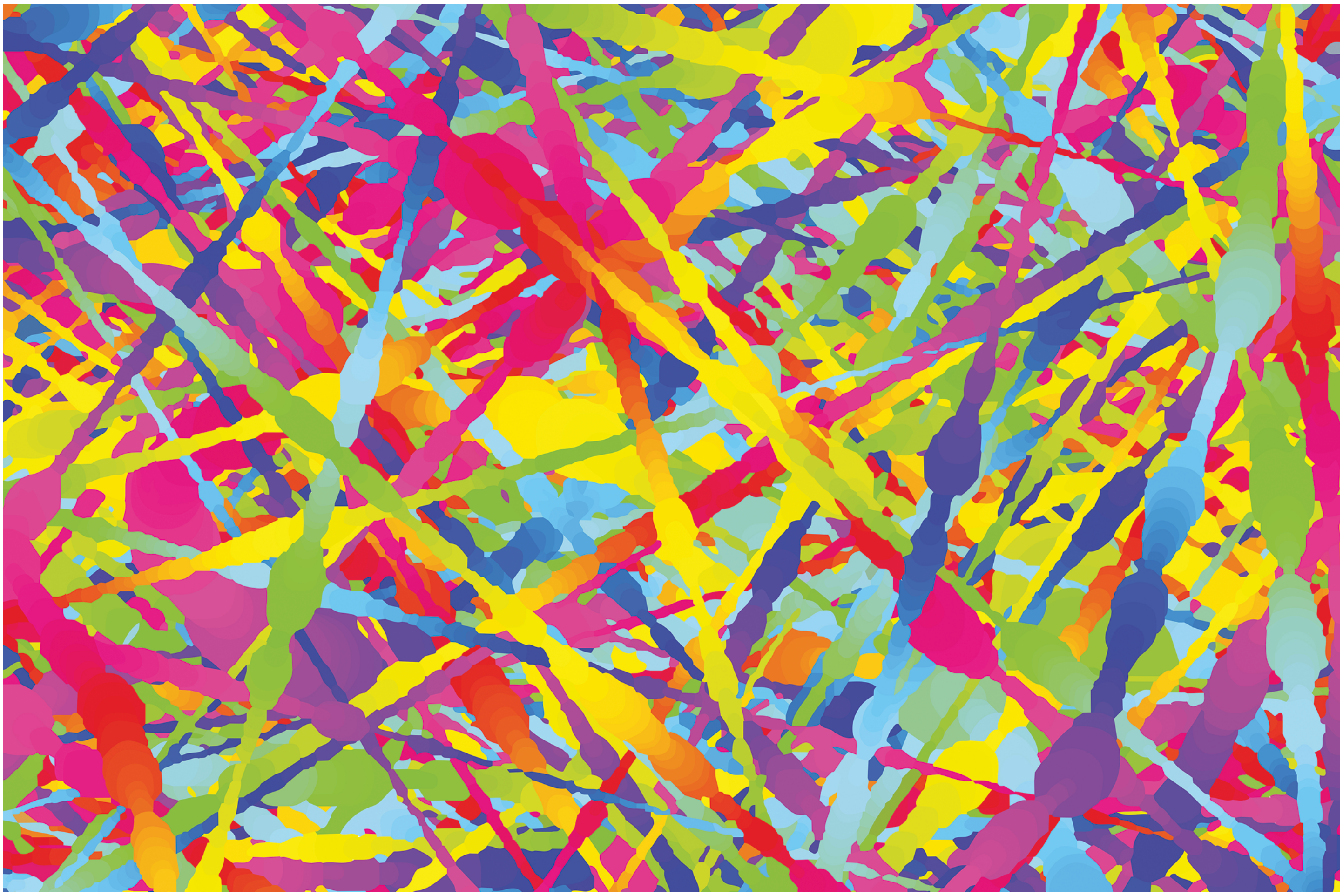
# Exploring communications technology



**About this free course**

This free course is an adapted extract from the Open University course TM355 *Communications technology* www.openuniversity.edu/courses/modules/tm355.

This version of the content may include video, images and interactive content that may not be optimised for your device.

You can experience this free course as it was originally designed on OpenLearn, the home of free learning from The Open University – http://www.open.edu/openlearn/science-maths-technology/exploring-communications-technology/content-section-0

There you'll also be able to track your progress via your activity record, which you can use to demonstrate your learning.

# Contents

# Introduction

Modern communication technology amalgamates many areas of knowledge such as electronics, radio-frequency engineering, information theory, cryptography, and signal processing. Nevertheless, some basic principles and recurring themes underpin much communications technology, and this free course, *Exploring communications technology*, concentrates on them. The first section focuses on digital modulation and on the widely used technique of Quadrature Amplitude Modulation (QAM). The second section looks at error control, and the third at data compression. The fourth and final section looks at the way orthogonal frequency division multiplexing (OFDM) underpins fourth generation mobile communications (4G), wi-fi and broadband.

This OpenLearn course is an adapted extract from the Open University course TM355 *Communications technology*.

# Learning Outcomes

After studying this course, you should be able to:

- demonstrate an understanding of the principles of, and reasons for, carrier-wave modulation, and its application to Quadrature Amplitude Modulation
- demonstrate an understanding of the roles of error detection and error correction, and be able to perform calculations involving checkeck digits and coding rates
- demonstrate an understanding of the basic techniques of perceptual coding and the reasons for their use
- demonstrate an understanding of how the use of multiple OFDM subchannels enables efficient use of communications channels in mobile communications, broadband and wifi.

# Communications

## 1 Signals and modulation

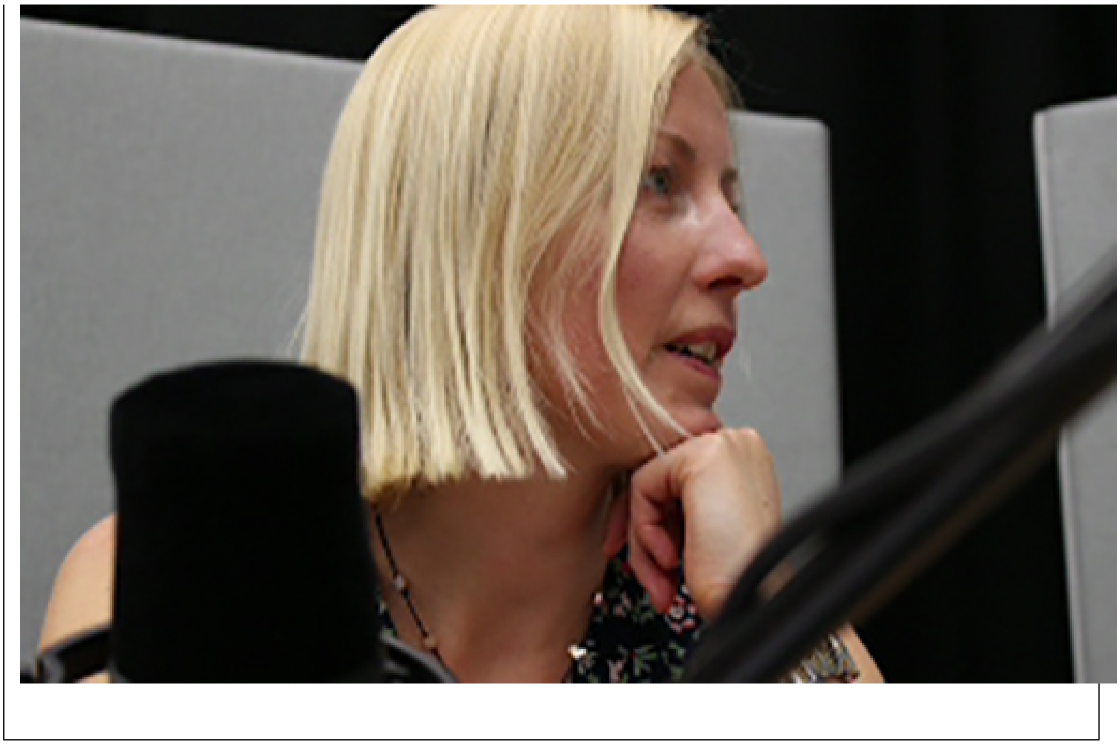
Modern communication, whether by smart phone, computer networks, or broadcast TV and radio, presents many challenges.

- There is generally a demand for faster broadband, faster mobile data, higher definition TV and video, etc. In the terminology of the subject, there is a demand for higher capacity links.
- Users of a shared medium (such as mobile communications) must somehow be kept separate from each other.
- Errors in transmission, which are unavoidable, should be minimised.
- Coverage can be limited, particularly with mobile devices.

Solutions to these problems can conflict with each other. For example, increasing the power of a broadcast transmitter improves coverage and reduces the possibility of errors, but might adversely affect other services in the area. In practice, therefore, compromises usually have to be found, which is typical of the engineering approach to problem-solving. This OpenLearn course looks at some of the theoretical background to the technology of modern communications. It is an adapted extract from the Open University course TM355, *Communications Technology*.

In this introductory audio, Allan Jones talks to Adrian Poulton and Helen Donelan, the authors of Section 1, about issues related to this topic.

Video content is not available in this format.



## 1.1 Periodic signals

Fundamental to communications is the analogue signal known as the **sinusoid** or sine wave, shown in Figure 1.1. Sinusoids are important not only because they turn up naturally in a wide variety of situations, but also for their mathematical simplicity. Figure 1.1 shows a sinusoidal changing voltage, but other properties can change in this way, such as current, power, pressure, and so on.

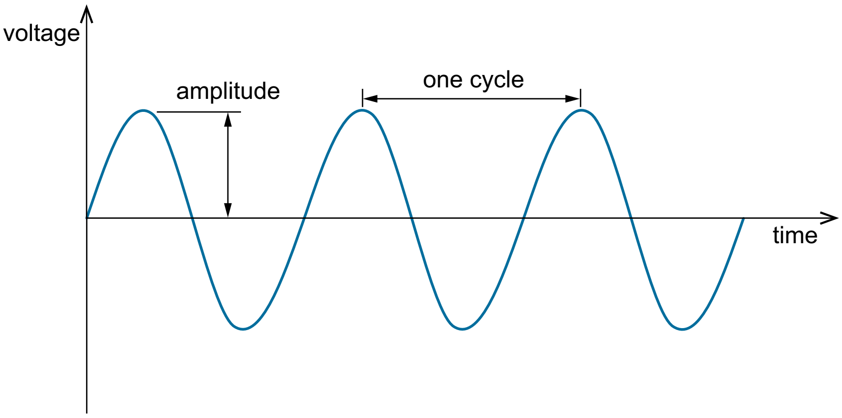



**Figure 1.1 A sinusoidal signal**

A sinusoid is an example of a **periodic signal**. It repeats at regular time intervals. Any non-sinusoidal periodic signal can be regarded as a sum of sinusoids.

A section of a periodic signal between two consecutive maxima (or any other corresponding points) is called a **cycle**. The duration of a cycle is the **period**. The number of cycles in one second is the **frequency**. The unit of frequency is the hertz (Hz), where 1 Hz = 1 cycle per second. If $f$ is the frequency in Hz and $T$ is the period in seconds, then:

and

Also shown in Figure 1.1 is the **amplitude**, the maximum value of the sinusoid.

