrature components – hence the name of the scheme. The modulated waves are summed, and the resulting waveform is a combination of both Phase Shift Keying (PSK) and ASK. In the digital QAM case, a finite number of at least two phases, and at least two amplitudes are used (4-QAM). However, Binary Phase Shift Keying (BPSK) and Quadrature Phase Shift Keying (QPSK) modulations can be viewed as special cases of QAM modulation. QPSK is equivalent to 4-QAM and BPSK has two amplitudes but a single phase. The operation which consists of mapping bit(s) to a symbol is illustrated in Figure 9.4 in the case of QAM modulation.

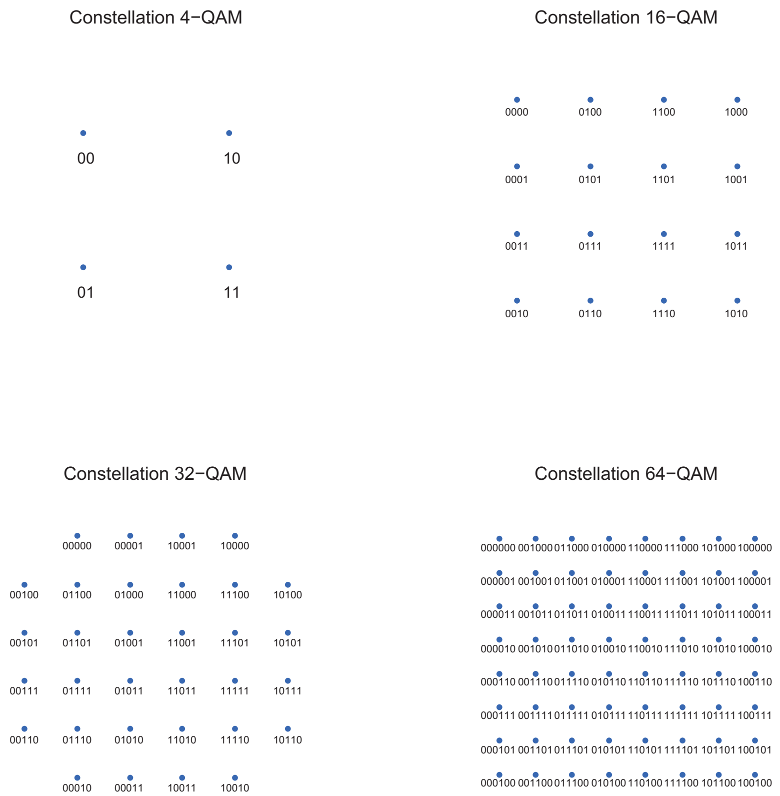

Fig. 9.4: Constellation of QAM signaling. Several bits are gathered to form one QAM symbol. The higher the modulation order the more efficient is the transmission. However, the distance between symbols becomes smaller as the number of symbols increase and is more subject to errors in presence of the transmission noise. In practice, 256 is the maximum supported order in wireless transmissions.

Pilot/Preamble Insertion and Removal. All digital wireless communication systems use pilot symbols in order to simplify the design of the receiver. Pilots are used to transmit data known in advance by the receiver. It uses them to perform synchronization and channel estimation. There are several types of pilots: preamble within the packet, few tones for OFDM systems, preamble for packet detection and null symbols for energy detection. The type of pilots used in current wireless transmission standards are listed in Table 9.3.

GSM	UMTS	LTE, IEEE 802.16
Training seq 25% of the resources	Training seq (Q) 50%	Preamble + Pilot Tones 30%

IEEE 802.11	DAB	DVB-T
Preamble + Pilot tones 30% of the resources	Preamble 5%	Preamble + Pilot Tones 10%

Table 9.3: Types of pilot techniques used in wireless transmission standards.

Spreading/De-spreading. Spreading is an advanced transmission scheme used for broadband wireless channels. It is in a sense the simplest FEC coding technique em- ployed in the symbol domain. In spread-spectrum a signal is generated which has a much larger bandwidth than would be required for conveying the pure stream of information. There are three basic techniques employed to spread the information signal: direct sequence, frequency hopping, or a hybrid of these. Spread spectrum makes use of a sequential noise-like signal structure to spread the information signal over a much wider band of frequencies. In direct sequence spread-spectrum, for example, each bit of the information stream is mapped to a sequence of bits (where typical ratios – the spreading factor – between input bit to output bits are 1/16 or 1/32 in UMTS, 1/11 in IEEE 802.11b). This sequence of bits is well-known in advance and is referred to as spreading code. Then, the resulting "spreaded" bit stream is fed to a modulator which generates now many more modulation symbols than would be required for the information stream. At the receiver the incoming signal is correlated with the spreading code to retrieve the original information signal referred to as de-spreading. Clearly, for de-spreading to work correctly, the transmit and receive spreading sequences must be the same and they must be synchronized. This requires the receiver to synchronize its sequence with the transmitter's sequence by taking the maximum of the cross-correlation function between the spreading code and received data. Good spreading codes are designed to appear as random sequences, i.e. not having long trails of 1's or 0's and overall almost the same number of 1's and 0's. Due to the pseudo-randomness of the spreading sequence, the resulting signal resembles white noise. Spreading by frequency hopping works in a similar manner. Here, a single bit is converted to one or several "hops" in frequency according to a predetermined sequence. Spread-spectrum has several advantages. First of all, it decreases the potential interference to other receivers as the transmit power of the spreaded signal is quite low. This does not harm the reception of the signal as the receiver – which knows the spreading sequence – sums up the transmit energy over 16 symbols in case of a spreading factor of 1/16 and can therefore tolerate a transmit power which is lower by a factor of 16 (equaling 12 dB). Furthermore, spread-spectrum signals mitigate inter-symbol interference caused by frequency-selective fading channels as a single bit is converted – in direct sequence spread spectrum with a factor of 1/16 – into 16 channel bits which achieves quite a coding gain. This effect even can be enhanced to resolve dominant paths of a multipath propagation environment known as RAKE receiver [446]. It is implemented in most of the UMTS receivers today. Furthermore, spread-spectrum enhances privacy as the spreading code has to be known to successfully decode the signal. Without knowing the spreading code, it is even hard to only detect the signal as the required transmit power for a large spreading factor is quite low. Due to these reasons, spread-spectrum systems were first developed by the military and are still widely applied there. Finally, spread-spectrum systems enable multiple access by assigning different data transmissions different spreading codes (in case of direct sequence spread-spectrum the corresponding multiple access schemes is referred to as Code Division Multiple Access (CDMA), see Section 10.1.4).

In current wireless standards, the two main spread-spectrum techniques are frequency hopping spread-spectrum (Bluetooth) and direct-sequence spread-spectrum (Global Positioning System (GPS), UMTS, IEEE 802.11b). In UMTS (and IS-95, CDMA2000) direct-sequence spread-spectrum is also applied for multiple-access. Its principle is illustrated in Figure 9.5.

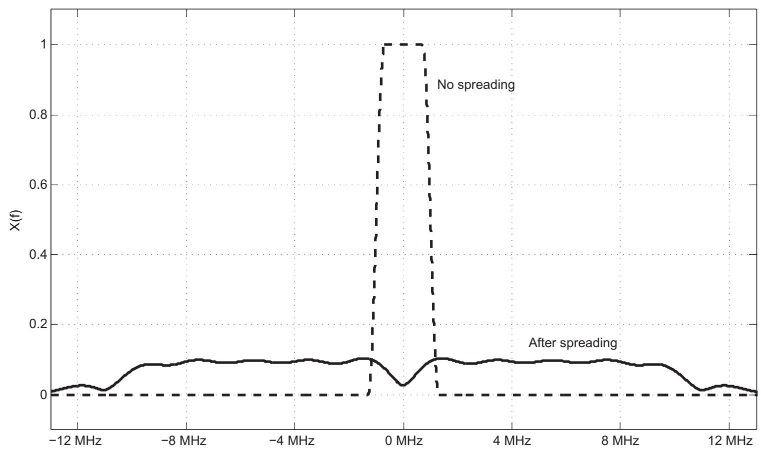

Fig. 9.5: Illustration of the spread spectrum technique. It consists of spreading spectrally the transmitted signal such that the signal is less sensitive to the frequency selectivity of the propagation channel or the interference of the other users in case of multiple-access scenario. As spreading sequence the one of IEEE 802.11b is considered: $1, -1, 1, 1, -1, 1, 1, 1, -1, -1, -1$.

