1

WIRELESS DATA COMMUNICATIONS

Ce n'est pas possible!...This thing speaks!
Dom Pedro II, 1876

The telecommunication revolution was launched when the great scientist and inventor Alexander Graham Bell was awarded in 1876, US patent no. 174,465 for his "speaking telegraph." His resolve to help the deaf led to his perseverance to make a device that would transform electrical impulses into sound. Bell's telephone ushered a revolution that changed the world forever by transmitting human voice over the wire. Dom Pedro II, the enlightened emperor of Brazil (1840 to 1889), could not be easily convinced of the telephone's ability to talk when Bell provided explanations at the Philadelphia centennial exhibition in 1876. But lifting his head from the receiver exclaimed "My God it talks" as Alexander Graham Bell spoke at the other end (see Huurdeman (2003), pages 163–164). Bell had been trying to improve the telegraph when he came upon the idea of sending sound waves by means of an electric wire in 1874. His first telephone was constructed in March of 1876 and he also filed a patent that month. Although the receiver was essentially a coil of wire wrapped around an iron pole at the end of a bar magnet (see Pierce (1980)), there is no doubt that Bell had a very accurate view of the place his invention would take in society. At the same time he stressed to British investors that "all other telegraphic machines produce signals that require to be translated by experts, and such instruments are therefore extremely limited in their application. But the telephone actually speaks" (see Standage (1998), pages 197–198). This chapter is dedicated to the important topic of signaling that makes communication possible in all wireless systems.

Nowadays, wireless communications are performed more and more using software and less and less using hardware. Functions are performed in software using digital signal processing (DSP). A very flexible setup is the one involving a general hardware platform whose functionality can be redefined by loading and running appropriate software. This is the idea of software defined radio (SDR). In the sequel, the focus is on the use of SDRs, rather than on their design. Essential DSP theory is presented, but only to a level of detail

Principles of Ad hoc Networking Michel Barbeau and Evangelos Kranakis

required to understand and use a SDR. The emphasis is on architectures and algorithms. Wireless data communications are addressed from a mathematic and software point of view. Detailed SDR hardware design can be found in the book by Mitola (2000) and papers from Youngblood (2002a,b,c, 2003). Detailed DSP theory can be found in the books by Smith (1999, 2001) and in a tutorial by Lyons (2000).

The overall architecture of an SDR is pictured in Figure 1.1. There is a transmitter and a receiver communicating using radio frequency electromagnetic waves, which are by nature analog. The transmitter takes in input data in the form of a bit stream. The bit stream is used to modulate a carrier. The carrier is translated from the digital domain to the analog domain using a digital to analog converter (DAC). The modulated carrier is radiated and intercepted using antennas. The receiver uses an analog to digital converter to translate the modulated radio frequency carrier from the analog domain to the digital domain. The demodulator recovers the data from the modulated carrier and outputs the resulting bit stream. This chapter explains the representation of signals, Analog to digital conversion (ADC), digital to analog conversion, the software architecture of an SDR, modulation, demodulation, antennas and propagation.

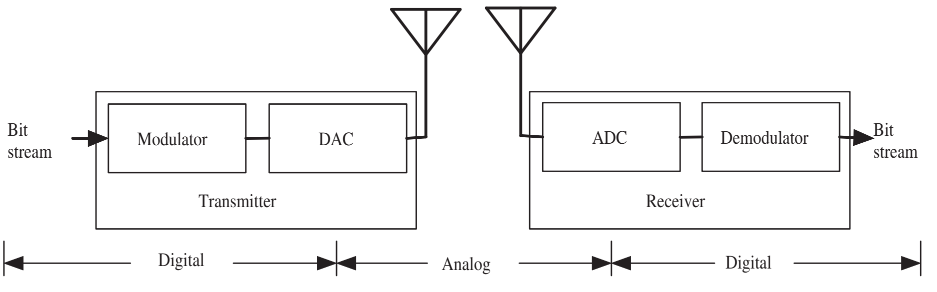

Figure 1.1 Overall architecture of an SDR.

1.1 Signal representation

This section describes two dual representations of signals, namely, the real domain representation and the complex domain representation.

A continuous-time signal is a signal whose curve is continuous in time and goes through an infinite number of voltages. A continuous-time signal is denoted as $x(t)$ where the variable t represents time. $x(t)$ denotes the amplitude of the signal at time t, expressed in volts or subunits of volts. A radio frequency signal is by nature a continuous-time signal. $\cos(2\pi f t)$ is a mathematic representation of a continuous-time periodic signal with frequency f Hertz. A signal $x(t)$ is periodic, with period $T = 1/f$, if for every value of t, we have $x(t) = x(t + T)$. A discrete-time signal is a signal defined only at discrete instants in time. A discrete-time signal is denoted as $x(n)$. The variable n represents discrete instants in time. A discrete-time sampled signal is characterized by a sampling frequency f_s, in samples per second, and stored into computer memory.

The aforementioned representations model signals in the real domain. Equivalently, signal can be represented in the complex domain. This representation captures and better explains the various phenomena that occur while a signal flows through the different functions of a radio. In addition, in the complex domain representation the amplitude, phase and frequency of a signal can all be directly derived. The complex domain representation of a signal consists of the in-phase original signal, denoted as $I(t)$, plus j times its quadrature, denoted as $Q(t)$, with $j = \sqrt{-1}$:

$$I(t) + jQ(t).$$

The quadrature is just the in-phase signal whose components are phase shifted by 90°. The representation of such a signal in the complex plane helps grasp the idea. A periodic complex signal of frequency f whose in-phase signal is defined as $\cos(2\pi ft)$ is pictured in Figure 1.2. Its quadrature is $\sin(2\pi ft)$, since it is the in-phase signal shifted by 90°. The evolution in time of the complex signal is captured by a vector, here of length 1, rotating counterclockwise at $2\pi f$ rad/s. The value of $I(t)$ corresponds to the length of the projection of the vector on the x- axis, also called the *Real axis*. This projection ends at position $\cos(2\pi ft)$ on the Real axis. The value of $Q(t)$ corresponds to the length of the projection of the vector on the y-axis, also called the *Imaginary axis*. This projection ends at position $\sin(2\pi ft)$ on the Imaginary axis.

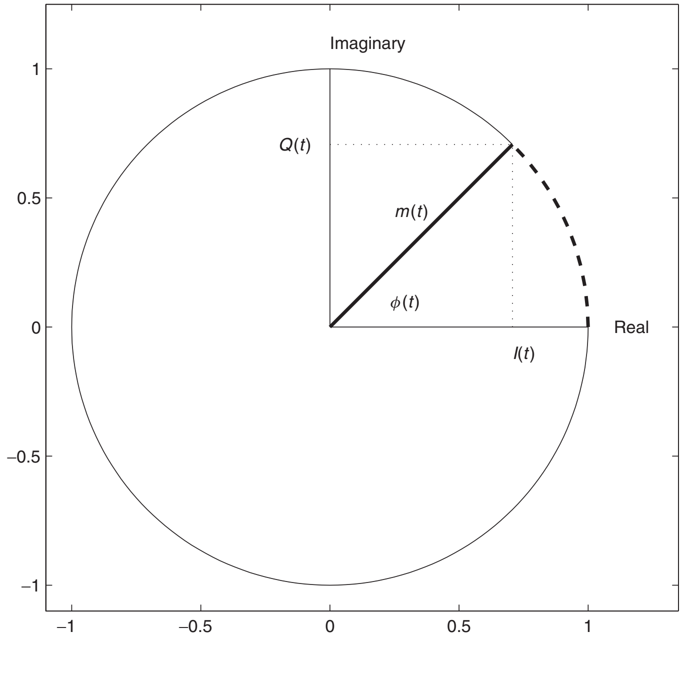

Figure 1.2 Complex signal.

Here is an interesting fact. Euler has originally uncovered the following identity:

$$\cos(2\pi ft) + j\sin(2\pi ft) = e^{j2\pi ft}.$$

The left side of the equality is called the *rectangular form* and the right side is called the *polar form*. Not intuitive at first sight! It is indeed true and a proof of the Euler's identity can be found in the book of Smith (1999) and a tutorial from Lyons (2000). Above all, it is a convenient and compact notation. Figure 1.3 depicts an eloquent representation of a signal in the polar form. The values of $I(t) = \cos(2\pi ft)$ and $Q(t) = \sin(2\pi ft)$ are plotted versus time as a three-dimensional helix. The projection of the helix on a Real-time plane yields the curve of $I(t)$. The projection of the helix on the Imaginary-time plane yields the curve of $Q(t)$. It is equally true that $\cos(2\pi ft) - j\sin(2\pi ft) = e^{-j2\pi ft}$. In this case, however, the vector is rotating clockwise.