### Activity 1.1 Self assessment

A sine wave has a frequency of 25 000 Hz (25 kHz). What is its period?

#### Answer

*T* = , so *T* = s = 0.00004 s or 40 μs.

Sinusoidal signals also have **phase**. This relates to the part of a cycle that the sinusoid has reached at a particular time. In Figure 1.1, for example, at zero time the signal is zero and rising. Shifting the signal to the right or left changes its phase. Phase is measured in degrees or radians, and ranges from 0° to 360° (0 to 2π radians).

### Activity 1.2 Exploratory

This activity demonstrates how a sinusoid can be generated, by measuring the height of a rotating line.

It will allow you to explore the different features of the sinusoidal waveforms you have just been reading about, so that you can become more familiar with how sinusoids can be created and what happens when the different properties of sinusoids are altered.

The activity has been pre-loaded with the required settings to create a basic sine wave. When you click on 'Generate', you will see how a sine wave can be created by rotating a line of length *a* at a constant speed about a fixed point O. As the line rotates, you can see how the variation of the line marked *y* plotted against time traces out the shape of a sine wave. The frequency, amplitude and phase of the sinewave can be changed using the sliders at the top. Changes can only be made when the rotating line is stopped.

Interactive content is not available in this format.

## 1.2 Non-periodic signals

Communications is all about transferring information. A periodic signal, though, has limited possibilities for conveying information because of its predictability. After receiving a few cycles and establishing what the pattern is, we know the cycles that follow will be exactly the same. The signal may convey important information when it begins, as in the case of a fire alarm, where there is a call to immediate action as soon as the sound is heard. But a signal that never varies in amplitude, frequency, phase or any other aspect conveys little if any further information to a recipient. So in practical communications, exactly periodic signals are the exception. Signals that carry real information, such as speech, music or video, do not repeat endlessly.

Non-periodic signals (also known as aperiodic signals), unlike periodic signals, do not have just one particular frequency. Instead, they are spread out over a continuous range of frequencies. For example, a speech signal ranges from around 100 Hz to a few thousand Hz (for telephone-quality speech, a range of 300 Hz to 3400 Hz is often assumed).

## 1.3 Digital signals and modulation

Radio waves are naturally sinusoidal, with frequencies covering a wide range. They are capable of travelling through space, and are widely used for communication. This is a brief explanation of how they are able to carry information. Many of the same principles apply to other communication media, such as optical signals and electric currents.

### Activity 1.3 Exploratory

Radio waves cover a wide range of frequencies, some of which are more suitable than others for a particular service. You can explore some uses of radio with this interactive chart.

Click on the image of the electromagnetic spectrum below to learn more about the highlighted part of the spectrum (radio and microwave frequencies). You will see that this part of the spectrum is conventionally divided into bands, each covering a decade in frequency (or wavelength). Make a note of the frequencies and wavelengths and the typical uses of each band.

[The radio and microwave frequencies interactive will open in a new window. After you have viewed the interactive, click on the link 1.3 Digital signals and modulation, to return to this page.]

Interactive content is not available in this format.

Generally a medium used for communication (such as radio waves) needs to be processed in some way to carry information. The process is called **modulation**. Two signals are combined in modulation:

- The message signal, called the modulating signal. (Often this is non-periodic.)
- A signal of the right frequency for transmission, called the carrier signal.

When they are combined, the modulating signal changes the carrier signal in some way, such as by changing its amplitude or frequency. This creates a new signal that contains the message information and is also at the correct transmission frequency. Note that although modulation of some kind is essential for wireless transmission, it is also used in much wired transmission, for example broadband and optical fibre.

In the next section, assume that the message to be sent is in the form of a **digital** signal (that is, a signal that is interpreted as a sequence of discrete values). In fact, most communications fall into this category; computer networks and almost all telephony, as well as digital TV and radio. Analogue signals such as speech are converted to digital form at one end of a communications link and back to analogue at the other. When the message signal is digital, modulation produces distinct states of the carrier wave that can be distinguished by the receiver and can be used to represent ones and zeros, or groups of ones and zeros. Next you will see some basic digital modulation schemes.

## 1.4 Amplitude-shift keying (ASK)

In ASK, only the amplitude of the carrier signal is modified in modulation. The simplest version is on–off keying (OOK). In OOK, either bursts of a carrier wave are transmitted or nothing is transmitted depending whether the input message signal is 1 or 0. Other versions of ASK use differing (non-zero) amplitudes to represent 1 and 0.

Figure 1.2(a) shows a digital message signal using two voltage levels. One level represents 1 and the other represents 0. The unmodulated carrier is illustrated in Figure 1.2(b). Figure 1.2(c) and (d) are the modulated waveforms using two versions of ASK. Figure 1.2(c) uses OOK, and 2(d) uses binary ASK, or BASK.

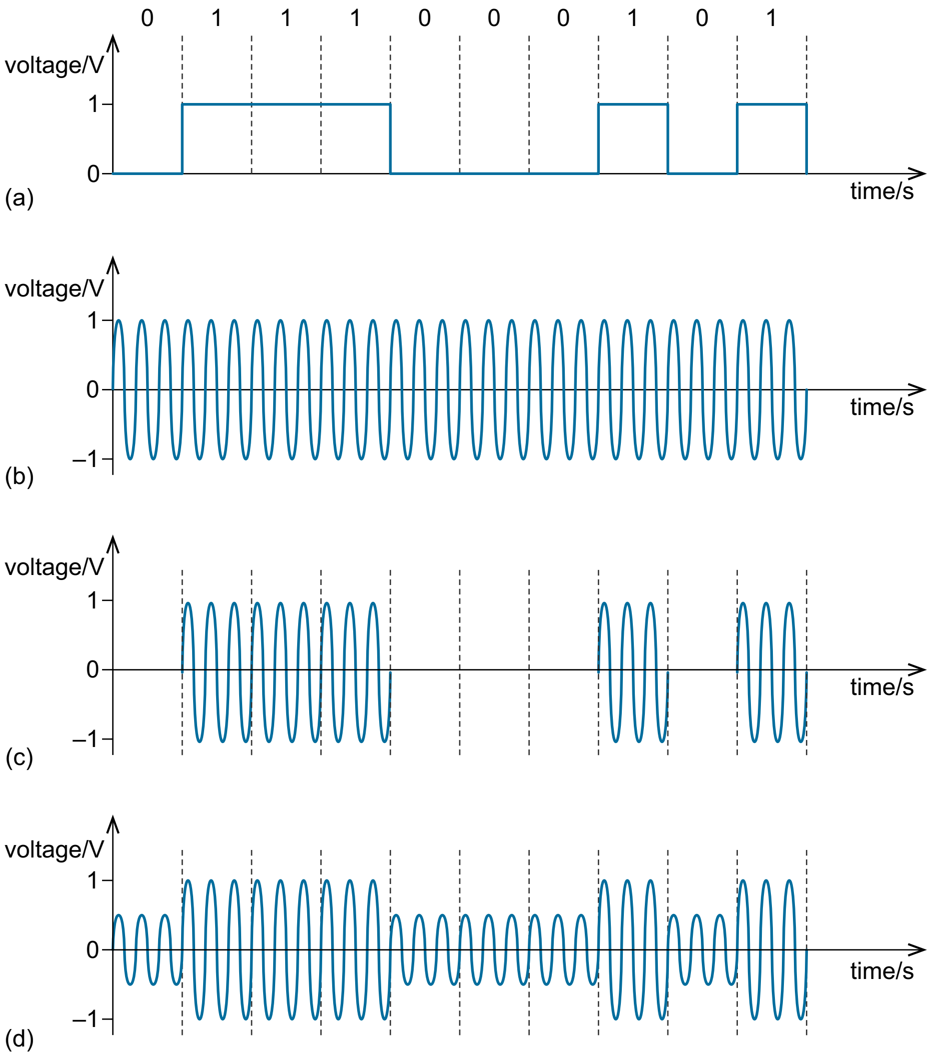



**Figure 1.2 ASK: (a) data; (b) unmodulated carrier; (c) on–off keying (OOK); (d) binary amplitude-shift keying (BASK)**

In OOK and BASK, the modulated carrier can take one of two different states: one state representing a 0, the other a 1. These different carrier states are what are known as **symbols**. If there are more than two possible carrier states – that is, more than two symbols available – then it is possible for each symbol to represent more than one bit.

Figure 1.3 shows ASK with four possible amplitude levels, or four symbols. With four symbols available, each symbol can be uniquely represented with a two-bit binary number. This is because there are just four possible two-bit binary numbers: 11, 10, 01 and 00.

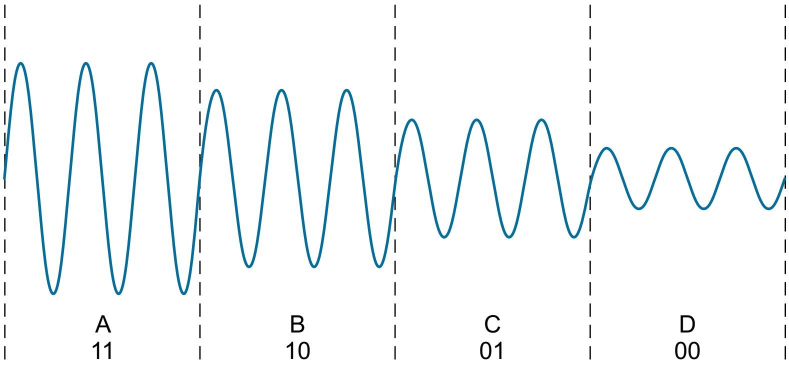



**Figure 1.3 ASK with four amplitude levels**

If there were eight symbols, each could represent three data bits. The relationship between the number of available symbols, *M*, and the number of bits that can be represented by a symbol, *n*, is:

$$M = 2^n$$

The term **baud** refers to the number of symbols per second, where one baud is one symbol per second.

Data rate (or bit rate) and baud are closely related.

### Activity 1.4 Self assessment

a. If a communications system uses 16 symbols, how many bits does each symbol represent?

b. If the same system has a symbol rate of 10 000 baud, what is the data rate?

#### Answer

a. If there are 16 symbols, then each of these can represent 4 bits, because $16 = 2^4$.

b. There are 10 000 symbols per second, and each symbol represents 4 bits, so the number of bits per second is 4 × 10 000 = 40 000. So the data rate (or bit rate) is 40 000 bit $s^{-1}$, also written 40 kbit $s^{-1}$.

Increasing the number of bits a symbol can represent means that higher data rates can be achieved.

## 1.5 Frequency-shift keying (FSK)

In FSK, the frequency of the carrier signal is modified. An illustration of binary FSK, or BFSK, is given in Figure 1.4. Here, bursts of a carrier wave at one frequency or bursts of a carrier wave at a second frequency are transmitted according to whether the input data is 1 or 0.