Orthogonal Frequency Division Multiplexing. is an advanced Frequency Division Multiplex (FDM) transmission scheme for broadband wireless communication channels. A large number of closely-spaced orthogonal subcarriers are used to carry data. The payload data is divided into several parallel data streams or channels, one for each subcarrier. Each subcarrier is modulated with a conventional modulation scheme (such as QAM, see the description above on "Digital Modulation") at a low symbol rate, maintaining total data rates similar to conventional single-carrier modulation schemes in the same bandwidth. This is achieved by taking the inverse DFT of the set of modulation symbols (one for each subcarrier) and processing afterwards the stream of samples representing the time domain signal. At the receiver, the time domain samples are then transformed back by a DFT to the frequency domain to retrieve the modulation symbols per subcarrier. OFDM has developed into a popular scheme in modems for wired transmission over phone lines applied in Digital Subscriber Line (DSL) systems as discrete multi-tone modulation, providing quite large rates for data communications. However, today it is also applied in wireless digital systems such as DVB, DAB and wireless local/cellular systems (IEEE 802.11a/g/n, LTE, IEEE 802.16). The primary advantage of OFDM over single-carrier schemes is its ability to cope with severe channel conditions - for example, narrow-band interference or frequency-selective fading due to multipath - without complex equalization filters. Channel equalization is simplified because OFDM may be viewed as using many slowly-modulated narrow-band signals rather than one rapidly-modulated wide-band signal. The low symbol rate makes the use of a guard interval between symbols affordable, making it possible to eliminate Intersymbol Interference (ISI). Although OFDM has been successfully deployed, it has still some disadvantages compared to single carrier systems: sensitivity to Doppler shift, sensitivity to frequency synchronization problems which requires generally a complex synchronization unit at the receiver as well as a high peak-to-average-power ratio, i.e. large power difference between the weakest and the strongest amplitudes on subcarriers. Finally, the loss of spectral efficiency caused by cyclic prefix/guard interval might not be negligible. For instance, the loss is about 20% in IEEE 802.11 a/g/n systems.

OFDM in its primary form is considered as a digital modulation technique since it is utilized for conveying one bit stream over one communication channel using one sequence of OFDM symbols. However, OFDM can be combined with multiple access using time, frequency or coding separation of the users. One particular important multiple access combination with OFDM is the frequency separation referred to as OFDMA, see Section 10.1.2. It is applied today in LTE and IEEE 802.16e systems.

The next two paragraphs discuss three techniques for transmission systems with multiple inputs or/and multiple outputs (in general summarized under the acronym MIMO). Interest for MIMO systems has considerably increased over the last decade for two main reasons: the throughput increases linearly with the number of inputs and the probability of transmission errors decreases linearly with the number of receive antennas for basic scheme. The second reason is that current powerful processors can support such techniques in wireless standards (UMTS, LTE, IEEE 802.16).

Antenna Beamforming. Beamforming is a signal processing technique used in wireless transmission systems for directional signal transmission or reception. It exploits "spatial selectivity" among a single signal transmitted from multiple antennas, or received by multiple antennas, or both. If beamforming is applied at the transmitter, the same transmit signal is sent from each antenna. However, a beamformer controls the phase and relative amplitude of the signal at each antenna in order to create a pattern of constructive and destructive interference in the wavefront. Effectively, this gives the signal a preferred direction such that the SNR at the receiver is strongly increased. However, the transmitter must know the position of the receiver (or even better, its channel characteristics with respect to each transmit antenna). When applied for re- ceiving, the incoming signal from different antennas is combined in such a way by delaying some signals and multiplying with adequate complex coefficients (phase and amplitude). As a result, the received signal's SNR is strongly increased (but again the receiver has to know the direction from which the transmitted signal is received or even better the exact channel characteristic). If both transmitter and receiver have multiple antennas, beamforming may also be applied both at the transmitter and the receiver which leads to an even better SNR. However, beamforming can be computationally expensive if applied for several antennas at the transmitter or receiver (while also requiring control overhead). Hence, only basic beamforming techniques are being used in wireless standards today: Transmit antenna selection in GSM and UMTS which consists of selecting the transmit antenna that provides the best SNR for the considered terminal. For LTE, a pre-coding based beamforming with partial Space Division Multiple Access (SDMA) can be used optionally if the system supports MIMO techniques. A flexible system is IEEE 802.11n, which permits the application of beamforming simultaneously at the transmitter and receiver, based on the channel state information sent from the receiver to the transmitter by control frames. This technique is sometimes referred to as closed loop beamforming.

Space-time Coding/Decoding. Also in case of wireless systems with multiple transmit antenna, Space Time Coding (STC) is another method employed to improve the reliability of data transmission. Space-time coding relies on transmitting multiple, redundant copies of a data stream to the receiver in the hope that at least some of them may "survive" the physical path between transmission and reception in a good enough state to allow reliable decoding. Space time codes may be split into two categories:

1. Space-time trellis codes [458] distribute a trellis code over multiple antennas and multiple time-slots and provide both coding gain and diversity gain.
2. Space-time block codes [427],[456] act on a block of data at once (similarly to block codes) and provide only diversity gain, but are much less complex in implementation terms than space-time trellis codes.

UMTS, LTE and IEEE 802.11n support the basic Alamouti scheme [427]. Alamouti invented the simplest of all the STBCs in 1998 [427]. It was designed for a two-transmit antenna system and one receive antenna. It takes two time-slots to transmit two symbols such that it can achieve its full diversity gain without needing to sacrifice its data rate. The significance of Alamouti's proposal in 1998 is that it was the first demonstration of a method of encoding which enables full diversity with linear processing at the receiver. Earlier proposals for transmit diversity required processing schemes which scaled exponentially with the number of transmit antennas [157]. Furthermore, it was the first open-loop transmit diversity technique which had this capability. Subsequent generalizations of Alamouti's concept have led to a tremendous impact on the wireless communications industry.

Spatial Multiplexing. If a transmitter has data to be sent to multiple receivers and if it has multiple transmit antennas, it can actually transmit the information to all terminals simultaneously. This is referred to as spatial multiplexing [468]. In spatial multiplexing the transmit signal of the data for each receiver is concentrated by beamforming on the location of the receiver. However, this concentration in some preferred direction leads to a strongly attenuated signal transmitted into several other directions. These directions are the ones where further receivers can be served without having them suffer from interference of a simultaneous packet transmission. Hence, not all receiver distributions can be supported in a similar manner by spatial multiplexing. Spatial multiplexing is furthermore limited by the number of antennas the transmitter has, i.e. for n antennas up to n terminals can be served at the same point in time. In a similar manner a receiver with n antennas can receive up to n packets simultaneously by applying a beamformer. This is referred to as *space division multiple access* (see Section 10.1.3). As with spatial multiplexing, this works well for some transmitter position combinations while other combinations can not be resolved by the beamformer efficiently. Furthermore, if the receiver has multiple antennas then the constraint on the positions of the transmitters becomes less significant by the application of interference cancellation algorithms. In fact, the application of interference cancellation allows even for several, different packets transmitted over different antennas to the same receiver (if the receiver has multiple antennas as well). This is the most promiment MIMO scheme which is often characterized by a linear increase of the system capacity with the (minimal) number of antennas at the transmitter and receiver side. However, the complexity of especially the receiver is currently a strong limitation factor. Still, it is expected that this MIMO technique will be strongly used in future wireless standards.