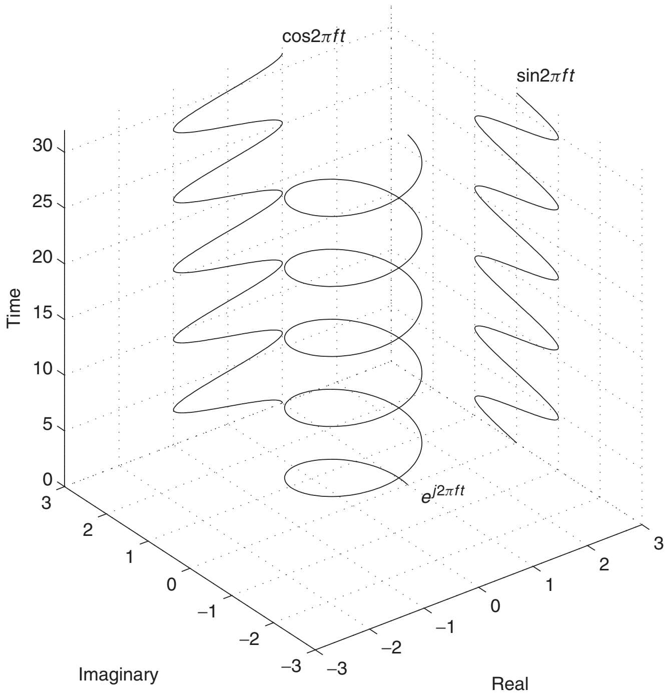

Figure 1.3 Complex signal in 3D.

The beauty of the representation is revealed as follows. Given its $I(t)$ and $Q(t)$, a signal can be demodulated in amplitude, frequency or phase. Its instantaneous amplitude is denoted as $m(t)$ (it is the length of the vector in Figure 1.2) and, according to Pythagoras' theorem, is defined as:

$$m(t) = \sqrt{I(t)^2 + Q(t)^2} \text{ Volts.} \tag{1.1}$$

Its instantaneous phase is denoted as $\phi(t)$ (it is the angle of the vector in Figure 1.2, measured counter clockwise) and is defined as:

$$\phi(t) = \arctan\left(\frac{Q(t)}{I(t)}\right) \text{ Radians.} \tag{1.2}$$

For discrete-time sampled signals, variable t is replaced by variable n in Equations 1.1 and 1.2. On the basis of the observation that the frequency determines the rate of change of the phase, the instantaneous frequency of a discrete-time sampled signal at instant n is derived as follows:

$$f(n) = \frac{f_s}{2\pi}\left[\phi(n) - \phi(n-1)\right] \text{ Hertz.} \tag{1.3}$$

The real domain representation and complex domain representation are dual to each other. It is always possible to map the representation in one model to another, as illustrated in Table 1.1. The column on the left lists real domain representations and their dual representations in the complex domain are listed in the right column. Note that no quantity is added nor subtracted during the translation from one domain to another.

Table 1.1 Equivalence of real and complex representations of signals.

Real domain	Complex domain
$\cos(2\pi ft)$	$\frac{1}{2}\left([\cos(2\pi ft) + j\sin(2\pi ft)] + [\cos(2\pi ft) - j\sin(2\pi ft)]\right) = \frac{1}{2}\left(e^{j2\pi ft} + e^{-j2\pi ft}\right)$
$\sin(2\pi ft)$	$\frac{1}{j2}\left([\cos(2\pi ft) + j\sin(2\pi ft)] - [\cos(2\pi ft) - j\sin(2\pi ft)]\right) = \frac{j}{2}\left(e^{-j2\pi ft} - e^{j2\pi ft}\right)$

1.2 Analog to digital conversion

ADC is the process of translating a continuous-time signal to a discrete-time sampled signal. According to Nyquist, if the bandwidth of a signal is f_{bw} Hertz then a lossless ADC can be achieved at a sampling frequency corresponding to twice f_{bw}. This is called the *Nyquist criterion* and is denoted as:

$$f_s \geq 2f_{bw}.$$

The architecture of an analog to digital converter is pictured in Figure 1.4. On the left side, the input consists of a modulated radio signal. It goes through a low pass filter (LPF) whose role is to limit the bandwidth such that the Nyquist criterion is met. Signal components in the input at frequencies higher than f_{bw} are cut off. Otherwise, false signal images, called *aliases*, are introduced and cause distortion in ADC. On the right side, the output consists of a discrete-time sampled signals. This method is termed *direct digital conversion* (*DDC*).

Direct conversion and sampling, as pictured in Figure 1.4, is limited by the advance of technology and the maximum sampling frequency that can be handled by top of the line processors. Actually to be able to reach the high end of the radio spectrum, an architecture such as the one pictured in Figure 1.5 is used. The modulated radio signal, whose carrier frequency is f_c, is down converted to an intermediate frequency (IF) or the baseband and LPF before ADC. Down conversion is done by mixing (represented by a crossed circle) the frequency of the modulated radio signal with a frequency f_{lo} generated by a local oscillator. If the frequency f_{lo} is equal to f_c, then the IF is 0 Hz and the signal is down converted to baseband. The cut off frequency of the LPF is the upper bound of the bandwidth of the baseband signal: f_{bw}.

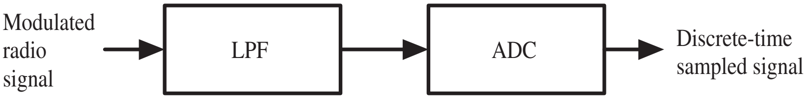

Figure 1.4 Architecture of ADC.

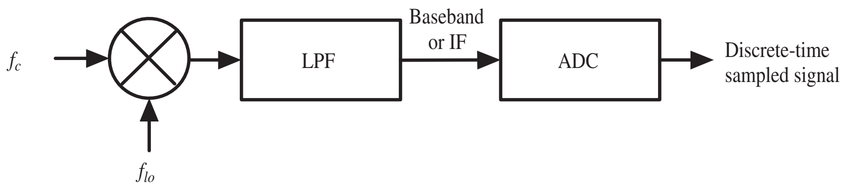

Figure 1.5 Architecture of down conversion and ADC.

When the signals of frequencies f_c and f_{lo} are mixed, their sum $(f_c + f_{lo})$ and differences are generated. The sum is not desired and eliminated by the LPF. One of the differences, that is, $f_c - f_{lo}$, is called the *primary frequency* and is desired because it falls within the baseband. The other one, that is, $-f_c + f_{lo}$, is called the *image frequency*. Note that we can dually choose $-f_c + f_{lo}$ as the primary frequency and $f_c - f_{lo}$ as the image frequency. The image frequency is introducing interference and noise from the lower side band of f_c, that literally *folds over the primary*, as explained momentarily. It is undesirable. The phenomenon is difficult to explain clearly with signals in the real domain representation, but becomes crystal clear when explained with signals in the complex domain. The mixing of two signals at frequencies f_c and f_{lo}, $\cos(2\pi f_c t)$ and $\cos(2\pi f_{lo} t)$ respectively, is defined as the product of their representation in the polar form:

$$\left(\frac{e^{j2\pi f_c t} + e^{-j2\pi f_c t}}{2}\right)\left(\frac{e^{j2\pi f_{lo} t} + e^{-j2\pi f_{lo} t}}{2}\right)$$

which is equal to:

$$\frac{e^{j2\pi(f_c+f_{lo})t} + e^{-j2\pi(f_c+f_{lo})t} + e^{j2\pi(f_c-f_{lo})t} + e^{-j2\pi(f_c-f_{lo})t}}{4}. \tag{1.4}$$

The term $e^{j2\pi(f_c+f_{lo})t} + e^{-j2\pi(f_c+f_{lo})t}$ is a signal in the complex domain whose translation in the real domain, $\cos(2\pi(f_c + f_{lo})t)$, is a signal corresponding to the sum of the two frequencies, cut off by the LPF. Without loss of generality, assume that $f_{lo} < f_c$. In the complex domain representation, the signal at the primary frequency $f_c - f_{lo}$, that is, the signal $e^{j2\pi(f_c-f_{lo})t}$, is a translation of the signal at frequency $f_{lo} + (f_c - f_{lo}) = f_c$. The signal at the image frequency $-(f_c - f_{lo}) = -f_c + f_{lo}$, that is, the signal $e^{-j2\pi(f_c-f_{lo})t}$, is a translation of the signal at frequency $f_{lo} - f_c + f_{lo} = 2f_{lo} - f_c$. Figure 1.6 illustrates the relative locations, on the frequency axis, of the different signals involved. The term $e^{j2\pi(f_c-f_{lo})t} + e^{-j2\pi(f_c-f_{lo})t}$ translated in the real domain representation is $\cos(2\pi(f_c - f_{lo})t)$, the difference of the two frequencies. The image frequency folds over the primary image.