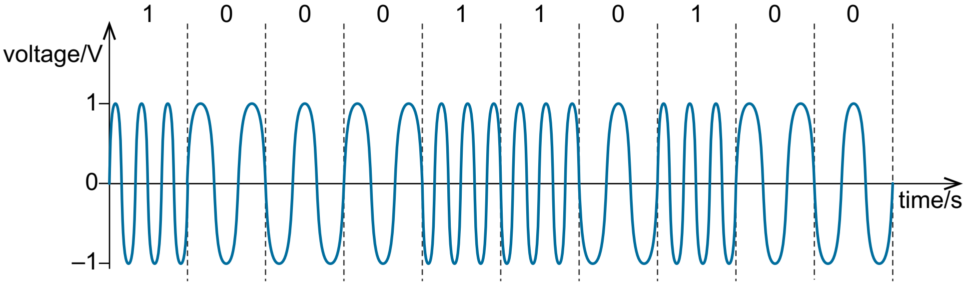



**Figure 1.4 Binary FSK**

## 1.6 Phase-shift keying (PSK)

The third fundamental digital modulation technique, and the most widely used in one form or another, is PSK. Its simplest form is Binary Phase-Shift Keying (BPSK).

In BPSK, 0 and 1 are represented by segments of sinusoids that differ in their phase. At the receiver, distinguishing between the two segments is easier if their phases differ by as much as possible. In BPSK the phases are separated by half a cycle (equivalent to π radians or 180°). See Figure 1.5.

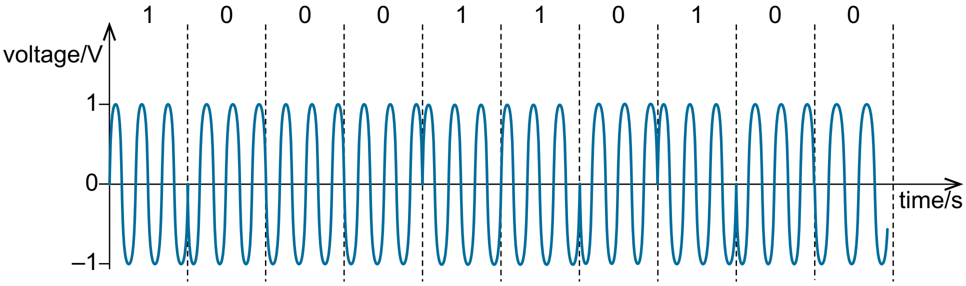



**Figure 1.5 BPSK**

A BPSK-modulated signal is less susceptible to certain kinds of noise than ASK.

### Activity 1.5 Self assessment

Figure 1.6 shows three examples of digitally modulated waveforms. For each example, decide which modulation scheme has been used and, based on the figures you saw earlier, work out what binary data each of these represents.

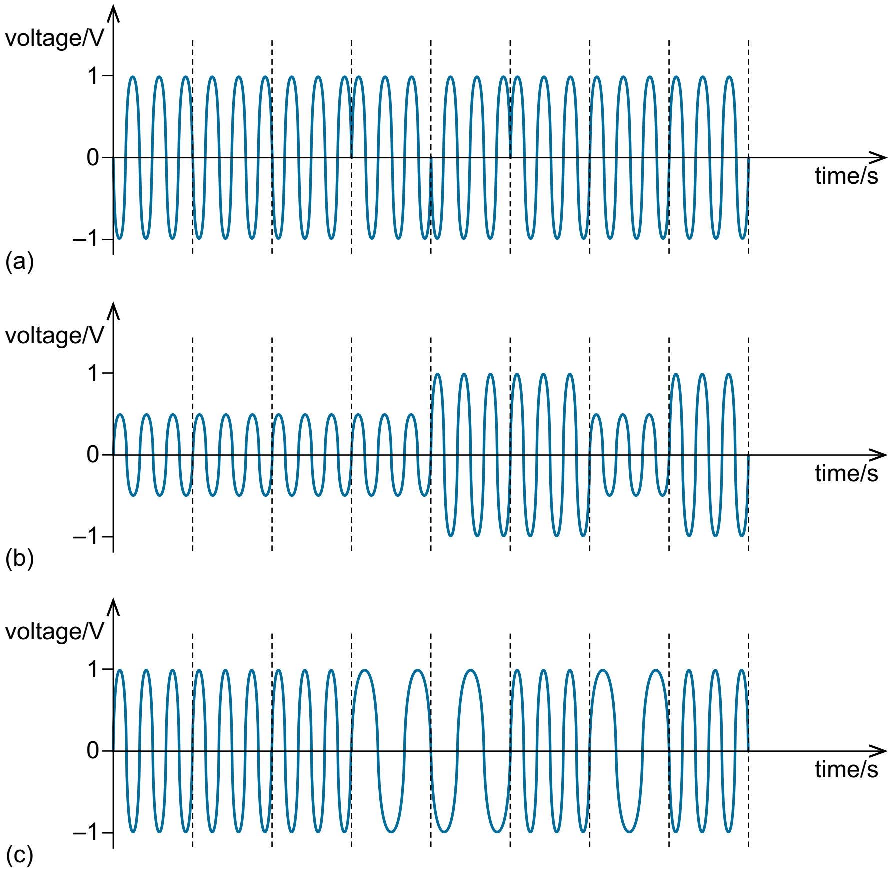



**Figure 1.6 Three digitally modulated waveforms.**

Answer

Waveform (a) is an example of a BPSK-modulated waveform representing the data:

0 0 0 1 0 1 1 1.

Waveform (b) is an example of a BASK-modulated waveform representing the data:

0 0 0 0 1 1 0 1.

Waveform (c) is an example of a BFSK-modulated waveform representing the data:

1 1 1 0 0 1 0 1.

## Activity 1.6 Exploratory

This interactive activity will allow you to explore the three binary digital modulation schemes: OOK, ASK, BFSK and BPSK.

Start the activity by clicking on the image or 'View' link below. You will see that you are invited to 'Create a binary data stream'. Enter a series of 0s and 1s, then click on 'Submit' to create a modulating waveform and use this to modulate a carrier using one of the modulation schemes. You can change the modulation scheme using the drop-down menu at the top left, and change the carrier frequency using the slider at the top right.

Try creating different modulated waveforms.

Interactive content is not available in this format.

## 1.7 Quadrature amplitude modulation (QAM)

It is possible to combine ASK, FSK and PSK. One benefit of combining different modulation methods is to increase the number of symbols available. Increasing the number of available symbols is a standard way to increase the bit rate, because increasing the number of symbols increases the number of bits per symbol. It is rare for all three methods to be combined, but very common for ASK and PSK to be combined to create **Quadrature amplitude modulation (QAM)**.

QAM is based on the application of ASK and PSK to two sinusoidal waves of the same frequency but with a phase difference of 90°. Sinusoidal waves 90° apart are said to be in a quadrature phase relationship. It is customary to refer to one of these waves as the **I wave**, or in-phase wave or component, and the other as the **Q wave**, or quadrature wave or component (Figure 1.7).

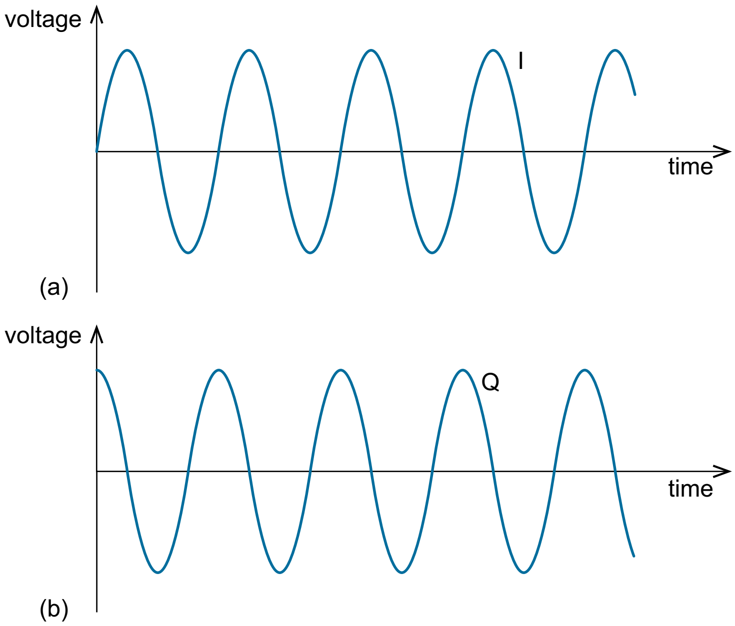



**Figure 1.7 (a) I (in-phase or sine) wave and (b) Q (quadrature or cosine) wave**

You may recognise the I wave in Figure 1.7 as a sine function and the Q wave as a cosine function. These functions are said to be **orthogonal** to each other. If two signals are orthogonal, when they are transmitted simultaneously one can be completely recovered at the receiver without any interference from the other.

The I and Q waves remain orthogonal if either or both of them are inverted (multiplied by –1, or flipped vertically). Negative amplitudes just mean that the wave is inverted.

The set of symbols in QAM can be conveniently represented on a **signal constellation diagram** (Figure 1.8). This is a plot of the I and Q amplitudes with I on the horizontal axis and Q on the vertical axis. Each dot in Figure 1.8 is a symbol, as it represents a unique combination of amplitude and phase of the I and Q waves. So, in each symbol period, only one of the 'dots' is transmitted. As there are 16 symbols, this version of QAM is called 16-QAM.

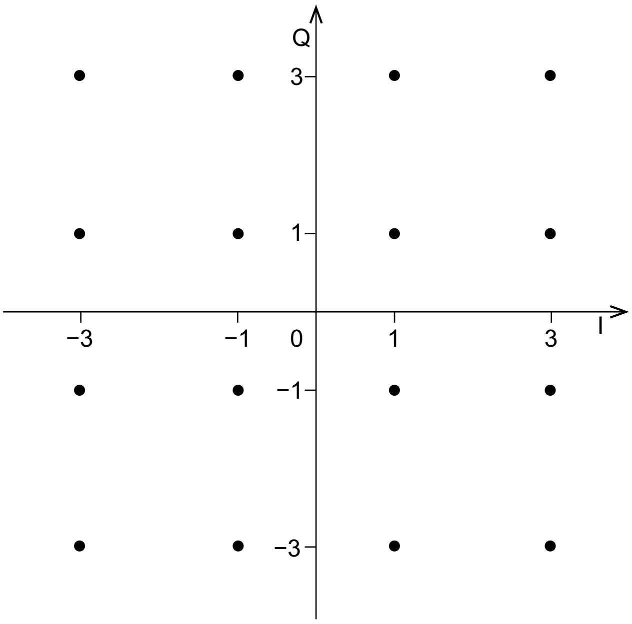



**Figure 1.8 Constellation diagram for 16-QAM.**

To understand what each dot in the diagram represents, take the top left one. This represents a symbol where the Q wave is at an amplitude of 3 and the I wave is at an amplitude of –3. The minus sign means the I wave is inverted (or phase shifted by 180°) relative to the I wave in Figure 1.7(a).

As the number of symbols increases, more data bits are transmitted per symbol. For example, 64-QAM is a QAM scheme with 64 symbols, and 256-QAM is a scheme with 256 symbols. 256-QAM conveys 8 bits per symbol (as 256 = $2^8$), so achieving twice the data rate of 16-QAM for the same symbol rate.