Symbol Interleaving/De-Interleaving. Symbol interleaving is used in digital data transmission technology to protect the data against burst errors occurring during the propagation. As with bit interleaving, if a burst error occurs, too many errors can be made within one code word, and that code word cannot be correctly decoded. To reduce the effect of such burst errors, the symbols of a number of frames are interleaved before being transmitted. This way, a burst error affects only a correctable number of symbols in each frame, and the decoder can decode the frame correctly. LTE and DVB systems are using symbol interleaving.

9.2.3 Components of the Sample and Waveform Domain

Frame/Packet Synchronization. Once a digital train of samples is obtained, the receiving circuits (in the digital domain) first have to synchronize to the transmitter. This refers to fine tuning to the exact timing with which the modulation symbols are transmitted and to the exact carrier frequency used by the transmitter (there is always some frequency shift between any two oscillators, hence, requiring the receiver to identify the shift and correct it). Special training sequences (also referred to as preambles) are added by the transmitter to any data transmission which the receiver can use to easily acquire a precise enough synchronization. However, if the transmission occurs over a quite bad communication channel, already the step of synchronization can become very difficult and requires special design [406].

Channel Estimation and Equalization. After obtaining synchronization, the receiver has to identify the possible random distortions that the wireless channel causes to the signal (like phase shifts and attenuations/gains). This is known as channel estimation. For this, so called pilot signals are added to the transmit signal which are transmitted with a known strength and with a known phase. After the channel has been estimated, the distortions of the channel are compensated which is known as equalization. The proper estimation and equalization of the channel is a prerequisite for decoding the payload signal (as synchronization is a prerequisite as well). As an alternate solution, *differential modulation* (DAB) can be utilized. Since the data information is contained in the phase difference between two consecutive symbols, the data can be retrieved if the phase shift occurring during the transmission is equal for both symbols.

Pulse Shaping Filter/Matched Filter. In digital telecommunication, pulse shaping is the process of changing the waveform of transmitted pulses. Its purpose is to make the transmitted signal suit better to the communication channel by limiting the effective bandwidth of the transmission. By filtering the transmitted pulses this way, ISI caused by the channel can be reduced. Also, pulse shaping is essential for making the signal fit in its frequency band. Typically, pulse shaping is nowadays implemented in the digital domain before the digital-to-analog conversion. Two main pulse-shaping filters are used today in wireless communication systems: Either a Gaussian filter (like in GSM) or a raised-cosine filter [375, Chapter 9] (UMTS, LTE, IEEE 802.16, IEEE 802.11, DAB, DVB). The impulse response and the spectrum of the raised-cosine filter are plotted in Figure 9.6 for several values of roll-off factor β ranging from 0 to 1. The bandwidth occupied by the signal beyond the sampling frequency $1/2T$ is called the excess bandwidth and is usually expressed as a percentage of the sampling frequency. For example, when $\beta = 0.22$ as in UMTS standard, the excess bandwidth is 22%. The overall raised cosine spectral characteristic is usually split evenly between the transmitting pulse shaping filter and the receiving filter.

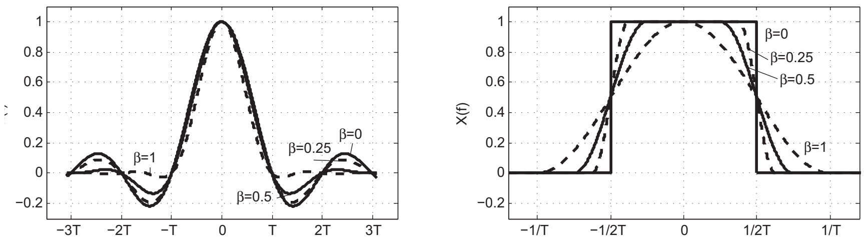

Fig. 9.6: Impulse response (left) and spectrum (right) of the raised cosine filter for several values of roll-off factor β. Most of the current standards are using raised cosine filter as pulse evenly split between the transmitter and the receiver (GSM: 0.2, UMTS: 0.22, DAB: 0.3).

Carrier Sensing/Energy Detection. Carrier sensing is usually employed by Carrier Sense Multiple Access (CSMA) protocols, as employed for example in the MAC of IEEE 802.11 systems. Through carrier sensing the transmitter evaluates the current state of the channel to determine whether the channel is idle or if it is currently busy with other data transmissions. There are different possibilities how to implement a carrier sensing algorithm. On the one hand there is energy detection which is purely checking for the current channel Received Signal Strength Indication (RSSI). If the RSSI is above a certain threshold – the so called Clear Channel Assessment (CCA) threshold – the channel is declared busy. Alternatively, a wireless device might try to sense a decodable signal on the channel, for example a header of a certain wireless system. Only if this header is decoded correctly, the channel is declared busy. The difference between this feature detection and pure energy detection is that in the first case all devices operating in the corresponding bandwidth can block the channel (by emitting energy) while in the second case only specific devices can block the channel. Carrier sensing is also used during the association process in all cellular networks (GSM, UMTS, LTE) that comprise multiple channels operating on orthogonal carrier frequencies. In these cases, the SINR of each channel is evaluated by sensing a beacon on each carrier.

DA/AD Conversion and IF Up/Down Conversion. In digital communication digital-to-analog conversion is used to convert the pulse shaping filtered output into an analog voltage which will be sent to the radio front-end of the transmitter. At the receiver, the reverse operation is known as analog-to-digital conversion. By the Nyquist-Shannon sampling theorem, a sampled signal can be reconstructed perfectly provided that the sampling frequency is at least twice as big as the Nyquist frequency of the transmitted signal in absence of noise [106]. However, even with an ideal reconstruction filter, digital sampling introduces quantization errors that make perfect reconstruction practically impossible. Increasing the digital resolution (i.e. increasing the number of bits used in each sample) or introducing sampling dither can reduce this error.

The performance of current digital-to-analog converters can support data rates as high as 1 Gigasample per second [217] with 16 bit integer resolution. Therefore, the up-conversion of the baseband signal to an intermediate frequency with values ranging from 1 MHz to 10 MHz in most of the devices can be done digitally.

9.3 Accurate Simulation of Physical Layers

With wireless transmission systems becoming more and more complex while the time-to-market is always decreasing, accurate simulations of upcoming standards are essential. Wireless communication standards become so involved due to the complexity of the system that the design and elaboration of the upcoming standards are done by means of using a simulator (and later in the process a hardware prototype).

Since most of the elementary operations in the components of the PHY layer are done on vectors of bits or symbols or samples, natural candidates are scientific programming languages, such as Matlab. As a point-to-point communication can be modeled as a chain of elementary matrix or vector operations, it is also natural to consider in addition block diagrams as in Matlab Simulink or Labview. The simulations consist usually of evaluating the symbol-, bit-, or packet-error rate of the transmitted PHY frames that have been corrupted by the modeled transmission channel. In order to take the randomness of the transmission system into account (randomness of the propagation medium, randomness of the transmitted data), a simulation provides the average performance for thousands of packet transmissions with different transmission channel coefficients and different data to transmit. The simulation flow graph is illustrated in Figure 9.7. Most of these Matlab

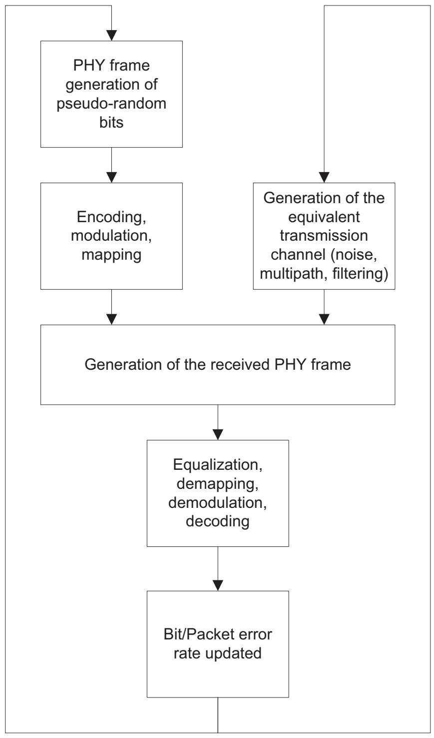

Fig. 9.7: Typical flow graph used for simulating performance of PHY layer transmission (Matlab, Labview, or C). The new algorithm(s) is (are) implemented either in the transmitter side (encoding, modulation) or on the receiver side (equalization which includes interference canceler, decoding,...).