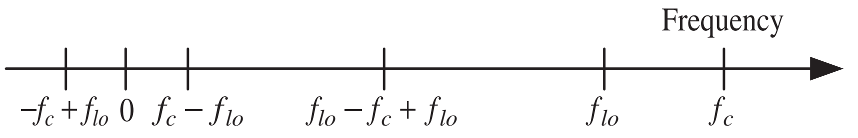

Figure 1.6 Frequencies involved in down conversion.

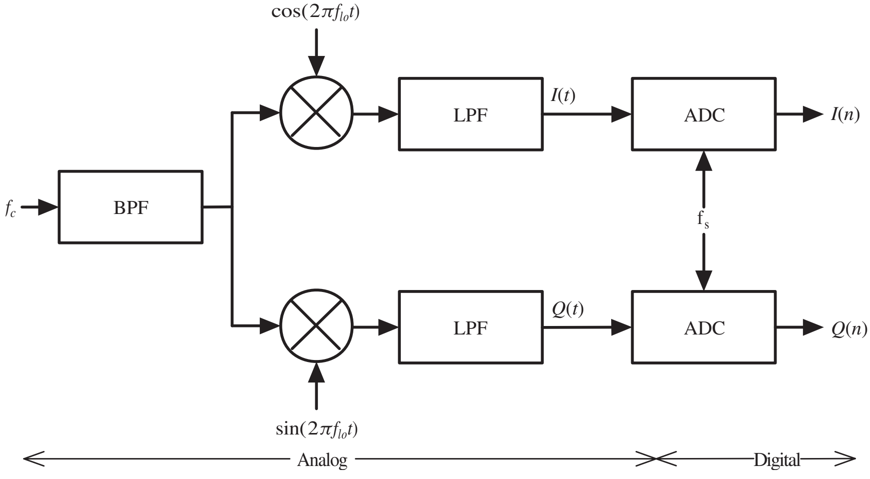

Figure 1.7 Architecture of quadrature mixing.

Image frequencies can be removed by using quadrature mixing, which is depicted in Figure 1.7. From the left side, the radio signal is first band pass filtered (BPF) . There is a low cut off frequency below which signals are strongly attenuated and a high cut off frequency above which signals are strongly attenuated. Signals between the low cut off frequency and high cut off frequency flow as is. The local oscillator consists of both a cosine signal and a sine signal. The top branch mixes the radio signal at frequency f_c with signal $\cos(2\pi f_{lo}t)$ while the bottom branch mixes it with signal $\sin(2\pi f_{lo}t)$. They are both individually low pass filtered to remove the signals at the sum frequencies. The two ADCs are synchronized by the same sampling frequency f_s. Modeled in the complex domain, the output of the top branch is:

$$I(n) = \frac{e^{j2\pi(f_c-f_{lo})n} + e^{-j2\pi(f_c-f_{lo})n}}{4} \tag{1.5}$$

while the output of the bottom branch is:

$$Q(n) = j\frac{e^{j2\pi(f_c-f_{lo})n} - e^{-j2\pi(f_c-f_{lo})n}}{4}. \tag{1.6}$$

Taking the sum,

$$I(n) - jQ(n) = \frac{e^{j2\pi(f_c-f_{lo})n}}{2}$$

cancels the image frequency in the complex domain representation.

Figure 1.8 pictures the signals that flow in quadrature mixing, assuming a baseband or intermediate signal at frequency 1 Hz, that is, $f_c - f_{lo} = 1$ Hertz, defined as $\cos(2\pi t)$. The phase evolves at a rate of 2πrad/s. In the top branch, between the LPF and ADC the signal, in-phase and continuous, is $I(t) = \cos(2\pi t)$. After the ADC, it is a discrete-time signal $I(n) = \cos(2\pi n)$. In the bottom branch, between the LPF and ADC the signal, phase shifted by 90° and continuous, is $Q(t) = \sin(2\pi t)$. After the ADC, it is a discrete-time signal $Q(n) = \sin(2\pi n)$. Note that, with this method, the quadrature of the signal is produced in an analog form, that is, the $Q(t)$, then it is digitized to form the imaginary part of the complex domain representation of a signal being processed. In other words, the quadrature is generated in hardware. An implementation of this method is described by Youngblood (2002a). Smith (2001) describes another method in which the quadrature is calculated after ADC using the Hilbert transform. In other words, the quadrature is generated in software.

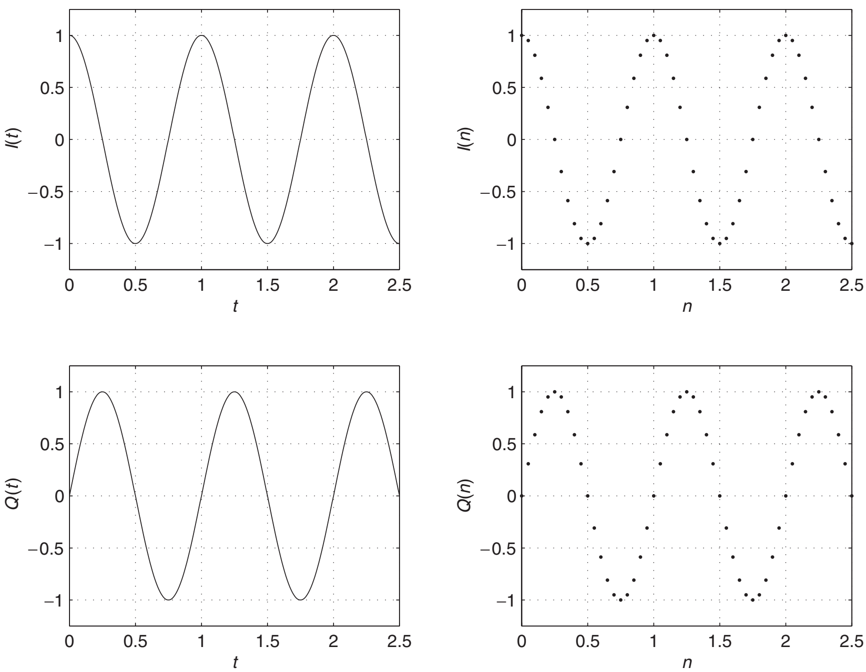

Figure 1.8 Flow of signals in quadrature mixing, with the assumption $f_c - f_{lo}$ Hertz.