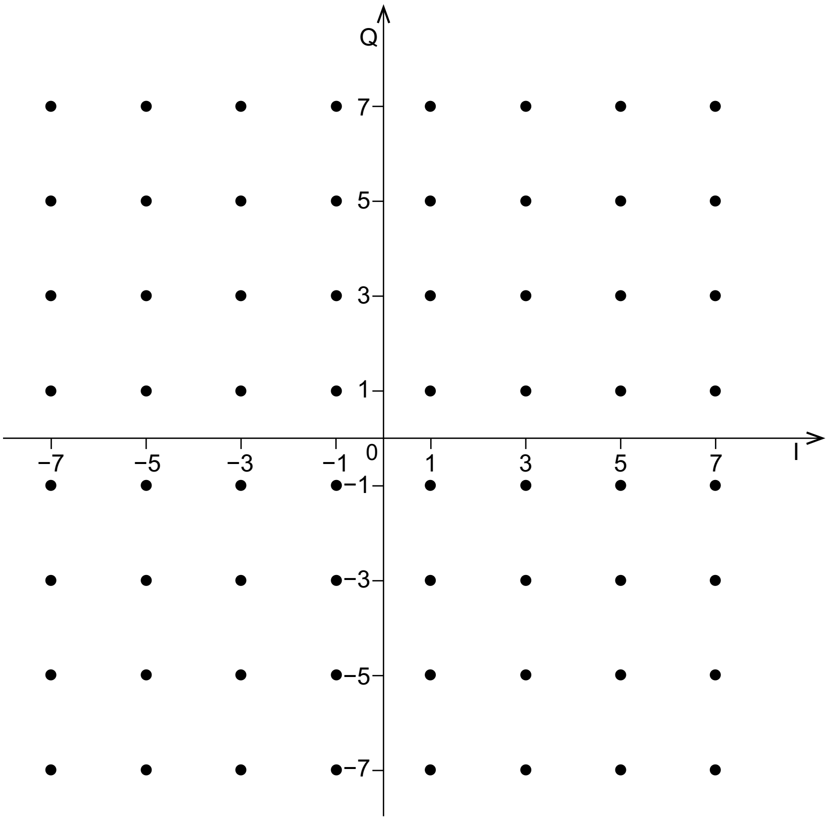
## Activity 1.7 Self assessment

How many bits are represented by each symbol in 64-QAM? Sketch a constellation diagram for 64-QAM.

Answer

For 64-QAM, the number of symbols $M$ = 64. There are six bits per symbol, as$M = 2^n$ and $64 = 2^6$.

A constellation diagram for 64-QAM might look like this:




Constellation diagram for 64-QAM

The points on the diagram in the answer to Activity 1.7 are placed at values of +/−1, 3, 5 and 7. The actual amplitudes used in practice are likely to be different; but if the spacing between constellation points remains the same (2 in this case) and we keep adding more points in this way, then we are increasing the power in the signal. The further away from the origin a constellation point is, the more power is required in the signal. Alternatively, it could be necessary to keep the maximum signal power constant whether we are using 16-QAM or 64-QAM, for instance. This would mean packing the points closer together in 64-QAM than in 16-QAM. However, if the points are closer together then adjacent symbols will be more likely to be misinterpreted at the receiver as a neighbouring symbol. One of the effects of noise (which is unavoidable in communication) is to add a degree of uncertainty about which symbol has arrived at the receiver.

## 1.8 Bandwidth

An important point to note with modulation schemes is that although the carrier signal is periodic, the resultant modulated signal is generally not periodic. (It would be periodic if the modulating signal were periodic, for example if it consisted of the repeating series 1, 0, 1, 0, etc.) Therefore, in frequency terms the modulated carrier wave occupies not just one frequency but a range of frequencies. The signal is said to extend over a certain bandwidth, measured in Hz. (Note that the word 'bandwidth' is also commonly used to mean the data rate of a digital signal, but here I am talking about *analogue* bandwidth.)

The bandwidth of an ASK signal is approximately $B_{ASK} = 2B$, where $B$ is the bandwidth of the modulating signal. This is also approximately true for PSK; $B_{PSK} = 2B$. The bandwidth of FSK depends how far apart the two frequencies used are. It is approximately:

$$B_{FSK} = 2(\Delta f + B).$$

where $2\Delta f$ is the frequency separation of the highest- and lowest-frequency symbols.

It follows that any channel conveying useful information has to use a section of the available frequency spectrum, not just one point on it. For a shared medium such as radio, this means there are limits to the number of channels that can be used at the same time and in the same place. This is a fundamental limitation in practical communications. Spectrum use is thus a major resource allocation problem.

## 1.9 Summary

Modulation is an essential part of digital communications. There are various schemes available and there are design compromises to be made between data rate, bandwidth, the likelihood of errors, complexity and so on. Modulation is used both in wireless and wired communication.

In digital communication, the unit of data transmission is the symbol. A symbol might represent a single bit of data (such as 0 or 1) or several bits (such as 0000, or 0001) depending on how many symbols are used in the modulation scheme.

Quadrature amplitude modulation (QAM) is a very widely used digital modulation system for providing multiple symbols.

# 2 Error control

Communication channels always suffer from noise, and consequently errors are unavoidable in digital communication. For example, you saw in Section 1 how noise can lead to QAM constellation points being misinterpreted for neighbouring points. The more noise there is, the greater is the likelihood of this kind of misinterpretation. Although the likelihood of errors can be reduced, for example by transmitting a more powerful signal, it can never be eliminated. Strategies such as using more powerful signals can, in any case, lead to further problems, such as increased interference with other communication channels or additional running costs.

When errors occur, they can sometimes be detected or corrected. This is **error control**. An everyday example of error control is the barcode, as in Figure 2.1. Figure 2.1 is known as an EAN-13 code. These codes are widely used as identifiers for articles such as books and other consumer items. The pattern of narrow and broad lines represent the 13-digit number shown below the lines. Scanners can read the pattern of lines and work out the number represented. Sometimes, though, the pattern is misread – perhaps because the code is unclear or damaged, or the scanner was not used properly.

A misread code produces an incorrect number, which the scanning system recognises as incorrect. Usually a warning sound then indicates that re-scanning is needed. How does the system know that the number has been misread? I will look at this in more detail shortly, but in essence only certain EAN-13 codes are valid, and valid codes are greatly outnumbered by invalid codes. Although a valid code could conceivably be misread as a different but valid code (so that, for example, a box of biscuits might be mistaken for a book), it is much more likely that misreading code will produce an invalid code – just as a random assortment of letters is more likely to be nonsense than a valid word.

The error control incorporated in EAN-13 codes is known as **error detection**. Another type of error control is **error correction**. Error correction not only allows you to know if there is an error in a code, but also corrects the error and recovers the intended data. Later on, we will look at Reed-Solomon coding, which is a widely-used method for error correction.

In this audio Allan Jones talks to David Chapman, the author of Section 2, about some of these issues.

Video content is not available in this format.

## 2.1 EAN-13 code and error detection

In the EAN-13 code, the first 12 digits of the number identify the item the code is attached to, and the final digit is a 'check digit'.

9 780521 425575

**Figure 2.1 An example of an EAN-13 barcode**

The check digit for an EAN-13 code is calculated as follows:

1. Count digit positions from the left to the right, starting at 1.
2. Sum all the digits in odd positions. (In the example shown in Figure 1, this is 9 + 8 + 5 + 1 + 2 + 5 = 30 – note that the final 5 is not included since this is the check digit, which is what we are currently trying to calculate.)
3. Sum all the digits in even positions and multiply the result by 3. (In the example, this is (7 + 0 + 2 + 4 + 5 + 7) × 3 = 75.)
4. Add the results of step 2 and step 3, and take just the final digit (the 'units' digit) of the answer. This is equivalent to taking the answer modulo-10. (In the example, the sum is 30 + 75 = 105, so the units digit is 5.)
5. If the answer to step 4 was 0, this is the check digit. Otherwise the check digit is given by ten minus the answer from step 4. (In the example, this is 10 – 5 = 5.)
6. The check digit is appended to the right of the 12 identification digits. The check digit can have any value from 0 to 9.

### Activity 2.1 Self assessment

The code below shows the first 12 digits of an EAN-13 code. Note: the hyphen between the 8 and the 0 has no bearing on the code. It is for convenience of reading, separating different elements of the identification. The 978 identifies this item as a book. (Not all EAN-13 codes have a hyphen in the same place.)

Calculate the check digit, and so derive the full EAN-13 code.

978–014102662.

#### Answer

Adding together the odd digits gives:

9 + 8 + 1 + 1 + 2 + 6 = 27.

Adding together the even digits and multiplying by 3 gives:

(7 + 0 + 4 + 0 + 6 + 2) × 3 = 19 × 3 = 57.

Adding the two together gives:

27 + 57 = 84.

The units digit is 4, so the check digit is given by:

10 − 4 = 6.

The full EAN-13 code is therefore 978–0141026626.

One way to check a received EAN-13 code for errors is to remove the received check digit and recalculate it based on the 12-digit identification code. If the recalculated value differs from the received value, there must be an error.

Alternatively, there is a shortcut to checking for errors because of the way the check digit is derived. You take the full 13-digit received code and do steps 1 to 4 from the calculation used above. If the code is correct, the value at step 4 will be 0. If the code is wrong, it will have some other value.

### Activity 2.2 Self assessment

Check whether the following codes are valid:

a. 978–0521425575
b. 978–1405322274.

#### Answer

a. 978–0521425575 is a 12-digit identification code 978–052142557 with a check digit of 5. Recalculating the check digit from the identification code gives 5, so the code is correct.

   Alternatively, using the ‘shortcut’, we take all 13 digits and go through steps 1 to 4.

   Adding together the odd digits gives:

   9 + 8 + 5 + 1 + 2 + 5 + 5 = 35.

   Adding together the even digits and multiplying by 3 gives:

   (7 + 0 + 2 + 4 + 5 + 7) × 3 = 25 × 3 = 75.

   Adding the two together gives:

   35 + 75 = 110.

   The units digit is 0, which shows that this is a valid EAN-13 code.

b. 978–1405322274 is a 12-digit identification code 978–140532227 with a check digit of 4. Recalculating the check digit from the identification code gives 0, so the code is incorrect.

   Alternatively, using the ‘shortcut’, we take all 13 digits and go through steps 1 to 4. This gives 34 + 20 × 3 = 94, which results in a units digit of 4. Since this is not zero, this is not a valid EAN-13 code.

One thing to notice about EAN-13 is that the numbers were treated as a string of separate digits, not as a single number. It was, for example, 9, 7, 8, 0, 5, 2, 1, 4, 2, 5, 5, 7, not 978 052 142 557 (i.e. *not* nine hundred and seventy-eight billion, fifty-two million, one hundred and forty-two thousand, five hundred and fifty-seven).

In EAN-13 the digits are denary: numbers to base 10. In base 10 a digit can be any one of 10 symbols, which we represent as 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. (The word ‘symbol’ is used here in a different, though parallel, sense from the symbols of modulation schemes.)

The check digit included in the 13-digit EAN code is an example of redundancy. This is a standard term for bits appended to data for error control. Unfortunately the term ‘redundancy’ suggests that these additional bits serve no purpose, which is not true. They are redundant, though, in the sense that they are not part of the message data.

All forms of error control involve the augmentation of a message with error control bits, which could be described as adding redundancy. If the number of bits (or bytes) in the message is $k$, and the augmented length is $n$ bits (or bytes), the ratio $k/n$ is known as the code rate. This is an important parameter.