implementations consider equivalent symbol timing and bypass network aspects beside interference within PHY frames. Queuing, network topology, network protocols (TCP, IPv6) are rarely considered for complexity issues since the computational complexity is increasing at least linearly each time a new random parameter is introduced (depth of the queue, location of the transmitter/receiver). Moreover, the computational complexity of the simulations increases polynomially (or even exponentially) with respect to the number of users so that the computational complexity of a simulation for a full cell is prohibitive and requires a higher level of abstraction as in the network simulators. However, this is basically not the main focus of Matlab or

Domains	Bit rate (Mbits/s)
Bit domain	22.8 Mbits/s
Symbol domain	28.5 Mbits/s
Sample domain	$>$ 914 Mbits/s

Table 9.4: Bit rate in the bit, symbol, and sample domains of the PHY layer for IEEE 802.11 system assuming a raw data rate of 10 Mbits/s at the MAC layer.

similar tools and there is no built-in libraries that allow networking considerations. Whereas their proposed environment allows fast development and design of advanced wireless techniques at the PHY layer, the next paragraphs discuss their main limitations. In the following we focus on three of them:

1. **Slow execution speed:** Whereas the Matlab environment is very convenient for developers, the fact that it is an interpreted language – especially when loops cannot be avoided – makes the simulations extremely slow. To have a better understanding of the computational complexity, the bit rate in the different domains of the PHY layer is calculated in Tab. 9.4 for IEEE 802.11 system with data rate of 10 megabits per second. From this example, it is clear that scientific programming languages as Matlab can deal well with the bit and symbol domains, but not with the sample domains (the computational complexity is 2 orders of magnitude more). One way to bypass this problem is to determine an equivalent transmission model in the symbol domain. This is very commonly done. However, there is a risk that the equivalent model does not capture accurately some effect as sample synchronization or collision at the sample level. It is also possible to use C for accelerating some functions that are not vectorial and require large *sequential* loops as in the Viterbi decoder. Matlab allows easily the integration of C functions (known as mex files).
2. **No discrete-event support:** While the sample or symbol processing at a transmitter or receiver imply a pseudo-continuous model of time, the processing steps of the higher layers of a wireless network happen much more asynchronous. In order to simulate this, discrete-event support is required, which is usually not provided. Furthermore, there is no library support for functions of the higher layers, meaning that all functionality needs to be implemented from scratch.
3. **Lack of interaction with network simulators:** While Matlab can be used via an interface for example from C code, the interfacing is not particularly easy to handle nor is it quite fast.

9.4 Physical Layer Modeling for Network Simulations

From the previous sections it is clear that physical layer simulations are quite demanding with respect to computational complexity and the required background knowledge. Hence, the question arises how physical layer behavior can and should be modeled in the context of network simulation, i.e. the typical simulation approach which only considers the packet domain.

A brute force method interfaces between a network simulator and simulator for the physical layer (like Matlab). While this provides a very detailed model, for larger deployments the approach does not scale with respect to complexity. Run times of simulations become so long, even for moderately complex topologies, that there is no benefit left from the detailed simulation model. Furthermore, analyzing simulation results becomes much more difficult as many more parameters have now an influence on the performance. Finally, as there is also a wide background knowledge required to design, execute and analyze such simulations, implementing the considered approach on a wireless prototyping platform might be a better choice. However, there is still tool support for fully detailed simulation models of wireless systems. A particular tool that can combine the different "simulation worlds" of higher layers (i.e. the simulators of the packet domain) and of the physical layer (among many other simulation domains) is Ptolemy from UC Berkeley [8]. Still, the above mentioned complexity problems remain.

Due to these reasons the common approach in wireless network simulation is to abstract from the many details of the physical layer. More precisely, almost all physical layer models in network simulators aim at capturing the physical layer impact on the transmission of frames or packets in terms of throughput, error probability and delay under the assumption of perfect synchronization. These models belong to the packet domain from Figure 9.1 and only take the following functions into account: FEC coding, digital modulation, advanced transmission schemes (like OFDM, spread-spectrum, antenna beamforming and space-time coding) and carrier sensing. All the other functions are assumed to work perfectly without any performance degradation (which is a very strong assumption compared to reality). Thus, only the packet error process (along with throughput and delay) can be taken into account. These models can not provide a "bit-true" characterization of the PHY, where the impact of the PHY layer on the transmission of each bit is modeled accurately. The difference between these two approaches leads to the important consequence that for packet-domain models only a statement can be derived if a packet/frame is received incorrectly but not which bits are wrong (which can be obtained from a bit-true model). Knowing the position of a bit error has some relevance for higher layers. If a header is corrupted the packet is most likely discarded. On the other hand, if payload bits are corrupted this may have quite different consequences for multimedia applications like video or voice, which can tolerate some bit errors. The quality degradation depends on the exact position of the bit error. However, bit-true physical layer simulation models are much more complex regarding their implementation.

Apart from other issues, two components are essential for all packet-domain models. On the one hand, most packet-domain models determine the channel quality between a transmitter and the corresponding receiver. Aspects of this step relate to considering path loss, shadowing, fading, noise and interference in combination with the used antenna and possibly advanced transmission schemes (like OFDM, spreading or MIMO systems). Another aspect of this step involves the question if the channel quality is accounted for on the average or if an instantaneous channel quality is considered, i.e. one value or even multiple values per transmitted packet. On the other hand, most packet-domain models translate the channel quality into an error rate (mostly a bit-error rate). This mapping can be quite complex and involves the modeling of the digital modulation as well as the FEC scheme. In the following, we give an overview of both modeling steps. Note that there are further aspects to be considered for a PHY layer packet-domain model, as discussed in Section 9.5.

9.4.1 Link-to-System Interface

In the following, we first discuss the mapping of the channel state to a bit-error probability for narrow-band, single-carrier transmission systems, i.e. for systems without advanced transmission schemes assuming that the channel gain is flat (see Section 11.7). For this we make in this section initially the assumption that the channel quality is fixed. This type of model is also referred to as static channel (see Section 11.12). In this case we can find a mapping of the channel state and the chosen transmission parameters (modulation type, transmit power, coding scheme, etc.) into a resulting physical layer behavior on the packet domain in terms of packet throughput, delay and bit-error rate. This mapping is referred to as *link-to-system interface*. In its simplest form, it is based on the fact that for a certain received channel quality, as measured by the SNR, the bit error rate can be derived depending on the chosen modulation type and transmit power. Once the bit error rate is determined, the corresponding packet error rate can be obtained as explained in Section 10.2. The SNR is given by

$$\gamma = \frac{P_{\mathrm{tx}} \cdot h^2}{\sigma^2} \tag{9.1}$$

where P_{tx} is the transmit power, h^2 is the channel gain (see Chapter 11) and σ^2 is the equivalent background noise power of the transmission. The transmit power P_{tx} is usually well known during network simulation, even if it is adapted by the transmit node. Therefore, it is readily available for the computation of the SNR. This is also true for the noise power which depends on specific PHY parameters of the receiver (see Section 11.10). Furthermore, h^2 denotes the channel gain between the transmitter and the receiver, which we assume to be constant initially. In this case the computation of the resulting bit error rate from a given SNR is rather easy and is also quite close to the real system behavior. Either exact or approximate formulas are used, as derived in [100] for QAM systems. If no formulas exist, the considered modulation system might still have been investigated by related work providing a SNR-to-BER curve. This curve can then be converted into a look-up table to be used in the simulation. Finally, if no data on the modulation system exists, the only way to obtain an SNR-to-BER curve is to perform extensive and accurate PHY simulations, for example using Matlab. Then, the obtained curve can be be mapped into a look-up table as shown in Table 9.5 for the four different modulation types of the IEEE 802.11b standard as taken from [355]. Given the bit-error rate, the next step is to determine the