1.3 Digital to analog conversion

Digital to analog conversion is an operation that converts a steam of binary values to a continuous-time signal. A digital to analog converter (DAC) converts binary values to voltages by holding the value corresponding to the voltage for the duration of a sample. Figure 1.9 pictures the architecture of a digital to analog conversion. Two signals are involved: a modulating signal and a carrier at frequency f_c. The modulating signal consists

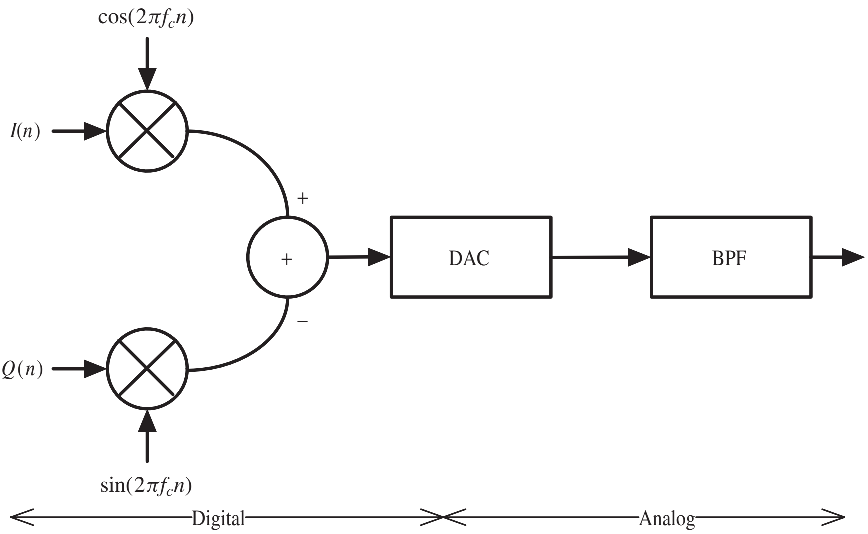

Figure 1.9 Digital to analog conversion.

of an in-phase signal $I(n)$ and its quadrature $Q(n)$. There are two digital mixers, represented by crossed circles. The top mixer computes the product $I(n)\cos(2\pi f_c n)$ while the bottom mixer does the product $Q(n)\sin(2\pi f_c n)$. The result of the first product is added to the value of the second product, with sign inverted. This sum is fed to a DAC then BPF. This method is called *direct digital synthesis* (*DDS*). With this method negative frequencies are eliminated. The DDS computes the real part of the product:

$$e^{j2\pi f_c n}(I(n) + jQ(n))$$

that is:

$$I(n)\cos(2\pi f_c n) - Q(n)\sin(2\pi f_c n)$$

the imaginary part:

$$j\left[I(n)\sin(2\pi f_c n) + Q(n)\cos(2\pi f_c n)\right]$$

does not need to be computed since it is not transmitted explicitly (it is in fact regenerated by the receiver, as discussed in Section 1.2).

1.4 Architecture of an SDR application

The good news is that given the $I(n)$ and $Q(n)$, theoretically, there is nothing that can be demodulated that cannot be demodulated in software. This section reviews the architecture of a software application, an SDR application, that does demodulation (as well as modulation), given a discrete-time signal represented in the complex domain as $I(n)$ and $Q(n)$. The application consists of a capture buffer, a playback buffer and an event handler, in which is embedded DSP.

The capture buffer is fed by the ADCs, in Figure 1.5, and stores the digital samples in the $I(n)$ and $Q(n)$ form. The capture buffer is conceptually circular and of size 2^k samples. Double buffering is used, hence the capture buffer is subdivided into two buffers of size 2^{k-1} samples. While one of the buffers is being filled by the ADCs, the other one is being processed by the SDR application.

The playback buffer is fed by the event handler. It stores the result of digital processing, eventually for digital to analog conversion if, for instance, the baseband signal consists of digitized voice. The playback buffer is also conceptually circular and of size 2^l samples. Quadruple buffering is used, hence the playback buffer is subdivided into four buffers of size 2^{l-2} samples. Among the four buffers, one is being written by the event handler while another is being played by the DAC. Playback starts only when the four buffers are filled. This mitigates the impact of processing time jittering, which arises if the SDR application shares a processor with other processes.

The SDR application is event driven. Whenever one of the capture buffers is filled, an event is generated and an event handler is activated. The event handler demodulates the samples in the buffer using DSP. The result is put in a playback buffer.

The initialization of the SDR application is detailed in Figure 1.10. There are three variables. Variable `i` is initially set to 0 and is the index of the last capture buffer that has been filled by the ADCs and is ready to be processed. The variable `j` is also initially set to 0 and is the index of the playback buffer in which the result of DSP is put when an event is being handled. Variable `first_round` is initially set to true and remains true until the four playback buffers have been filled.

```
Initialization:
   // index over capture buffer
   i = 0
   // index over playback buffer
   j = 0
   // true while in first round of playback buffer filling
   first_round = true
```

Figure 1.10 Initialization of the SDR application.

The event handler of the SDR application is described in Figure 1.11. Firstly, the samples in the current capture buffer are processed and the result is put in the current playback buffer. Then, if the value of the variable `first_round` is true and all four playback buffers are filled and ready, playback is started and variable `first_round` is unasserted. Finally, variable `i` is incremented modulo 2 and variable `j` is incremented modulo 4 for the next event handling instance.

If the sampling frequency is f_s, then the event rate is:

$$r = \frac{f_s}{2^{k-1}} \text{ events/second}$$

Each event should be processed within a delay of:

$$d = \frac{1}{r} \text{ seconds}$$

After initialization, the delay before playback is started is therefore at least $4d$ seconds.

```
Event handling:
   process capture buffer[i]
   put result in playback buffer[j]
   if first_round and j = 3 then
      start playback
      first_round = false
   i = (i + 1) mod 2
   j = (j + 1) mod 4
```

Figure 1.11 Event handler of the SDR application.

1.5 Quadrature modulation and demodulation

The goal of modulation is to represent a bit stream as a radio frequency signal. Table 1.2 reviews the modulation methods of radio systems employed for ad hoc wireless computer networks (note that WiMAX/802.16 does not strictly have an ad hoc mode, but its mesh mode has ad hoc features). Bluetooth uses Gaussian-shape frequency shift keying (GFSK) combined with frequency hopping (FH) spread spectrum (SS) transmission (Bluetooth, SIG, Inc., (2001a)). WiFi/802.11 uses two or four-level GFSK, for the data rates 1 Mbps and 2 Mbps respectively, combined with FH SS transmission (IEEE (1999a)). WiFi/802.11 also uses differential binary phase shift keying (DBPSK) and differential quadrature phase shift keying (DQPSK), for the data rates 1 Mbps and 2 Mbps respectively, combined with direct sequence (DS) SS. WiFi/802.11b uses complementary code keying (CCK) (IEEE (1999c)). WiFi/802.11a uses orthogonal frequency division multiplexing (OFDM) (IEEE (1999b)). WiMAX/802.16, with the single-carrier (SC) transmission and 25 MHz channel profile, uses quadrature phase shift keying (QPSK) (downlink or uplink) and quadrature amplitude modulation-16 states (QAM-16) (downlink only) (see IEEE et al. (2004)). WiMAX/802.16, with the OFDM transmission and 7 MHz channel profile, uses QAM-64 states (note that WiMAX/802.16 defines other transmission and modulation characteristics).

Table 1.2 Modulation schemes.

System	Bandwidth (MHz)	Modulation	Rate (Mbps)	Transmission
Bluetooth	1	GFSK	1	FH SS
802.11	1	GFSK	1 and 2	FH SS
	10	DBPSK	1	DS SS
	10	DQPSK	2	DS SS
802.11b	10	CCK	11	CCK
802.11a	16.6	OFDM	54	OFDM
802.16 SC-25	25	QPSK	40	SC
802.16 SC-25	25	QAM-16	60	SC
802.16 OFMD-7	7	QAM-64	120	OFDM

Frequency shift keying (FSK) and Phase shift keying (PSK) modulation are discussed in more detail in the sequel. SS transmission is presented in Section 1.6.

The frequency shifts according to the binary values are tabulated for WiFi/802.11 GFSK in Tables 1.3 and 1.4.

Table 1.3 Two-level GFSK modulation.

Symbol	Frequency shift (kHz)
0	−160
1	+160

Table 1.4 Four-level GFSK modulation.

Symbol	Frequency shift (kHz)
00	−216
01	−72
10	+216
11	+72

The phase shifts according to the binary values are tabulated for WiFi/802.11 DBPSK and DQPSK in Tables 1.5 and 1.6. For DBPSK, the phase of the carrier is shifted by 0°, for binary value 0, or 180°, for binary value 1.

Table 1.5 DBPSK modulation.

Symbol	Phase shift (°)
0	none
1	180

Generation of a discrete-time sampled FSK signal, for playback, can be done as follows. Let f_o be the frequency of a symbol to represent and T_b be the duration of a symbol. The modulating signal (used in Figure 1.9) for the period T_b is:

$$e^{j2\pi f_o n}.$$

Note that $I(n) = \cos(2\pi f_o n)$ and $Q(n) = \sin(2\pi f_o n)$. This modulating signal shifts the instantaneous frequency of the carrier f_c proportionally to the value of f_o:

$$e^{j2\pi f_c n} e^{j2\pi f_o n} = e^{j2\pi (f_c + f_o) n}.$$

As mentioned in Section 1.3, only the real part needs to be computed and transmitted.