Code rate is a measure of how much redundancy has been added to the code. If lots of check digits (that is, lots of redundancy) are appended to a small number of message digits, the code rate will be small (much less than 1). If only a few check digits are appended to a big message, the code rate will be close to 1. (It can never exceed 1.) Code rates found in, for example, mobile communications and WiFi typically range from 1/4 to 5/6.

Sometimes a code is specified by the numbers ($n$, $k$), in that order, with brackets around them and a comma between the numbers. Codes which take $k$ message digits and create an $n$ digit code word are described as ($n$, $k$) block codes.

### Activity 2.3 Self assessment

a. Describe an EAN-13 code using the ($n$, $k$) notation
b. What is the code rate of an EAN code?

---

#### Answer

a. There are 12 digits in an EAN-13 code, to which one check digit is added, so it is a (13,12) code
b. The code rate of an EAN-13 code is $k/n$ = 12/13

Next, we will look at how Reed–Solomon codes are used for error control on data that is structured into sequences of eight-bit bytes.

## 2.2 Reed-Solomon codes and error correction

Reed–Solomon codes (RS codes) are error-correcting codes invented in 1960 by Irving S. Reed and Gustave Solomon of the Massachusetts Institute of Technology (MIT). They have a wider range of application than EAN-13 codes. Reed-Solomon codes are used in compact disks, Blu-ray disks, DSL broadband, and in many other devices and media.

RS codes are ($n$, $k$) block codes (as are EAN-13 codes), but, whereas EAN-13 codes operate on denary digits, the digits of RS codes are bytes. A byte is a group of eight bits, but in the context of RS codes, bytes are thought of as single entities. In the same way that there are ten different possible denary digits (the digits 0 – 9), there are 256 different possible bytes. They are the different possible combinations of eight bits: 00000000, 00000001, 00000010 ... 11111111. So, whereas $n$ and $k$ are the numbers of denary digits for EAN-13 codes (13 and 12 respectively), they are numbers of bytes for RS codes. (In fact the theory of RS codes is very general and can be used with other types of digits, but in practical applications they are used with bytes.)

In RS codes, $k$ bytes of message (referred to as the message digits) have appended to them additional bytes (the check digits) to create a codeword containing a total of $n$ bytes. RS code words are much bigger than EAN-13 code words, and $n$ can be up to 255 (255 bytes).

### Activity 2.4 (exploratory)

How many bits are there in a block of 255 bytes?

#### Answer

There are 8 x 255 = 2040 bits

It is not always convenient to use the full block size, and RS codes can be **shortened**. Conceptually, the block is ‘padded’ with 0s, in other words, some of the message bits are replaced by 0s. Since the decoder knows they are 0s, they do not need to be sent, and so the block size actually used is reduced.

As RS codes are error-correcting codes, the receiver can put right errors. So, if one of the bytes in the message was sent as 01100111, for example, but was received completely differently, such as 10000100, the RS decoder will be able to change it back to the correct value: 01100111. However, error correction can only be done if there are not too many errors.

There are two steps to correcting errors in RS codes:

1. Identify which of the digits (bytes) have errors
2. Correct those digits.

Both steps involve mathematical procedures. Unlike the procedures used with EAN-13 codes, the maths is done on bytes rather than denary digits, and the theory and methods are much more advanced than those used with EAN-13 for detecting errors. Let’s look at the ‘dimensions’ of the codes and how many errors can be corrected.

RS codes can be designed with differing error-correction ability, depending upon the number of message digits in the block. For a given block size, the more message digits there are, the less redundancy there can be and therefore the fewer errors can be corrected. There will be more details on this trade-off shortly.

# Reed-Solomon codes and error correction continued

An interesting feature of RS codes is that if you already know the locations of the errors – that is, which bytes contain errors – more errors can be corrected than if you don't know where the errors are. In effect, all the information provided by the check digits can be used in correcting the bytes, instead of some of it being needed to identify which bytes have errors. It might seem a strange idea that the location of the error could already be known, but this feature is exploited in some practical applications – one of which, Blu-ray discs, shall be explained in a little more detail shortly. This type of error, where the location is known, is called an **erasure**. Bytes that have errors where the location is not known in advance are just called **symbol errors**. (Confusingly, 'symbol' and 'digit' are used to mean the same thing in this context, and the symbols/digits are bytes.)

In general, in each block of $n$ digits an ($n$, $k$) RS code can:

- correct up to symbol errors
- correct up to n − k erasures.

More generally, if there are both erasures and symbol errors to be corrected, the code can correct $v$ symbol errors and $\rho$ erasures, where:

$2v + \rho \leq n - k$.

## Activity 2.5 (self-assessment)

A popular base-256 (byte) RS code is the (255, 223) code.

a. How many symbol errors can this code correct? How many bits is that?
b. How many erasures can it correct? How many bits is that?
c. What is the code rate and the redundancy of this code?

---

### Answer

For this code, $n$ = 255 and $k$ = 223.

a. The number of symbol errors it can correct is given by:

   So the code can correct up to 16 symbol errors (16 bytes) in each block of 255 bytes. In terms of bits, it can correct up to 16 × 8 = 128 bits in each block of 255 × 8 = 2040 bits.
b. The number of erasures the code can correct is given by:

   $n - k = 255 - 223 = 32$.

   So the code can correct up to 32 erasures (32 bytes) in each block of 255 bytes. In terms of bits, it can correct up to 32 × 8 = 256 bits in each block of 255 × 8 = 2040 bits.
c. The code rate is 223/255 = 0.87 (to 2 s.f.). The redundancy (to 2 s.f.) is

   which is 13%.

Activity 2.5 showed that RS codes can correct a large number of bits. These errored bits can all be adjacent in the received bit sequence, so they could form long bursts of errors. This ability to deal with long bursts of errors is a key feature of RS codes, which is the reason for their selection in many applications.

Blu-ray discs, for example, use RS codes in a way that not only exploits their inherently good burst-error protection, but also extends it through the way data is written in the disc. Two RS codes are used, both working with bytes (base 256) and shortened from the (255, 223) code described in Activity 2.5, and they encode different kinds of data. One code, referred to as the long-distance code (LDC), protects user data – that is, the content of the disc intended for its user. This uses a (248, 216) RS code. The other code, called the burst-indicating subcode (BIS), encodes information needed for addressing and control within the disc. This is a (62, 30) RS code.

User data on the disc is organised into 64 KB chunks or ‘clusters’ (see box below). After LDC encoding, the clusters are interleaved. Interleaving is a common way of combatting bursts of noise that could affect several consecutive data units (such as clusters in the present context, or frames in other contexts). Prior to transmission, the order of the clusters is shuffled. When the clusters are restored to their correct order by the receiver, any noise bursts should have their effect dispersed, so that affected clusters are distributed among unaffected clusters, rather than being consecutive. So, if there is a mark on the disc, instead of possibly obliterating an entire coded block of data, it should cause lesser damage to a number of blocks. This by itself increases the burst lengths that can be corrected, but the BIS helps further.

**Working with G, M, K in memory sizes**

The 64 KB clusters on Blu-ray discs contain 65 536 bytes, because in memory sizes the multipliers K, M and G mean the powers of 2 that fall closest to $10^3$, $10^6$ and $10^9$ respectively. Thus:

K is $2^{10}$ = 1024 (i.e. close to $10^3$ = 1000)

M is $2^{20}$ = 1048 576 (i.e. close to $10^6$ = 1000 000)

G is $2^{30}$ = 1073 741 824 (i.e. close to $10^9$ = 1000 000 000).

Notice the relationship between K, M and G: each is $2^{10}$ = 1024 times the previous one. So if you have a number expressed using the multiplier M, say 2 Mbits, you can express this using the multiplier K by multiplying by 1024. Thus 2 Mbits = 2048 Kbits. Similarly, you can multiply a number expressed using the multiplier G by 1024 in order to express it using the multiplier M. Loosely, we might say G = 1024M and M = 1024K, and G = M × K.

Note that K has a different meaning from k, which always means $10^3$ rather than $2^{10}$.

The burst-indicating subcode, as its name suggests, is involved in detecting error bursts. The BIS data is recorded on the disc at frequent and regular intervals, so that there is a short length of encoded user data (38 bytes) between single bytes of BIS data. This can be seen in Figure 2.2, which shows the structure of a 64 KB ECC (error-correcting code) cluster, including the so-called ‘picket codes’ of the BIS data. (This particular cluster has 496 rows and 155 columns. ‘Rows’ and ‘columns’ here refer to the arrangement of bytes on a Blu-ray disc, and are not directly connected with the RS codes.)

If it is found that two or more consecutive bytes of BIS data have been corrupted, there is a fair chance that the LDC data between them will have been corrupted as well. This information is passed to the LDC decoder, which now treats the bytes in question as erasures, meaning it can correct twice as many of them compared to what it could have done if it had not known they were at fault.

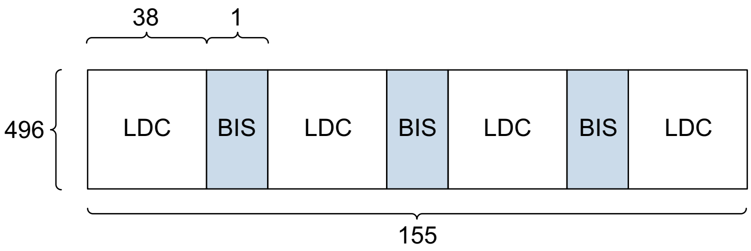



**Figure 2.2 Data structure on a Blu-ray disc**

## 2.3 Summary

In this part you have seen examples of the two broad categories of error control codes: an error detecting code (the EAN-13 code) and an error correcting code (RS code).

All error control involved the addition of redundancy to a message. If the number of message bits (or bytes) is $k$, and the augmented length after the addition of redundancy is $n$ bits (or bytes), the ratio $k/n$ is the code rate.

# 3 Perceptual source-coding and lossy compression

## 3.1 Introduction

Source coding is the representation of a phenomenon such as a sound or an image in a form suitable for communication or storage. You are probably familiar with the example of digital sampling for creating digital representation of sounds or images. Prior to transmission, such representations often have error control incorporated, as you saw earlier. There is more to source coding than just sampling, though, because of the desirability of representing a source as efficiently as possible. Usually this means using as few bits as possible, consistent with achieving a desired level of fidelity to the original source. This raises the issue of data compression, and in particular lossy compression which I will look at shortly. Lossy compression is widely used in the source coding of sound, images and video. MP3 and JPEG files use lossy compression. Usually this compression relies on human perceptual characteristics that enable some data to be discarded with no apparent degradation.

In this introductory audio, Allan Jones talks to Laurence Dooley, the author of Section 3, about some of the issues from this section.

Video content is not available in this format.

# 3.2 Sampling and quantisation

Sampling is the process of converting a continuous analogue time signal into a discrete time representation. It is one of the first stages in converting an analogue signal (of which sound is a prime example) to a digital equivalent.

Figure 3.1 shows an analogue signal sampled at regular intervals of $T_s$. During sampling, the value of the source signal is measured. In principle each sample can perfectly represent the value of the waveform at that instant.