SNR (dB)	**BPSK (1Mbps)**	**QPSK (2Mbps)**	**CCK5.5 (5.5Mbps)**	**CCK11 (11Mbps)**
...	...	...	...	...
-5	6e-2	0.5e0	0.5e0	0.5e0
-4	2e-2	0.5e0	0.5e0	0.5e0
-3	8e-3	0.5e0	0.5e0	0.5e0
-2	4e-3	1e-1	0.5e0	0.5e0
-1	1e-4	8e-2	0.5e0	0.5e0
0	3e-5	2e-2	0.5e0	0.5e0
1	1e-5	5e-3	8e-2	0.5e0
2	1e-6	1.2e-3	4e-2	0.5e0
...	...	...	...	...

Table 9.5: An example Bit-Error Rate (BER) lookup table for the four different (uncoded) modulation types of the IEEE 802.11b Wireless Local Area Network (WLAN) standard.

packet error rate. This step is described in detail in Section 10.2. Once the packet error probability is obtained, for each transmitted packet a random decision is performed according to the packet error rate threshold and the corresponding packet is then marked to be either erroneous or not.

So far we have considered a static channel quality with a single modulation type and a simple (single-carrier) transmission system. The mapping from SNR to bit-error rate gets already more complicated if a FEC coding scheme is assumed. To account for FEC coding, two general approaches exist. Either the coding effect is taken into account by modifying the SNR. In this case, coding simply “increases” the SNR leading to a better bit error rate. However, this assumes a constant coding gain between the coded and uncoded system which is usually not the case at high or low SNRs. Hence, one has to obtain a detailed look-up table for the coded bit error rate of the FEC codes in combination with the modulation scheme depending on the SNR. If such tables are not provided by books and research papers, they have to be obtained from extensive physical layer simulations (usually performed in the symbol- or sample domain of Figure 9.1). There are also limited ways to capture the coded system behavior by formulas, see Section 12.1.2 for an example mapping for convolutional coding. Still, ultimately a packet error rate is obtained and for each transmitted packet a binary decision is performed if the packet is erroneous or not.

Next, let us consider a varying channel gain. In wireless systems a varying channel gain is almost always encountered in reality. Hence, it is likely to be included in a simulation study. The channel gain depends in general on the distance between transmitter and receiver (therefore, the chosen mobility model – see Section 14 – has an impact on the channel gain), but there are also additional time-varying, random components to the channel gain referred to as shadowing (see Section 11.6) and fading (see Section 11.7). All these effects ultimately lead to a varying h^2 in Equation 9.1 and thus the SNR γ varies over time. A quite common assumption for such cases is that the channel gain h^2 is constant during a single packet transmission but varies in between. Such channel models are also referred to as block-fading channels (see Section 11.12). This leads to determining an instantaneous SNR at the time a packet is transmitted. From this instantaneous SNR an instantaneous bit-error rate is determined using the same method as above (formulas or look-up tables for the modulation and coding scheme considered). Finally, a packet-error rate is determined and a random decision is performed if the packet is received correctly or not. Depending on the considered distribution of the channel gain, this method can lead to a very different average packet error rate behavior than considering a static channel quality. This is important to note if one is only interested in the average PHY layer behavior but fading or shadowing is to be taken into account.

9.4.2 Equivalent Channel Quality Models

The modeling of the channel quality and the corresponding PHY layer performance becomes more complicated if the channel quality is assumed to be variable during a packet transmission. This can happen due to fading, as explained in Section 11. However, interference can also contribute to a varying channel quality during a packet transmission. If interference is present, the channel quality is measured by the SINR as given below:

$$\gamma = \frac{P_{\text{tx}} \cdot h^2}{\sum_{\forall j} P_j^I \cdot h_j^2 + \sigma^2} \tag{9.2}$$

In this case, the received power in the numerator is divided by the noise power, denoted by σ^2, as well as the sum over all interfering signals multiplied by the respective channel gains between the interference sources and the considered receiver. Note that these channel gains might all be subject to stochastic variations which makes the analysis of such scenarios quite complicated.

If either fading or interference are time varying within the packet transmission, the common approach is to consider an *equivalent SINR*, meaning that a constant substitute SINR has to be found which results in the same packet error rate as the varying channel has. In general this is quite difficult and has to be redone every time a new PHY architecture or a new channel behavior is considered. For example, if three levels of channel quality are assumed to occur during a packet reception, the equivalent channel quality can be computed by the average of these three levels (weighted by their durations). However, as the mapping from channel quality to bit error rate is usually non-linear, a better approach is to average the corresponding bit error rates of the three levels weighted by the durations. Note that the correctness of this averaging for an equivalent model depends on the modulation, FEC coding and interleaving scheme used. Still, it is the best that can be done for packet-domain models. If more accurate models are to be considered, a bit-true model must be employed.

9.4.3 Modeling Advanced Transmission Systems

An accurate modeling of the PHY layer for network simulation becomes more complicated if advanced transmission systems are considered even if the transmitter/receiver pair is assumed to be perfectly synchronized (and hence the components involved in synchronization are not considered). The main reasons for the modeling difficulties are the following:

1. **Interaction between the channel and advanced transmission schemes:** Most current and upcoming standards for wireless systems employ a system bandwidth which is much larger than 500 kHz. For such bandwidth the channel becomes frequency-selective (see Section 11.7). Even for simple transmission schemes the performance on top of a frequency-selective channel is not easy to characterize. This becomes much harder if advanced schemes are employed. Even worse, if mobility is assumed, the channel might become time-selective, which adds to the modeling complexity. Finally, all these arguments also apply to interfering signals, which interact with the advanced transmission system as well. Especially if many possible transmitter/receiver/interferer constellations are considered in a large-scale simulation, the scalability of the simulation model becomes crucial [392].
2. **Multi-parameter dependency:** Any model of advanced transmission schemes such as OFDM, MIMO or spread-spectrum requires a lot of parameters to characterize the input/output behavior. It is difficult to plug a statistical characterization to each parameter, especially when they are

not mutually independent. This applies for instance to the fading gain coefficients between the antennas in MIMO systems and/or the fading gain coefficients between subcarriers in OFDM systems. Some interesting solution based on the principle of maximum entropy has been presented recently for MIMO systems [115]. Furthermore, models for so called "outage analysis" have been derived recently. The outage probability has been first considered by Shamai [345] in the context of vehicular networks for simple transmission systems. Later, this approach has been extended to more advanced schemes like MIMO [287], OFDM [114], and spread spectrum [478]. The analysis of the outage probability however has several drawbacks. Most importantly, all mentioned work consider Gaussian signaling instead of discrete constellation settings (i.e. Shannon capacity versus real modulation schemes) and the outage probability can be determined only for specific channel behaviors like the Rayleigh fading distribution.

3. **Advanced coding schemes on top of advanced transmission schemes**: For advanced FEC coding schemes– especially the decoding algorithms invoked at the receiver– there is often no analytical relationship between the input(s) and the output(s) even if simple transmission schemes are considered. Notable examples are Turbo-codes for which the algorithm was found before any analytical framework was proposed. Hence, the performance gain stemming from these advanced FEC schemes is difficult to quantify. Several new methods addressing this problem have been proposed recently [461, 389]. However, they require large computational power and can only model basic schemes.
4. **Adaptation and channel feedback**: Finally, many advanced transmission schemes are applied in an adaptive manner, i.e. there is a feedback loop from the receiver to the transmitter with channel state information and the transmitter modifies its behavior depending on this feedback. In fact, most of the current standards support feedback channels for transmitting periodically some channel state information such as acknowledgment frames (IEEE 802.11, IEEE 802.16 and LTE) or even channel state information (IEEE 802.16, LTE). Modeling of the behavior of such systems is generally difficult since it can require for example application of control system theory (complex Markov process) or some notion of the transformed channel behavior. Notable works in this field are [326, 179].