Table 1.6 DQPSK modulation.

Symbol	Phase shift (°)
00	none
01	90
10	–90
11	180

Production of a PSK signal can be done as follows. Let ϕ be the phase of a symbol and T_b its duration. The modulating signal (usable in Figure 1.9) for the duration T_b is:

$$e^{j\phi}.$$

This modulating signal shifts the instantaneous phase of the carrier f_c proportionally to the value of ϕ:

$$e^{j2\pi f_c n} e^{j\phi} = e^{j(2\pi f_c n + \phi)}.$$

This is termed *exponential modulation*. The algorithm of a software exponential modulator is given in Figure 1.12. The bits being transmitted are given in an output buffer. The samples of the modulation, that is, the modulated carrier, are stored by the modulator in the playback buffer. Let r be the bit rate, the number of samples per bit corresponds to:

$$s = \frac{f_s}{r}.$$

The length of the playback buffer corresponds to the length of the output buffer times s. A for loop, that has as many instances as the length of the playback buffer, generates the samples. The sample at position i encodes the bit from output buffer at index $\lfloor \frac{i}{s} \rfloor$. The index of the first bit is 0 in both the output buffer and playback buffer. The function `fo()` returns the value of the frequency shift as a function of the value of the symbol. The time of the first sample is 0 and incremented by the value $\frac{1}{f_s}$ from sample to sample. The symbol `fc` represents the frequency of the carrier. The term `exp(j*x)` corresponds to the expression e^{jx}.

```
for i = 0 to length of playback buffer, minus one
   // determine value of symbol being transmitted
   symbol = output buffer[floor(i / s)]
   // determine the frequency shift
   shift = fo(symbol)
   // determine time
   n = i * 1/fs
   // Generate sample at position "i"
   playback buffer[i] =
      real part of exp(j*2*pi*(fc+shift)*n)
```

Figure 1.12 Algorithm of a software exponential modulator.

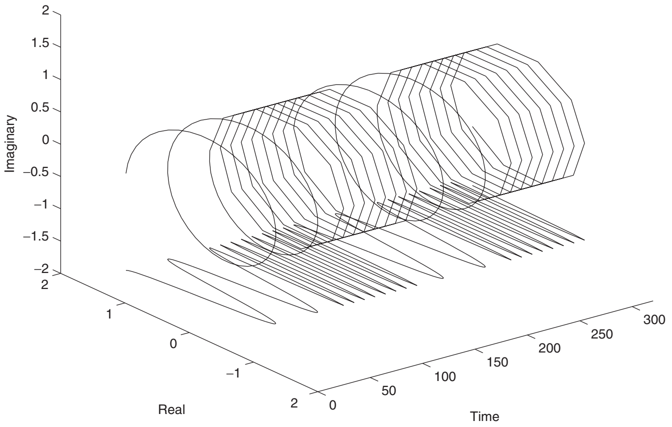

Figure 1.13 Exponential modulation of bits `1 0 1 0`.

An example modulation is plotted as a three-dimensional helix in Figure 1.13. The picture also contains a projection of the helix on a real-time plane, which is the signal effectively transmitted. For this example, the frequency of the carrier is 5 Hz, the frequencies of the modulating signal are +3 Hertz for bit 0 and −3 for bit 1, the rate is 1 bps and the sampling frequency is 80 samples/s. The picture corresponds to the modulation of bits `1 0 1 0`.

The pseudo code of a FSK demodulator is given in Figure 1.14. The algorithm consists of a for loop that has as many instances as there are samples in the capture buffer, parsed into the `Quadrature` buffer and `InPhase` buffer. Given an instance of the loop, the frequency of the previous sample is available in `prev_f` and phase of previous sample in `prev_p`. While a carrier is being detected and frequency of the carrier is invariant, the variable `count` counts the number of samples demodulated. The variable `count` is initialized to 0. When the variable `count` reaches the value s, a complete symbol has been received and its value is determined according to the frequency shift of the carrier. Then, the variable `count` is reset.

1.6 Spread spectrum

There are two forms of SS transmission in use: direct sequence (DS) and frequency hopping (FH).

With DS SS, before transmission the *exclusive or* of each data bit and a pseudorandom binary sequence is taken. The result is used to shift the carrier. DBPSK or DQPSK may be used for that purpose. The length of the resulting bit stream is the length of the original bit stream multiplied by a factor corresponding to the length of the pseudorandom binary sequence. This applies as well to the data rate and bandwidth occupied by the radio signal: the signal is spread over a larger bandwidth.

```
// Initialization
prev_f = 0
prev_p = 0
count = 0
// Demodulation loop
for i = 0 to length of capture buffer, minus one
   // Compute the instantaneous phase
   phase = atangent Quadrature(i) / InPhase(i)
   // Compute the instantaneous frequency
   freq = fs * ((phase - prev_p) / (2 * pi) )
   // Detection of carrier
   if freq == (fc + fo(1)) or freq == (fc + fo(2))
      if (count==0)
         // no bit is being demodulated, start demodulation
         count = 1
      else if freq==prev_f
         // continue demodulation while frequency is constant
         count = count + 1
      else
         count = 0
   // determine if a full bit has been demodulated
   if count==s
      if freq==fc+fo(1)
         symbol = 0
      else
         symbol = 1
      count = 0
   // save phase and frequency for the next loop instance
   prev_p = phase
   prev_f = freq
end
```

Figure 1.14 Algorithm of a software demodulator.

In data communications, the pseudorandom binary sequence is often of fixed length, the same from one data bit to another and shared by all the transmitters and receivers communicating using the same channel.

The 11-bit Baker pseudorandom binary sequence is very popular:

$$10110111000$$

Application of the Baker sequence to a sequence of data bits is illustrated in Figure 1.15.

The popularity of the Baker sequence is due to its self-synchronization ability. That is to say, bit frontiers can be determined. Indeed, as the bits that were transmitted are received they pass through an 11-bit window. The autocorrelation of the bits in the 11-bit window and bits of the 11-bit Baker sequence is calculated. The autocorrelation is the value of a counter, initialized to 0. From left to right and i from 1–11, if the bit

Data bits	0	1	0	0	0
Transmitted sequence	10110111000	01001000111	10110111000	10110111000	10110111000

Figure 1.15 Application of the Baker sequence.

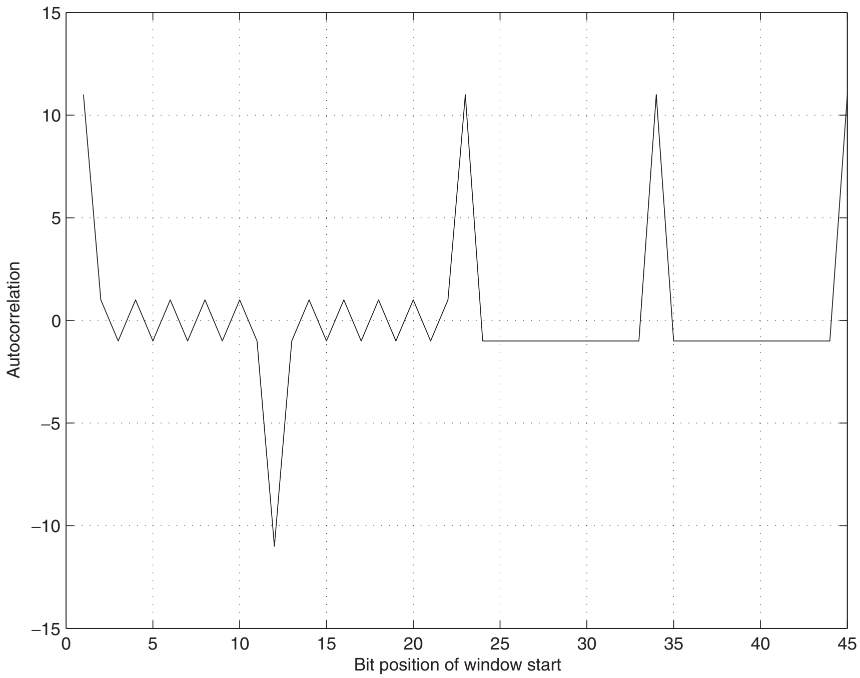

Figure 1.16 Autocorrelation with Baker sequence.

at position i in the window matches the bit at the same position in the Baker sequence, then the counter is incremented. Otherwise it is decremented. In the absence of errors, the count varies from minus 11 (lowest autocorrelation) to 11 (maximum autocorrelation). Maximum autocorrelation marks the starting positions of bits with value 0. Figure 1.16 plots the autocorrelation of the bit sequence of Figure 1.15 when the 11-bit window slides from left to right.