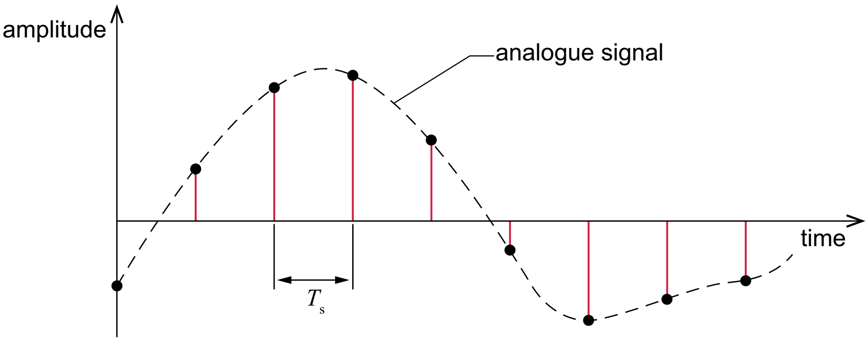



**Figure 3.1 Sampling an analogue signal**

To represent the analogue signal satisfactorily, certain sampling criteria must be met. One relates to the rate of sampling. This is governed by the **sampling theorem,** which defines the unique relationship between the source signal's bandwidth $f_b$ and the sampling frequency , where *!Warning! Calibri not supported*$T_s$ is the sampling period. There is no loss of information between the original and sampled signals *if and only if* the signal is sampled at a rate that is at least twice $f_b$:

$$f_s \geq 2f_b.$$

So, for example, a signal with a bandwidth of 4 kHz must be sampled at least 8000 times per second to preserve all the signal's information. As sampling rates are commonly given in kHz or MHz, the minimum sampling rate here would be given as 8 kHz.

If the sampling theorem is not upheld, **aliasing** occurs. A frequently encountered visual example of aliasing is the apparent backward rotation of spoked wheels in films showing vehicles travelling forwards. Audio aliasing, in which a spurious signal is represented by the samples (Figure 3.2), can be heard in the following activity.

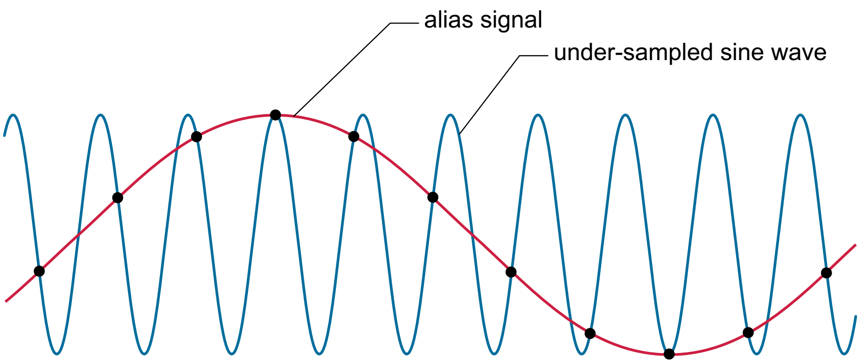



**Figure 3.2 Aliasing of an under-sampled signal**

### Activity 3.1 Exploratory

In this audio track you hear audio aliasing. A short piece of speech (‘In my garden I have an apple tree, a hazel tree and a pine tree’) is heard six times. On each repetition the sampling rate is half the rate of the one before. As the sampling rate reduces, aliasing becomes more prominent. The six sampling rates are: 44.1 kHz, 22.05 kHz, 11.025 kHz, 5.512 kHz, 2.756 kH and 1.378 kHz.

Audio content is not available in this format.

## 3.3 Quantisation

The sample values measured during sampling must be quantised to produce a digital representation of the analogue signal. That is, each value is approximated to its nearest quantisation level. Quantisation levels are pre-determined levels, like the rungs of a ladder, between the lowest possible sample value and the highest. The closeness of the approximation between a sample value and its nearest quantisation level depends on the number of quantisation levels available. For example, using a thousand levels involves less approximation than using a hundred levels.

Each quantisation level is represented by a unique binary number. The number of levels is therefore related to the number of bits, $n$, used in the binary numbers that represent the quantisation levels. For example, using 3 bits provides eight ($2^3$) discrete levels represented by: 000, 001, 010, …, 111. Eight bits and twelve bits will quantise the signal into 256 ($2^8$) and 4096 ($2^{12}$) levels respectively. In general, an $n$-bit **analogue-to-digital converter (ADC)** provides $2^n$ quantisation levels.

### Activity 3.2 Self assessment

Compact disc audio uses 16-bit quantisation.

a. How many quantisation levels are there?
b. One quantisation level is designated as zero. Can there be equal numbers of levels for positive and negative values?

---

Answer

a. For 16-bit quantisation, the numbers of quantisation levels is $2^{16}$. This is 65 536.
b. The number of quantisation levels in (a) is even, so there is no central value that can be designated as zero with equal numbers of positive and negative levels on either side. This is true whatever number of bits is used. For example, 3 bits gives 8 quantisation levels. If level 5 is taken as zero, there are four levels (1 to 4) on one side but three (6 to 8) on the other.

Quantisation inevitably introduces errors because the analogue signal can potentially take infinitely many values, whereas the number of quantisation levels is finite. You could compare this to a person standing on a staircase. The person’s height above the floor is restricted to the possibilities provided by the stairs. An object released over the side, however, passes through infinitely many heights as it falls to the floor.

Figure 3.3 shows the effect of differing numbers of quantisation levels. A sine wave of amplitude 1 V is shown quantised into 8 levels ($n$ = 3) in (a), 32 levels ($n$ = 5) in (b), 256 levels ($n$ = 8) in (c) and 65 536 levels ($n$ = 16) in (d). The corresponding digital representations are shown in red and resemble steps outlining the sine wave. The difference between the original and digital signals is called **quantisation noise (quantisation error)** and is displayed in green. As $n$ increases, the corresponding digital representation improves in fidelity because the quantisation noise (step size between adjacent discrete levels) is reduced. In Figures 3.3(c) and (d) the quantisation error is visually very small; nevertheless there will still be some quantisation noise present. This is why quantisation is lossy – in contrast to sampling, which is lossless provided the sampling theorem is upheld.

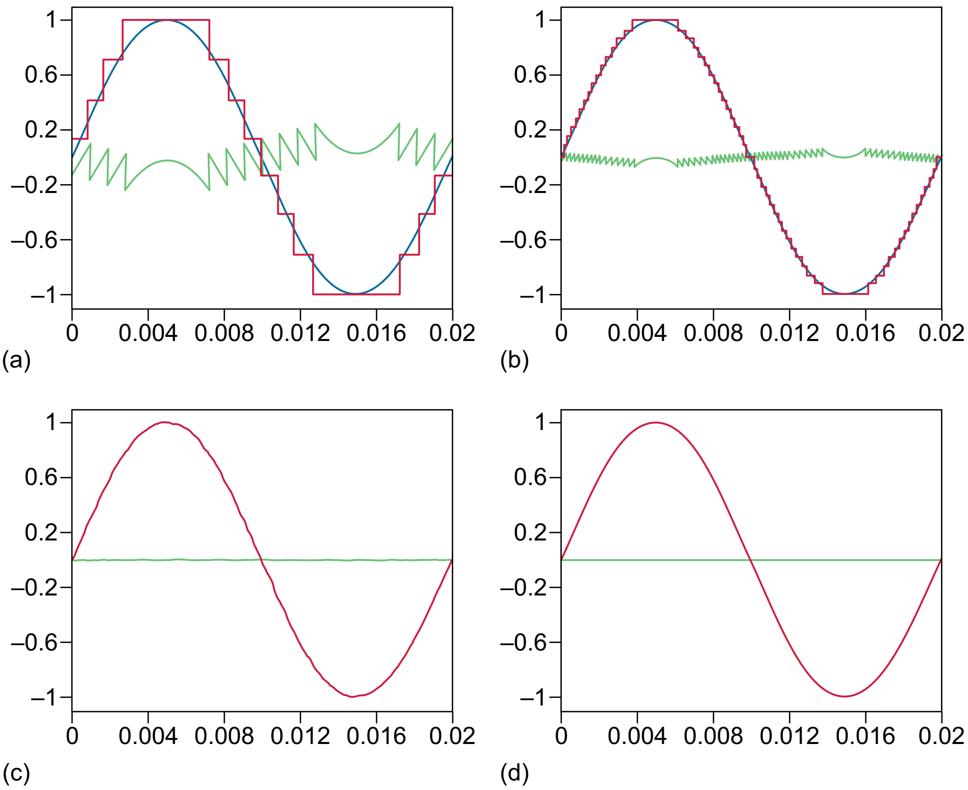



**Figure 3.3 Examples of quantisation error (green) for different ADCs: (a) 3-bit, (b) 5-bit, (c) 8-bit and (d) 16-bit**

Quantisation is an example of lossy encoding because information is irretrievably lost when infinitely variable sample values are approximated by a set of quantisation levels. In audio work, 16 bits per sample is usual for the final files, giving 65 536 quantisation levels. The choice of 16 bits per sample is an example of perceptual, lossy encoding, in the sense that the choice of 16 bits per sample is based on properties of human hearing. More bits per sample is not considered to give any audible benefit and therefore results in unnecessarily large files. (Professional recording and editing often use more than 16 bits per sample, but this is not so much for higher fidelity as because some editing processes cause a deterioration of fidelity.) However, there is more to perceptual encoding than choosing a set of quantisation levels.

# 3.4 Perceptual encoding

Sounds of certain frequencies or certain colours are perceived better than others. Useful reductions of file size or data rate can often be achieved if this fact is exploited during encoding of the source. MP3 music files for example are typically one-tenth of the size of equivalent, uncompressed music files (such as CD files).

Humans are more sensitive to frequencies in the range 1 to 5 kHz than to those outside this range. This is shown Figure 3.4. The red line is the threshold of hearing. Sounds below the threshold are inaudible. The threshold is lowest between 1 and 5 kHz. It rises above 5 kHz and below 1 kHz. At these frequencies, the quietest audible sounds are louder than the quietest audible sounds between 1 kHz and 5 kHz.

In Figure 3.4, two single-frequency tones A and B are shown with the same amplitude, but A is audible and B is inaudible.

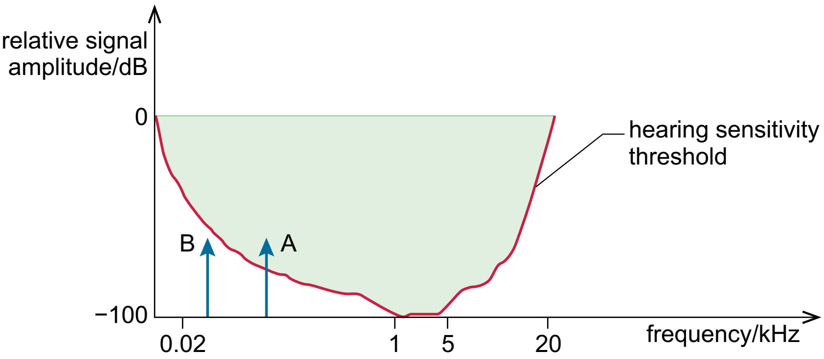



**Figure 3.4 Hearing sensitivity threshold response curve of the human ear with two equal-amplitude frequency tones, A and B**

A relatively loud sound at a particular frequency reduces our sensitivity to neighbouring frequencies. This is **frequency masking**. Figure 3.5 shows a loud sound A raising the perceptual hearing threshold in its vicinity. Sound B, which would otherwise be audible, is made inaudible. Under these circumstances, it would be unnecessary to encode sound B.