Due to these many difficulties, accurate performance models of advanced transmission schemes in the PHY layer are a challenging and still open research field today while they are essential for network simulation in the future. In the next paragraph, we illustrate the problems of modeling an OFDM system accurately in the context of IEEE 802.11 WLAN

In addition to the time-varying channel behavior, the modeling of the performance of an OFDM system requires some assumption about the frequency-varying channel behavior. Let us consider a block-fading channel behavior in the time-domain. The simplest assumption for the frequency domain is to model it static. In this case, the channel quality, varying from packet to packet transmission due to the block-fading assumption, is the same for all subcarriers. We further assume that the same modulation type is used per subcarrier. Hence, all subcarriers have the same SNR and bit-error rate. Based on the bit error rate, a packet error process can be obtained in a similar way as discussed above. Even if FEC coding is applied, the coding scheme can be taken into account by either shifting the SNR (which yields a better bit error rate) or by considering a direct mapping between input bit error rate and output bit error rate for the specific code considered.

The modeling already gets much more complicated if no static channel behavior in frequency can be assumed. In reality, this is the usual case in broadband OFDM systems, especially in indoor communication scenarios with multiple reflections recombining at the receiver. A suitable assumption in this case is to assume a narrow-band flat fading channel attenuation for the whole subcarrier spacing, but this attenuation varies in general from subcarrier to subcarrier, potentially in a correlated manner (see Section 11.7). Hence, per subcarrier we obtain a different SNR and therefore a different bit error rate. Mapping this bit error rate to a packet error probability is only possible without further assumptions if neither coding nor interleaving is applied. In this case, one simply determines the probability that all bits have been transmitted correctly (based on the recombination pattern of forming the payload packets from the parallel subcarrier bits). However, if coding and interleaving are applied, further simplifying assumptions are required. In particular, one can assume "perfect" interleaving in frequency. This allows then to average the bit error rate per subcarrier yielding a joint channel bit error rate of the system. Next, if coding is applied the joint bit-error rate might be mapped by an input-output bit-error rate characterization into a joint coded bit-error rate from which the packet-error probability can be derived. However, this method is based on a large set of assumptions. It is discussed in detail in Section 12.1.2 and in [42] for convolutional coding. This method can also be applied if different modulation types are employed per subcarrier.

From the above discussion it is clear that the probability of error of a point-to-point transmission in advanced transmission schemes is generally unknown due to the randomness of the transmission medium and the complexity of the input-output relationship of each component at the receiver. This also applies to spread spectrum systems and to MIMO systems (and more generally to any multiple access system with random interference). Even if sophisticated modeling techniques based on the statistics of the random variables of the transmission system have been developed in the past years [115], there is no general technique available. Hence, the only alternative today is to use look-up tables that can accurately predict the end-to-end performance of a PHY layer as soon as advanced techniques are used and combined.

9.5 An Example Packet Domain Physical Layer Simulation Model

In addition to the general considerations of modeling the PHY for network simulations, in the following we present an exemplary packet domain PHY simulation model that explicitly formulates the PHY functionalities required in most network simulations. The described model particularly provides the PHY functionality necessary to support DLL modeling as described in Chapter 10. Figure 9.8 shows the general structure of the presented PHY model and its interface to the DLL. It generally consists of two units: a *radio unit* that models actual radio hardware component functionality, and an *evaluation unit* that includes all functionality that is necessary for simulation but has no counterpart in real hardware, including the link-to-system model described above. Note that we do not rely on a specific link-to-system model, but keep the model open for using an arbitrary one. An exemplary implementation of the model can be found in the *MiXiM framework* [319, 262, 493] for *OMNeT++* [475].

9.5.1 Radio Unit

In general, each communication entity can act as a transmitter or receiver. Whether the entity acts as a receiver or transmitter is usually determined by the state of its radio. In principle, radios can be categorized according to their dialog modes into simplex, half-duplex, and full-duplex radios. While full-duplex radios are capable of sending and receiving simultaneously at all times (e.g. by using different frequency bands), half-duplex radios can perform one task at a time only. Moreover, in energy aware networks, devices might support sleeping modes, in which the radio is powered down and, thus, not able to send or receive frames. In the following, we focus on half-duplex operation featuring sleeping modes as the most general case. All other cases can be derived from that by setting parameters accordingly.

A half-duplex radio has three states: *send*, *receive*, and *sleep*. The radio state is controlled by the DLL, i.e. switching information is passed as *PHY control info* via the DLL-PHY interface. Note that it takes time to switch the radio from one state to another. Depending on the investigation, these *Radio Switching Times* might need to be modeled as well.

Transmitting Frames. For an entity to act as a transmitter, its radio has to be in the send state. Additionally, it has to be ensured that no other frame is currently being sent from the same node at the same time using the same resources. Normally, the DLL layer should only pass a message to the PHY if both conditions are fulfilled. In order to ensure the correct interaction between DLL and PHY and catch potential errors at an early point, both conditions should be checked and violations be reported (see Figure 9.9).

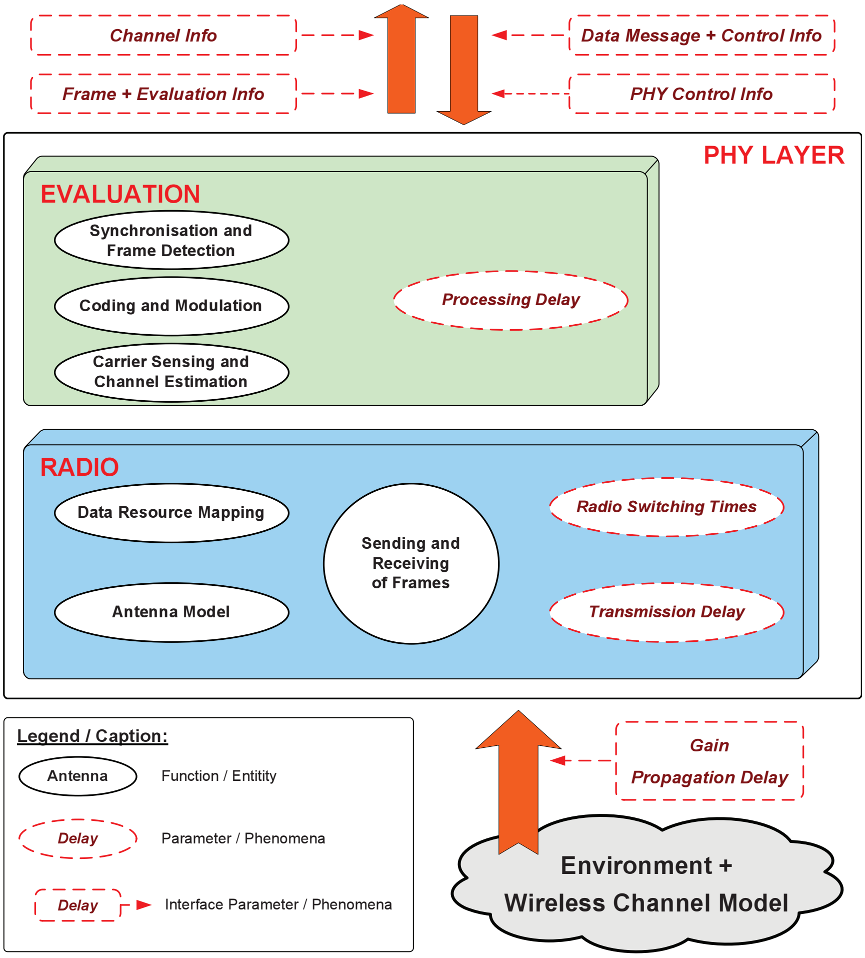

Fig. 9.8: Functional blocks of the presented PHY model including the interfaces to the environment/channel and the DLL.