The 802.11 radio system uses DS SS (IEEE (1999a)). The carrier is modulated using DBPSK and DQPSK at respectively 1 Mbps and 2 Mbps. The 11-chip Baker pseudorandom binary sequence is used.

DS SS offers resilience to jamming and higher capacity, given a power of signal to power of noise ratio. An analysis by Costas (1959) shows that it is more difficult to jam a broad channel because the required power is directly proportional to the bandwidth.

Shannon (1949) has developed a model giving the capacity of a channel in the presence of noise:

$$C = W \log_2 \left(1 + \frac{S}{N}\right)$$

where C is the capacity in bps, W the bandwidth in Hertz, S the power of the signal in Watts and N the power of the noise in Watts. If S/N cannot be changed, then higher capacity can be achieved if the signal is using a broader bandwidth as in DS SS.

With FH SS, a radio frequency band is divided into segments of the same bandwidth. Each segment is characterized by its centre frequency. The transmitter jumps from one frequency to another according to a predetermined hopping pattern. The transmitter sits on each frequency for a duration called the *dwell time*. The receiver(s) synchronizes with the transmitter and jumps from one frequency to another in an identical manner. FSK is used to send data during dwell time.

The Bluetooth radio system uses FH SS (Bluetooth, SIG, Inc., (2001a)). In North America, the number of frequencies is 79 and they are defined as follows, for $i = 0 \ldots 78$:

$$f_i = 2402 + i \text{ Mega Hertz.} \tag{1.7}$$

Each segment has a bandwidth of 1 MHz. The carrier is modulated using GFSK. The hopping rate is 1600 hops/s. This is termed *slow FH* because the data rate is higher than the hopping rate. A short frame can be sent during one dwell time.

The WiFi/802.11 radio system uses FH SS as well (IEEE (1999a)). Three sets of hopping patterns are used, hence defining three logical channels. 802.11 uses 79 frequencies defined as in Equation 1.7. The carriers are also modulated using GFSK. At 2.5 hops/s, 802.11 is also slow FH.

FH offers resilience to narrow band interference. If a source of interference is limited to one frequency, the frequency can be retracted from the hopping pattern.

1.7 Antenna

Antennas are to wireless transmission systems what speakers are to sound systems. A sound system is not better than its speakers. A wireless transmission system is not better than its antennas. Antennas are devices that radiate and pick up electromagnetic power from free space. There are directional antennas and omni-directional antennas. *Directional antennas* radiate a focused electromagnetic power beam and pick up a focused source of energy. *Omni-directional antennas* spread and pick up electromagnetic power in all directions. This section discusses in a more formal manner the directivity and the maximum separation distance between two antennas.

The directivity of an antenna is captured mathematically using the notion of decibel (dB) and dB isotropically (dBi). The dBs translate the magnitude of a ratio of the quantities x and y as:

$$z\ dB = 10 \log_{10} \frac{x}{y}.$$

Directivity means that an antenna radiates more power in certain directions and less in others. An antenna that would radiate power uniformly in all directions is called an *isotropic* *antenna*. Such an antenna cannot be physically realized, but serves as a reference to characterize designs of real antennas.

Given an antenna and the power P_1 it radiates in the direction of strongest signal, a value in dBi translates the magnitude of the ratio of this power over the power P_2 radiated in a direction by an isotropic antenna in the same conditions. This ratio,

$$g\ dBi = 10 \log_{10} \frac{P_1}{P_2}$$

is called the *gain* of the antenna.

The *radiation pattern* defines the spatial distribution of the power radiated by an antenna. A radiation pattern can be graphically represented by a three-dimensional surface of equal power around the antenna. Figure 1.17 represents the radiation pattern of an omni-directional antenna. It is an antenna of type ground plane with a gain of 1.9 dBi. Figure 1.18 shows the radiation pattern of a directional antenna. It is an antenna of type Yagi with a gain of 12 dBi.

For terrestrial microwave frequencies (1 GHz–40 GHz), antennas shaped as parabolic dishes are commonly used. They are used for fixed, focused, line-of-sight transmissions.

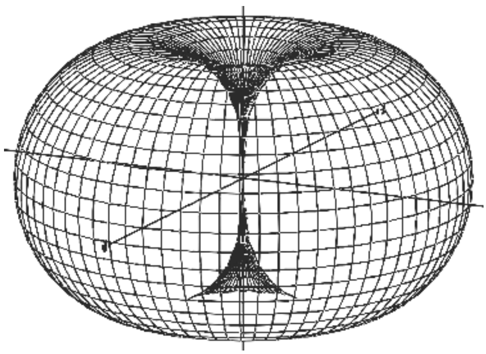

Figure 1.17 Radiation pattern of an omni-directional antenna.

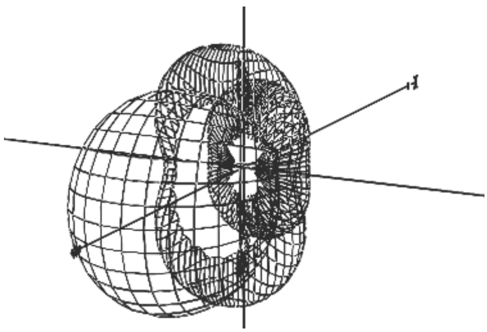

Figure 1.18 Radiation pattern of a directional antenna.

Line-of-sight transmission is achievable whenever two antennas can be connected by an imaginary non-obstructed straight line.

Let h be the height, in meters, of two antennas. The maximum distance between the antennas with line-of-sight transmission, in kilometers, is computed by the following formula:

$$d = 7.14\sqrt{Kh}.$$

K is an adjustment factor modeling refraction of waves with the curvature of the earth (hence waves propagate farther), typically $K = 4/3$. Maximum distance between antennas at heights up to 300 m is plotted in Figure 1.19.

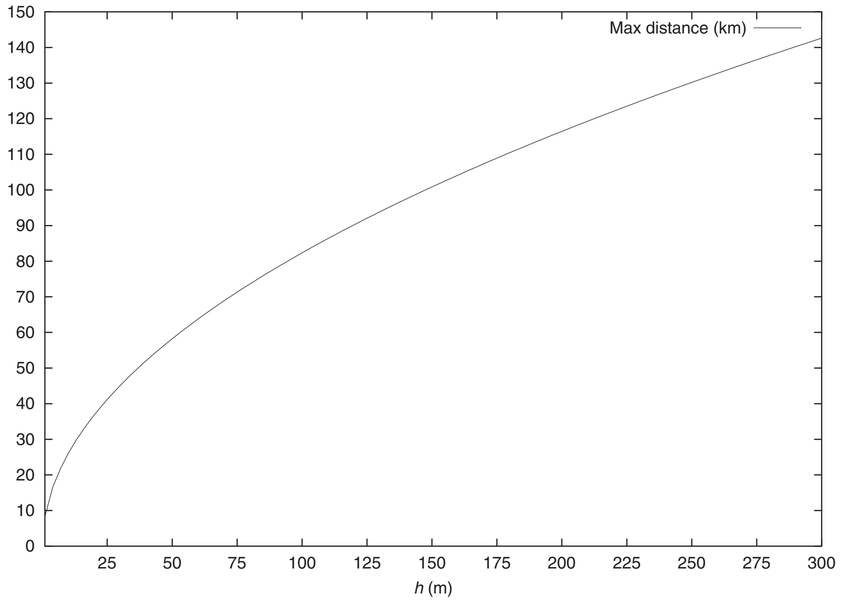

Figure 1.19 Maximum distance between antennas.

1.8 Propagation

Successful propagation in free space of a radio frequency signal from a transmitter to a receiver is subject to a number of parameters, namely, the:

1. output power of the transmitter (P_T),
2. attenuation of the cable connecting the transmitter and transmitting antenna (C_T),
3. transmitting antenna gain (A_T),
4. free space loss (L),
5. receiving antenna gain (A_R),

6. attenuation of the cable connecting the receiving antenna and receiver (C_R) and

7. receiver sensitivity (S_R).