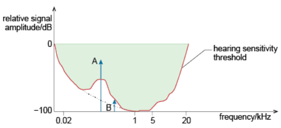



**Figure 3.5 Frequency masking for two single-tone frequencies, A and B, with A louder than B**

Another form of masking is temporal masking. This arises because our sensitivity to sounds in a narrow frequency range is reduced for a short period before and after the presence of a relatively strong sound in that frequency range. You may be surprised that sensitivity can be reduced before as well as after a relatively loud sound. This is a result of the way the auditory system and brain process audio information.

Following a loud sound, it takes the ear up to 50 ms to be able to respond again to a much quieter sound. The resulting temporal masking envelope is displayed in Figure 3.6. The shaded region represents inaudible signal amplitudes following a very strong signal at time *T*.

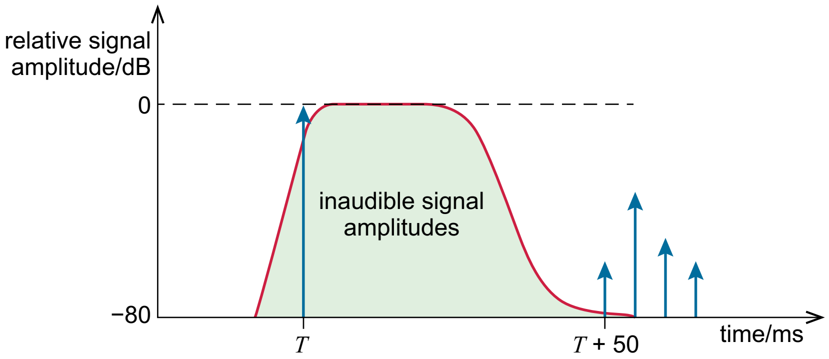



**Figure 3.6 Temporal masking effect of a loud sound at *T* and resulting inaudible envelope**

## 3.5 MPEG audio layer 3 (MP3)

In connection with frequency masking, we said that the masked sound B in Figure 3.5 did not need to be encoded. You might wonder how sounds can be selectively encoded if others are present at the same time. The answer is by splitting the audio band into sub-bands which are encoded separately. If a masked sound occupies a different sub-band from a masking sound, one can be ignored and the other encoded.

Figure 3.7 shows the elements of the creation of an MP3 audio file. The source input is generally assumed to be an audio data stream from either a CD ($f_s$ = 44.1 kHz) or studio-recorded material ($f_s$ = 48 kHz). The signal is filtered into 32 critical frequency sub-bands that are designed to reflect the way the ear perceives sounds.

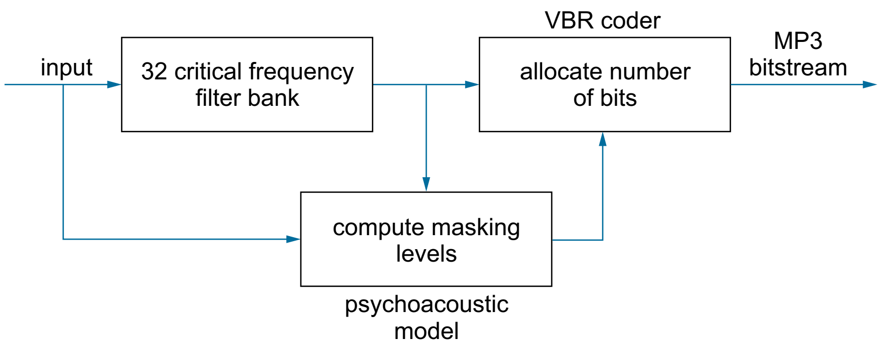



**Figure 3.7 MP3 encoder**

The 32 critical sub-bands are sampled separately, yet this does not increase the total number of samples beyond what would be required if the audio band were not split into sub-bands. Sub-bands typically have a width of 750 Hz, for which the sampling theorem requires a minimum sampling rate of 2 × 0.75 kHz = 1.5 kHz. Therefore, across the 32 sub-bands, the minimum number of samples per second must be 32 × (1.5 × $10^3$) or 48 × $10^3$. This is exactly the same as for a single band with a total bandwidth of 32 × 0.75 kHz = 24 kHz, for which the sampling theorem requires the minimum sampling rate to be 48 kHz.

Once the source signal has been split into critical sub-bands, the next step is to determine the amount of masking in each sub-band and its effect on adjacent bands – the so-called **mask-to-noise ratio (MNR)**. This makes extensive use of the two psychoacoustic masking effects of the ear discussed above to govern the appropriate quantisation levels to be used in each different frequency sub-band. Collectively these define the *masking threshold*, which determines which frequencies will and will not be coded.

If the signal level in a sub-band is below the masking threshold, it is not encoded; if it is above the threshold, it will be coded using variable bit-rate coding (VBR). In VBR, the number of bits allocated to represent each frequency component is based upon the level of quantisation noise. In digital audio, the *S*/*N* ratio is approximately equivalent to ~6 dB bit$^{-1}$ so the more bits allocated the higher the S/N ratio.

As an example, Table 3.1 shows the output levels of the first 12 critical sub-bands at a specific instant for an MP3 encoder. The output levels indicate the extent to which the level in any particular sub-band exceeds the threshold of hearing in that sub-band. If the output level were 0 in any sub-band, encoding would not be required in that sub-and because the output level would be on the threshold of audibility.

### Table 3.1: Outputs from a sub-band MP3 encoder filter

| **Critical Sub-band** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Output level/dB** | 18 | 14 | 42 | 58 | 12 | 5 | 10 | 8 | 6 | 1 | 4 | 2 |

Sub-band 4 has a high output level of 58 dB. Suppose this produces an effective masking threshold of 16 dB to sub-band 5. As 16 dB exceeds sub-band 5's output level of 12 dB, sub-band 5 does not need to be encoded in the time period covered by these output levels.

### Activity 3.3 Self assessment

Suppose sub-band 4 produces an effective masking threshold of 20 dB to sub-band 3. Does sub-band 3 need to be encoded?

#### Answer

The output level of sub-band 3 is 42 dB, which is above the masking threshold of 20 dB provided by sub-band 4, so this sub-band needs to be encoded.

# MP3 continued

The last activity showed that sub-band 3 is not, in this instance, masked by the loud sound in sub-band 4. However, the raising of the threshold by 20 dB means that for encoding purposes sub-band 3's output level is reduced. Specifically, as the output level of sub-band 3 exceeds the threshold by (42 – 20) dB, the effective level that needs to be encoded is only 22 dB. In the VBR encoding used in MP3, 1 bit is allocated per 6 dB of level above the threshold. This means that sub-band 3, which exceeds the threshold by 22 dB needs an allocation of 4 bits to encode this sample. An allocation of 3 bits would be insufficient as 3 × 6 dB = 18 dB, which is below the level of 22 dB, whereas 4 × 6 dB = 24 dB, which is above 22 dB.

The procedure outlined here has to be carried out across all sub-bands where there is frequency masking from other sub-bands. The effect of temporal masking also has to be taken into account. These processes have to be repeated to cover the entire duration of the recording.

MP3 achieves high-quality audio reproduction at 128 kbit $s^{-1}$. This contrasts markedly with the CD bit rate of 1.4112 Mbit $s^{-1}$. MP3 generally achieves 10:1 compression without introducing notable subjective effects into the reconstructed sound. Incidentally, it is common to refer to compressed audio files in terms of a bit rate in kbit $s^{-1}$ or Mbit $s^{-1}$ rather than as an actual file size. The reason for this convention is that MP3 and other audio formats are extensively used in streaming applications where the emphasis is on throughput and *quality of service* (QoS) rather than storage capacity.

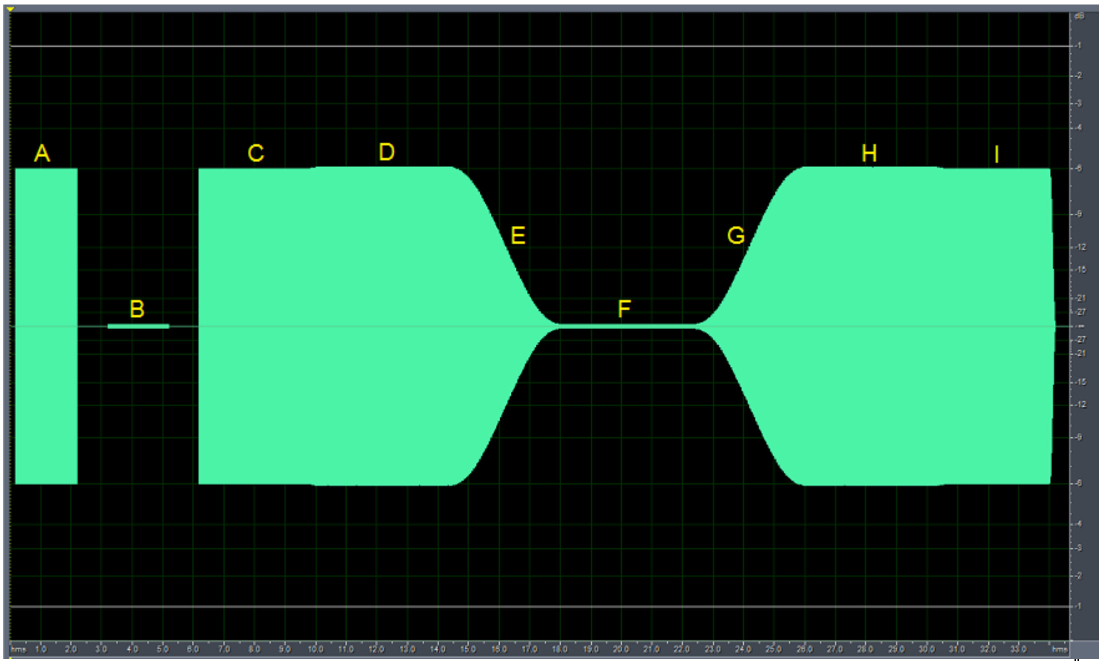
The majority of MP3 recordings are made at 128 kbit $s^{-1}$, which provides sufficient audio quality that the majority of people (apart from hi-fi buffs) would not notice the difference. As the bit rate drops to 64 kbit $s^{-1}$, however, the loss becomes much more perceptible at the top (treble) end. The bass response also tends to degrade, and higher frequencies take on a distinctly artificial digital tone. The reason for this is that the MP3 developers decided to limit the audio bandwidth to approximately 16 kHz for 128 kbit $s^{-1}$ and only approximately 8 kHz for 64 kbit $s^{-1}$.

### Activity 3.4 Exploratory

This activity: 'Perceptual sensitivity and masking' allows you to explore some audio examples of the relative hearing sensitivity response of the ear, as well as frequency and temporal perceptual masking effects.

## Frequency masking

To demonstrate frequency masking, you will hear a relatively loud sine-wave tone (440 Hz) masking a quieter tone at a different frequency (652 Hz). The image below provides a visual representation of the audio clip: the horizontal direction represents time, the vertical direction represents amplitude, and the green shapes are the envelopes of the sine-wave tones. The sine waves are too closely packed for their cycles to be visible. You will probably find it helpful to look at this image while you play the audio clip.