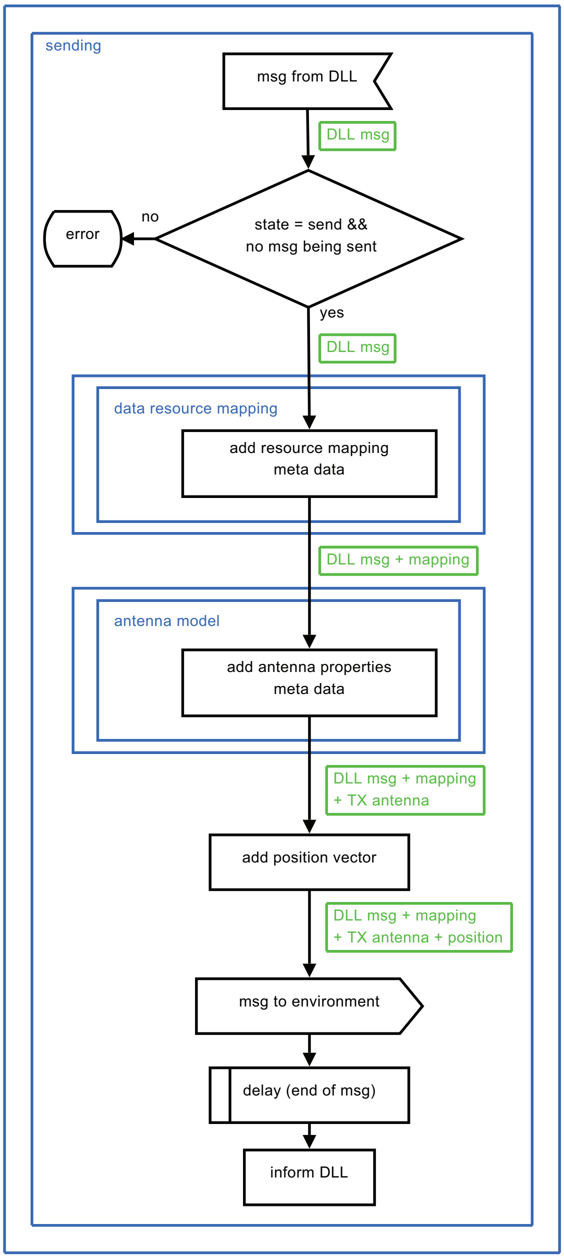

Fig. 9.9: Sending process

As mentioned earlier, the detailed effects of FEC coding and interleaving are modeled by bit-true simulation models on the bit-, symbol-, or sample domain (see Section 9.2 & 9.3). For packet domain simulations, the impact of coding and interleaving are not considered per bit, but evaluated in combination with the chosen modulation scheme as well as other transmission parameters (cf. Section 9.4.1). As all this information has to be processed at the receiver, meta data is added to the transmitted frame which contains all information that the receiver needs to accurately determine the reception of the frame. Appending all information regarding the transmission parameters is done in the *Data Resource Mapping* block. In addition to typical transmitter settings (like modulation and coding type or transmit power), also further scheduling decisions regarding the used resources have to be appended like the used sub channels in OFDMA systems or the used spatial streams in multiple antenna systems, see Section 10.1.5. Finally, information is added regarding the used antenna type and its impact on the transmit power, see Section 11.11.

The frame with its meta data is then passed to the environment (which models the channel including interference effects among other elements, cf. Chapter 11). The environment is responsible for delivering the frame to the appropriate receiving entities. Since the wireless medium is a broadcast channel, there might be more entities than the specified receiver of the frame. In order to correctly calculate the channel gain at the moment of the frame arrival at the receiver, the transmitter additionally needs to provide its momentary position, speed and direction of movement (referred to as a *position vector*) to the environment. Before passing the frame to the environment, the sending and receiving block thus attaches the position vector as meta data to the frame.

Since in simulation the whole frame is "sent" at once, the *Transmission Delay* has to be modeled explicitly, i.e. the time between sending of the first and the last bit of the frame. This is usually done by some delay process. Once the last bit of the frame is sent, the PHY has to inform the DLL, so that it can take appropriate actions (e.g. sending the next message or switching the radio back into the receive or sleep state). A state diagram for the whole sending process is shown in Figure 9.9.

Data Resource Mapping. The task of this functional block is to map the data for transmission on the system resources as determined by a resource manager (see Section 10.1.5). However, since we do not consider a bit-true model, only PHY frames with an attachment of meta data are considered. The meta data is necessary for the receiving entity to be able to process the transmitted frame. It comprises the transmission power that is reflected by the RSSI in a real system. In adaptive modulation and coding systems, the selected parameters for modulation and coding need also be attached in order to enable the receiver to correctly evaluate the transmitted data. In multi-carrier and/or multi-antenna systems, this information is necessary per carrier/antenna.

The decision on these transmission parameters is made at the transmitter by the DLL and then passed as *control info* to the PHY layer via the DLL-PHY interface. An example for a complete set of parameters in a MIMO multi-carrier transmission system can be the following: "use sub-bands 4, 6, and 8 for transmission on antenna 2: use QPSK and 100 mW on sub-band 4, QPSK and 50 mW on sub-band 6, and 64-QAM and 200 mW on sub-band 8."

Receiving Frames. The received analog signal is attenuated and distorted by the environment it traveled through on its way from the transmitter to the receiver. Depending on the modeled system, the distortion has to be calculated for different subcarriers, antennas, and once or multiple times per frame in the time domain. These effects are described in detail in Chapter 11. In order to calculate the distortions, the position vector of the receiver (as well as of the transmitter) is required. All this information is appended to the frame's meta data before it is delivered to the receiver's PHY. The details of the receiving process are shown in Figure 9.10 and described in the following.

The decision whether an entity is able to receive a frame or not depends on several parameters. First, its radio has to be in receive mode. Second, only one frame can be received at a time. Thus, if the PHY is currently in the process of receiving a frame, all other frames arriving at that time will be treated as noise (and added to the "noise messages" as shown in Figure 9.10). Alternatively, the receiver might also lock to the strongest signal received, even if it arrives later than some other frame (capturing effect). Which frame to receive and which to treat as noise is determined by the synchronization and frame detection block. Once the frame is received, bit errors have to be calculated according to some link-to-system interface.

9.5.2 Evaluation Unit

Synchronization and Frame Detection. For each frame arriving at the PHY layer, the receiver has to decide whether to receive it or treat it as noise. The point in time for the decision (denoted as t_1 in Figure 9.10) depends on the chosen model. One possibility is to make the decision immediately after the packet's arrival ($t_1 = 0$). This is useful if only a single channel gain value is present for the whole frame. Another possibility is to decide at a later time, e.g. after the preamble or header is received and there is at least one channel gain value corresponding to the preamble transmission. The decision whether to receive the frame or treat it as noise is based on the channel gain and other transmissions currently ongoing on the medium. The PHY has to derive the