Antenna gain is discussed in Section 1.7. The remaining six parameters are discussed in the following text.

In the gigahertz range, the output power of transmitters is often weak, in the order of milli Watts (mW) and conveniently expressed in decibel milli Watts (dBm). Given a power level P in mW, the corresponding value in dBm is defined as:

$$Q\ dBm = 10 \log_{10} PmW$$

Tables 1.7 and 1.8 gives the power level requirement of various Bluetooth, WiFi/802.11 and WiFi/802.16 radios. The data was extracted from product standards. For instance, the output power of an 802.11 radio is 100 mW or 20 $dBm = 10 \log_{10} 100\ mW$.

The sensitivity of a receiver is also expressed in dBm as a function of two performance parameters: data rate and error rate. The rate of error is given as a Bit Error Rate (BER), Frame Error Rate (FER) or Packet Error Rate (PER). Frames are assumed to be of length 1024 bytes while packets are assumed to be of length 1000 bytes. For instance, according to Table 1.8 a 802.11 radio should achieve a data rate of 1 Mbps with an expected FER of 3% if the strength of the signal at the receiver is −80 dBm or higher. If the hardware parameters are fixed, then the table also indicates that the data rate can be increased solely if the strength of the signal at the receiver can be increased. Note that the specifications of products from manufacturers often exhibit better numbers than the requirements. It often difficult to compare products from different manufacturers because the performance figures are often given under different constrains. that is, different BER, FER or PER. There is no

Table 1.7 Transmission performance parameters of 802.11 and Bluetooth radios.

Radio	Frequency (GHz)	Power (dBM)
Bluetooth Class 1	2.4–2.4835	20
Bluetooth Class 2		4
Bluetooth Class 3		0
801.11	2.4–2.4835	20
801.11b	2.4–2.4835	20
801.11a	5.15–5.35	16–29
802.16 SC-25 QPSK	10–66	≥ 15
802.16 SC-25 QPSK	10–66	≥ 15
802.16 SC-25 QAM-16	10–66	≥ 15
802.16 SC-25 QAM-16	10–66	≥ 15
802.16 OFMD-7	2–11	15–23

Table 1.8 Reception performance parameters of 802.11 and Bluetooth radios.

Radio	Rate (Mbps)	Error	Sensitivity (dBm)
Bluetooth Class 1	1	10^{-3} (BER)	−70
Bluetooth Class 2	1	10^{-3} (BER)	−70
Bluetooth Class 3	1	10^{-3} (BER)	−70
801.11	1	3 % (FER)	−80
	2	3 % (FER)	−75
801.11b	11	8 % (FER)	−83
801.11a	54	10 % (PER)	−65
802.16 SC-25 QPSK	40	10^{-3} (BER)	−80
802.16 SC-25 QPSK	40	10^{-6} (BER)	−76
802.16 SC-25 QAM-16	60	10^{-3} (BER)	−73
802.16 SC-25 QAM-16	60	10^{-6} (BER)	−67
802.16 OFMD-7	120	10^{-6} (BER)	−78 to −70

standard reference. Independent evaluations, when available, might be the best source for performance comparisons.

The strength of a signal, that is, its attenuation, decreases with distance. In cables or free space, what is important is the relative reduction of strength. It is normally expressed as a number of decibels (dB). A value in dB expressing a ratio of two powers P_1 and P_2 is given by the following formula:

$$10 \log_{10} \frac{P_2}{P_1}.$$

If P_1 expresses the power of a transmitted signal and P_2 the power of the signal at the receiver, which is normally lower, then a loss corresponds to a negative value in dB. Through an amplifier, P_1 may express the power of an input signal and P_2 the power of the signal at the output, which is normally higher, then the gain is expressed by a positive value in decibels.

The total gain or loss of a transmission system consisting of a sequence of interconnected transmission media and amplifiers can be computed by summing up the individual gain or loss of every element of the sequence. Equivalently, it could be expressed by a ratio obtained by multiplying together the gain or loss ratio of every element. Doing calculations in dB is much more convenient because additions are more convenient to calculate than multiplication.

If the frequency of the radio signal is fixed, then the cable attenuation, in dB, is a linear function of distance. Attenuation, per unit of length, can be extracted from manufacturer specifications (Table 1.9). For instance, a radio signal at frequency 2.5 GHz is subject to an attenuation of 4.4 dB per 100 feet over the LMR 600 cable.

Table 1.9 Cable attenuation per 100 feet.

Type	Frequency (GHz)	Attenuation (dB)
Belden 9913	0.4	2.6
	2.5	7.3
	4	9.5
LMR 600	0.4	1.6
	2.5	4.4
	4	5.8
	5	6.6

On the other hand, free space loss, in dB, is proportional to the square of distance. If d is the distance and λ the wavelength, both expressed in the same units, the loss is given by the following formula:

$$L = 10\log_{10}\left(\frac{4\pi d}{\lambda}\right)^2 \ dB. \tag{1.8}$$

Other factors are not taken into account. For instance, rainfall increases attenuation, in particular at frequency 10 GHz and above.

Figure 1.20 compares attenuation presented to a 1 MHz signal, over a wireless medium, with a Category 5 cable, where loss in dB is proportional to a factor times the distance. Loss over a wireless medium is significantly higher.

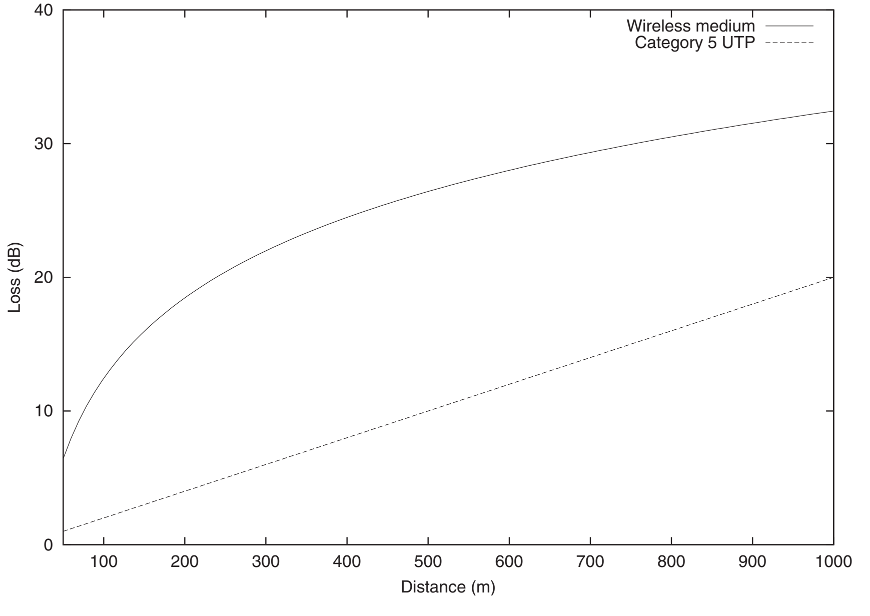

Figure 1.20 Comparison of attenuation of a 1 MHz signal over a wireless medium and a Category 5 cable.

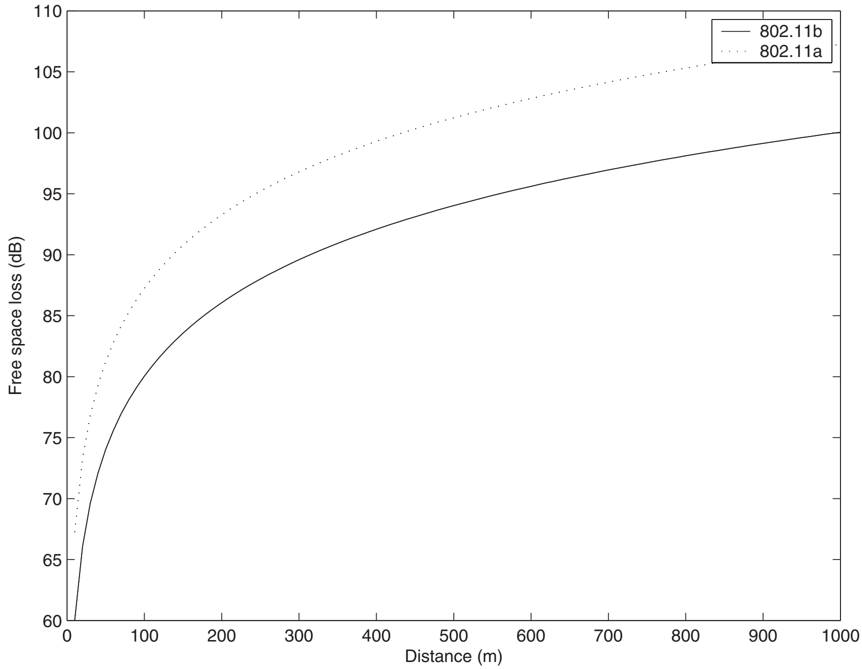

Figure 1.21 Comparison of attenuation of 802.11a and 802.11b.

Figure 1.21 compares free space loss for 802.11a (5.5 GHz) and 802.11b (2.4 GHz) as a function of distance. At equal distance, 802.11a has an additional cost of 7.2 dB in free space loss.

One can observe that, in contrast to cable medium, free space loss becomes quickly high and stays substantially higher. What is the consequence of that? Higher loss means that the signal at the receiver is weaker relative to noise. It means that the probability of bit errors is higher. For the sake of comparison, Table 1.10 gives BERs typically obtained with wireless, copper and fiber media. The relatively higher BERs over a wireless medium cause problems when a protocol such as Transmission ControlProtocol (TCP), devised for wired networks, is used for wireless communications. This has to with the way TCP handles network congestion. TCP interprets delayed acknowledgements of transmitted data segments as situations of congestion. In such cases, TCP decreases its throughput to relieve the

Table 1.10 Typical BERs as a function of the medium type.

Medium	BER
Wireless	10^{-6}–10^{-3}
Copper	10^{-7}–10^{-6}
Fiber	10^{-14}–10^{-12}

network by increasing the value of its retransmission timer, which means that it waits longer before retransmitting unacknowledged segments of data. On a wireless medium, however, absence of acknowledgements is mainly because of the high BER. Hence, reduction of this throughput is useless since it does not improve the BER and it delays delivery of data segments.

The performance of transmitting system (i.e. a transmitter, a cable and an antenna) is termed the *effective isotropic radiated power* (*EIRP*) and is defined using the following equation:

$$EIRP = P_T - C_T + A_T.$$

Assuming line of sight between a transmitting antenna and a receiving antenna and if directional, orientation towards each other, the maximum free space loss, given a data rate and a BER can be calculated as:

$$L = EIRP + A_R - C_R - S_R.$$

The maximum coverage of a system can be calculated by resolving the variable d which appears in Equation 1.8.

The performance (i.e. the data rate and BER) of a radio system can be improved either by:

- increasing the power of the transmitter,
- lowering the sensitivity of the receiver,
- increasing the gain of the transmitting antenna or receiving antenna,
- shortening lengths of cables or
- shortening the distance.

This model of performance is consistent with models used by manufacturers of wireless interfaces, for instance Breeze, Wireless Communications Ltd. (1999). Other factors do affect the performance of radio systems and limit their coverage. Some of these are multipath reception, obstruction of line of sight, rain, antenna motion due to wind, noise, interference and hardware or software faults. McLarnon (1997) reviews some of them in more detail in the context of microwave communications.

1.9 Ultrawideband

A transmission system is said to be an ultrawideband (UWB) if the bandwidth of the signal it generates is much larger than the center frequency of the signal. Since, according to Shannon's equation, the capacity of a channel is proportional to its bandwidth, UWB supports relatively much larger data rates. It is targeted for low power (1 mW or less) and short distance operation (i.e. few meters).

Parameters of a typical UWB system are given in Table 1.11 (Stroh (2003)). In UWB, the carrier consists of a stream of very short pulses, from 10–1000 ps each. By nature, pulsative signals occupy a very large amount of bandwidth. Neither up or down frequency conversion of signals is required.

Table 1.11 Parameters of an UWB system.

Bandwidth	500 MHz
Frequency range	3.1 GHz to 10.6 GHz
Data rate	100 Mbps to 500 Mbps
Range	10 m
Transmission power	1 mW

Table 1.12 Shape of modulated UWB pulses.

Modulation	1	0
Amplitude	Full	Half
Bipolar	Positive	Inverted
Position	Non delayed	Delayed

Modulation is done by changing the amplitude, direction or spacing of the pulses. Possible UWB modulation schemes are described in Table 1.12 (Leeper (2002)). An example is pictured in Figure 1.22. Binary values are represented in part (a). Part (b) illustrates pulse amplitude modulation. A full amplitude pulse represents the binary value 1 while a half amplitude pulse represents the binary value 0. With pulse bipolar modulation (c), a positive

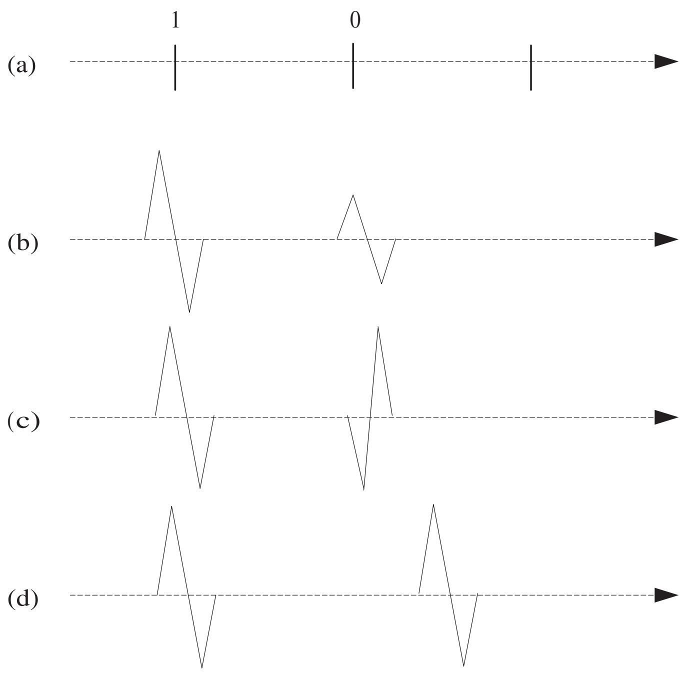

Figure 1.22 UWB modulation.

rising pulse represents the binary value 1. The binary value 0 is coded as an inverted falling pulse. Pulse position modulation (d) encodes the binary value 1 as a non-delayed pulse, whereas the pulse for the binary value 0 is delayed in time.

In few words, UWB is a very short range high data rate wireless data communications system. The signal being very large bandwidth, UWB has a natural resilience to noise. On the other hand, UWB may create interference to other systems.

1.10 Energy management

Energy management is an issue in ad hoc networks because devices are battery powered with limited capacity. Energy refers to the available capacity of a device for doing work, such as computing, listening, receiving, sleeping or transmitting. The amount of work is measured in Joules. The available capacity for doing work, the energy, is also measured in joules. The rate at which work is done is measured in Joules per second, or Watts.

The energy consumed by a wireless interface in the idle, receive, sleep and transmission modes has been studied by Feeney and Nilsson (2001). Table 1.13 lists representative energy consumption values. There is an important potential of saving energy when a device is put in sleep mode. This is an aspect that is studied in more detail in Section 3.4.4.

Table 1.13 Energy consumption.

State	Consumption (mW)
Idle	890
Receive	1020
Transmit	1400
Sleep	70

1.11 Exercises

1. Explain the math between the following phenomenon. In North America, the audio of the TV channel 13 is transmitted at frequency 215.75 MHz. A receiver that has the capability to handle signals within 216–240 MHz and an IF of 10.8 MHz, can actually receive the audio of the channel 13 at frequency 237.35 MHz.

2. Demonstrate that Equations 1.5 and 1.6 hold.

3. Given a sampling frequency f_s equal to 44,100 samples/s and a capture buffer of size 2^{12} samples, calculate the event rate, event processing delay and delay before start of playback.

4. Given a radio system with a maximum free space loss L operating at wavelength λ, develop an expression representing the maximum separation distance d (in meters) between a transmitter and receiver.

5. Compute the maximum coverage for each type of radio listed in Tables 1.7 and 1.8.

6. Given a distance d in Km, demonstrate that at 2.4 GHz, the expression of free space loss can be simplified as $100 + 20\log_{10} d$.