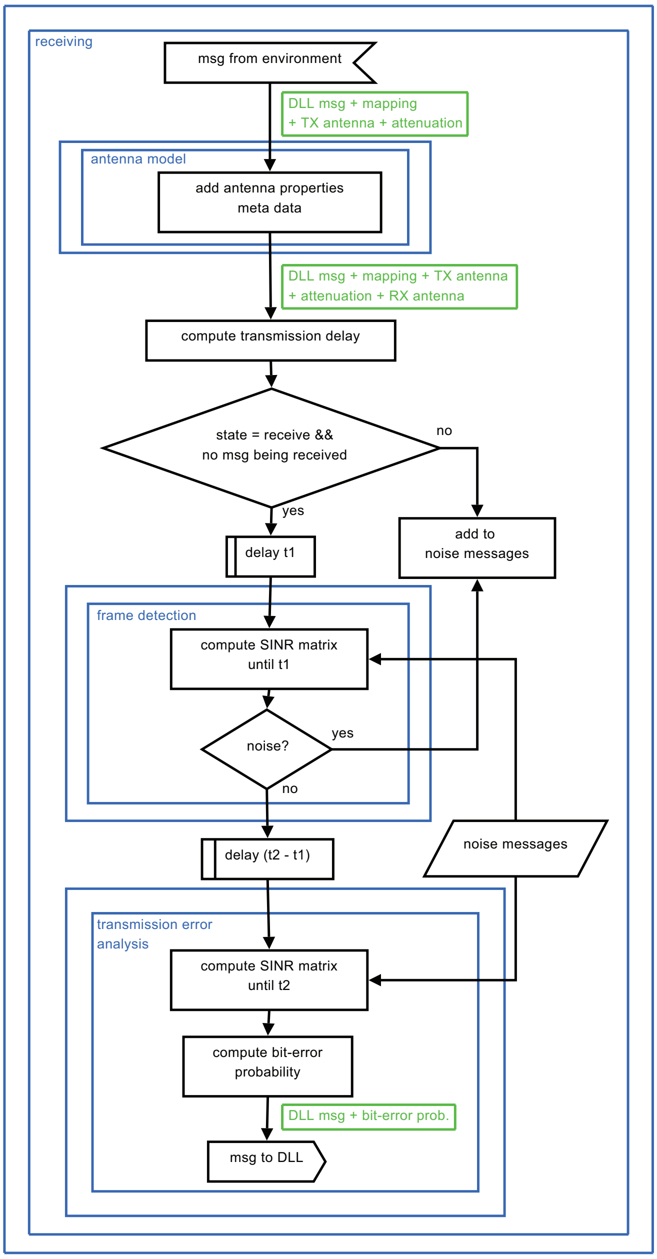

Fig. 9.10: Receiving process

SINR values for the frame or the part of the frame it has already received. If desired, this is also the place to model capturing effects, as mentioned above.

If the message is treated as noise, it is added to the noise power. If not, it has to be delayed until the end of the frame (t_2) and passed to the resource de-mapping, demodulation, and decoding functional block, which calculates the bit-error probability and passes the message to the DLL.

An example is shown in Figure 9.11. Here we assume, that a frame is received, if its header can be decoded. In this case "msg2" is the message being evaluated. At the start of receiving "msg2" there has already been "msg1" on the channel, which is assumed to be treated as noise. At time t_1, the header of "msg2" is completely received. In order to evaluate the SINR at this point, all interfering messages have to be considered. In Figure 9.11 these are "msg1" and "msg3" – "msg4" does not intersect with the header and thus is not of interest. Later, at time t_2, also "msg4" is of interest to calculate the bit errors as described in the *Demodulation, and Decoding* sub-section.

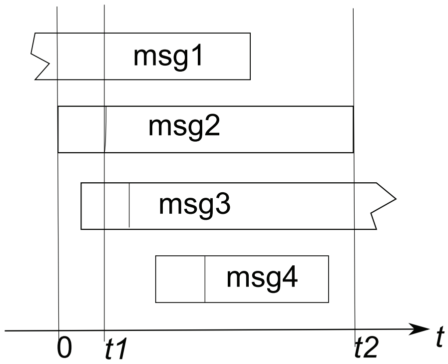

Fig. 9.11: Receiving of multiple messages

The attached channel gain h^2 and transmission power values P_{tx} of the frame in question ("msg2") and the corresponding values of the interfering gains h_j^2 and the interfering transmit powers P_j^I of all intersecting frames i ("msg1" and "msg3"), as well as the noise σ^2 are used to calculate the SINR as shown in Equation (9.3).

$$\gamma = \frac{P_{\text{tx}} \cdot h^2}{\sum_{\forall j} P_j^I \cdot h_j^2 + \sigma^2} \tag{9.3}$$

Note that in general the SINR is not only a single value, but a matrix of values in frequency (different sub-channels), space (different antennas), and time (if multiple values are to be considered per frame time or if advanced transmission schemes are employed).

Channel Estimation. As mentioned above, channel estimation is performed on the one hand at the receiver to better decode the modulation symbols. For a packet domain simulation model, this type of channel estimation is not considered but is assumed to function perfectly. On the other hand, channel estimation is required such that the transmitter can adapt transmission parameters by resource management to the channel state. This type of channel estimation is often considered by packet domain PHY simulation models. One way to model this is to let the environment determine the channel state at frame transmission analytically, according to the estimation algorithm applied. The resulting channel gains are attached to the frame as meta data (see Figure 9.10). The channel estimation part is modeled by passing the received channel attenuation values on to the DLL that saves it together with a time stamp and transmitter information for future use. In a Time Division Duplex (TDD) system, if channel reciprocity can be assumed, the stored channel state information can be used for transmission parameter adaptation (e.g. power, modulation, coding), once the receiver in turn wants to transmit something to the former transmitter. Whenever channel reciprocity cannot be assumed, e.g. in a Frequency Division Duplex (FDD) system, the receiver's DLL needs to signal the channel information back to the transmitter in order to enable it to adapt the transmission parameters in the upcoming slots. For signaling the channel state information back to the transmitter an additional signaling channel is needed.

Carrier Sensing. Carrier sensing at the transmitter is necessary for particular MAC protocols such as CSMA (like in 802.11, see Section 12.1). Carrier sensing is a service provided by the PHY to the DLL. Whenever the MAC wants to access the channel, it requests the carrier sensing information from the PHY. The carrier sensing functional block then has to provide the channel status information and pass this information via the PHY DLL interface. Based on this information, the MAC decides whether the channel is busy or idle. Possibilities to model carrier sensing range from simply introducing a random delay (done at the DLL) to explicitly modeling the carrier sensing process as described above in Section 9.2.

Depending on the type of carrier sensing, feature detection versus energy measurement, the PHY either has to calculate the RSSI or evaluate the SINR to determine whether the signal is decodable or not. One option to decide on the decodability of a frame is to consider either the SINR or the corresponding BER. The calculated value (RSSI, SINR or BER) is then compared to some threshold in order to decide whether the channel is busy or idle. In addition, the DLL has to specify how long the carrier sensing needs to be performed. In simulation, it is often assumed that carrier sensing can be done in zero time. In this case the PHY would just evaluate the most recent SINR available. For more accurate simulations however, the delay for the carrier sensing should be simulated.

Demodulation and Decoding. Coding and modulation have a major impact on the transmission quality of a frame. At the packet domain, this impact can be accounted for by a link-to-system interface (as explained above) which takes the SINR as input and determines an instantaneous BER as output. If no detailed description of the coding impact is at hand, a simple coding gain can be assumed which is added to the SINR of the frame. The calculation of transmission errors is usually done once the frame is completely received (t_2). The receiver has to calculate the SINR as described above and shown in Equation (9.3) and Figure 9.11. The SINR is used together with the BER curves or tables to get the bit-error probabilities for the message: the combination of SINR and indicated modulation/coding choice is taken as an input for the table or curve look-up, the error probability is the output (as shown in Table 9.5). The level of detail of the bit-error calculation mainly depends on the number of SINR values available. The simplest model is to only have one SINR value per frame. More detailed models have multiple values to evaluate multiple sections of the frame. One example is to have one value for the PHY header which are usually transmitted at a lower modulation to ensure the correct frame detection, and the payload, respectively. Furthermore, there might be another value to distinguish between the MAC header and the MAC payload in order to be able to evaluate both of them individually. The PHY payload is passed to the DLL layer via the PHY-DLL interface accompanied by the bit-error probability meta data.