The first two sounds in the clip are simply to familiarise you with the masking tone (440 Hz, shown in the figure at **A**) and quieter masked tone (652 Hz, shown at **B**). You will hear 2 seconds of each.

The masking demonstration follows. There are 4 seconds of the 440 Hz masking tone (**C**). The quieter 652 Hz tone is then added, and the two tones are played for 4 seconds (**D**). When you play the audio clip, try to identify whether you can hear the 652 Hz tone during this part.

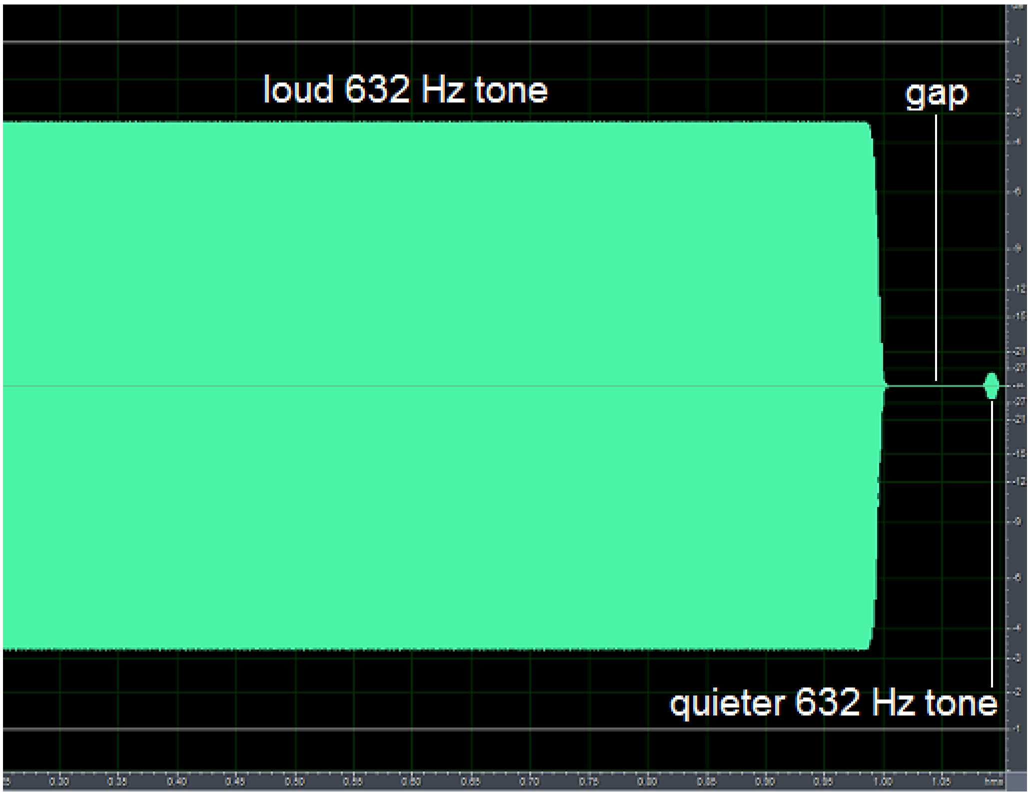
The 440 Hz masking tone then fades out (**E**), leaving just the 652 Hz tone for 4 seconds (**F**). Finally, the 440 Hz masking tone gradually fades back in (**G**) and should eventually mask the 652 Hz tone (**H**). The audio clip ends with 4 seconds of just the 440 Hz tone again (**I**).

Play the audio clip now.

Audio content is not available in this format.

## Temporal masking

In temporal masking, a loud sound makes a closely following sound inaudible. The effect is most noticeable when the following sound is relatively quiet, and when it follows after a very short gap. The image below shows the sequence of sounds used in each of the demonstration audio clips.




In each of the four audio clips below, a relatively long, large-amplitude 632 Hz tone is followed by a gap, and then a quieter version of the same tone.

In the first clip, the gap between the two tones is fairly long (60 ms). The tone after the gap is audible as a very short blip (like a faint echo) after the main tone.

In successive clips, the gap gets shorter. You should find that the final blip becomes inaudible as the gap decreases to 10 ms.

Audio content is not available in this format.
Gap = 60 ms

Audio content is not available in this format.
Gap = 40 ms

Audio content is not available in this format.
Gap = 20 ms

Audio content is not available in this format.
Gap = 10 ms

## 3.6 MPEG-4 AAC (advanced audio coding)

MPEG-4 AAC (advanced audio coding) was designed as the successor to MP3 for low-bit-rate perceptual audio compression, with efficient internet multimedia streaming applications in mind. Its development was also motivated by the quest for efficient coding of multichannel surround-sound signals. So-called '5.1 surround sound' includes five full bandwidth channels (left, right, centre, left surround and right surround), with the 'point 1' referring to a dedicated **low frequency effect (LFE)** channel carrying bass information in the 3 to 120 Hz band.

AAC has now been formally embedded in both the MPEG-2 and MPEG-4 audio standards; it is the default format for various multimedia applications and services, from YouTube to Apple's iTunes. The broad consensus is that, subjectively, the AAC encoder (.mp4 files) provides better audio quality for the same bit rate as MP3, with greater flexibility and functionality. In comparison with MP3, AAC offers a range of sampling rates up to 96 kHz, and also supports up to 48 channels (mono, stereo and multichannel surround sound). In terms of coding, it uses either 2048 or 256 sub-bands compared to 32 for MP3, thus providing better frequency resolution for the psychoacoustic modelling and perceptual masking steps.

Another noteworthy feature of AAC encoders is that audio files do not have to be encoded at a specific streaming speed. Instead the file is coded once, then streamed at a variable bit rate depending on the connection speed and network traffic conditions. This is a consequence of AAC supporting scalable representations in terms of sample amplitudes (or *S*/*N* ratio) and sampling rates.

MPEG-4 AAC and its variants excel at low bit rates by virtue of a series of extensions and tools that have evolved and subsequently become embedded into the standard. Figure 3.8 identifies three key tools that have been instrumental in the advancement of this standard:

perceptual noise substitution (PNS)

spectral band replication (SBR)

parametric stereo (PS)

Further information on each of these is readily available on the Web. While each tool to some extent adds complexity to the encoder, it also provides notable improvements in coding efficiency and corresponding audio quality.

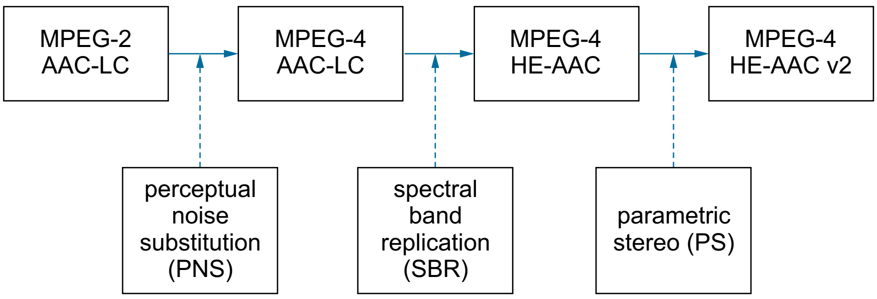



**Figure 3.8 MPEG-AAC audio encoder family**

AAC-LC (low complexity) is the most widely used coding profile in this standard, and the default format for Apple's iTunes. Since AAC involves many varied processes in analysing different types of audio signal, no single algorithm is able to meet the diverse set of requirements it must fulfil. Therefore AAC has integrated different applications into a single framework covering music synthesis, low-bit-rate speech coding, text-to-speech synthesis and general perceptual audio compression across a host of different bit rates.

The most recent AAC extension is High-Efficiency AAC (HE-AAC) also known as AACplus. It is specifically optimised for very-low-bit-rate applications such as audio streaming and podcasting, and is now the standard technology used in digital radio broadcasting. It embraces SBR technology to encode and store high frequency information as part of the standard, and is able to deliver near-CD quality sound at 64 kbit $s^{-1}$. At the time of writing, the most recent version is HE-AAC version 2, which employs the third major extension in Figure 3.8 – *parametric stereo* (PS) – to improve the audio quality at low bit rates and increase compression by up to 40%. This analyses the spatial characteristics between the left and right channels of a stereo signal to exploit inter-channel redundancies. PS characterises the inter-channel features of the stereo signal and, depending on the source, typically provides a bit-rate saving of up to a factor of 10.

### Activity 3.5 Exploratory

This 'Audio coding' activity allows you to compare several versions of the same audio sample that have been compressed using different standards.

In this activity you will hear a sample of speech that has been processed with different compression formats. In the order in which you will hear the speech samples, the formats used are the following four:

- MP3
- AAC LC
- HE-AAC v1
- HE-AAC v2.

This is theoretically the order of increasing quality.

All four extracts are at a bit rate of 16 kbit $s^{-1}$. This low bit rate has been chosen to emphasise the differences in quality between the formats, which are less noticeable at higher bit rates.

The speech extract used consists of the following two sentences:

> In my garden I have an apple tree, a hazel tree and a pine tree. My neighbours have an apple tree too.

With each repetition the quality should improve, although many people find little difference between the second and third versions (AAC LC and HE-AAC v1).

Play the audio clip now.

Audio content is not available in this format.

Since the greatest difference is between the first and last extracts in the above sample, the following sample uses just those extracts (that is, MP3 followed by HE-AAC v2).

Audio content is not available in this format.

## 3.7 Image and video compression

Section 3 has concentrated on lossy, perceptual coding in audio files, but it is widely used in image coding and video coding.

Lossy compression in JPEG image coding exploits the fact that the human visual system is less sensitive to fine detail in an image than to broader features. JPEG coding transforms an image to spatial frequency components using a discrete cosine transform (DCT), then uses fewer bits to encode the higher spatial frequencies than the lower ones. Thresholding completely removes those components with very low amplitudes. This is a lossy process. Further lossless compression completes the process, but overall the process is lossy.

As moving images consist of sequences of still images, the first step in video compression is to compress the individual still images (frames). The MPEG family of standards use methods based on JPEG for compressing the still images, but then use techniques based upon motion prediction and compensation to exploit the temporal similarities between consecutive image frames. Recent trends in video coding have led to the development of systems that incorporate multiview and 3D information, as well as more distributed approaches to video coding that shift some of the coding complexity from the encoder to the decoder.

## 3.8 Summary

Lossy compression is a type of source coding in which information is irretrievably lost. Quantisation is an example, but the term is mostly used in connection with perceptual coding in connection with audio, image and video.

MP3 audio coding uses lossy compression by exploiting frequency masking and temporal masking. The MP3 encoder identifies parts of a sound that are not perceived due to these two masking effects and does not encode them.

Advanced audio coding (AAC) (used in MPEG 4) supports efficient multimedia streaming and is also used with surround sound. It offers a wider range of options than MP3 coding, and can use three key audio quality extensions without excessively increasing the bit rate: perceptual noise substitution, spectral band replication and parametric stereo.

Perceptual coding is used in image files (such as JPEG) and in MPEG video files.

# 4 Broadband, mobile and WiFi

In this introductory podcast Allan Jones talks to Helen Donelan about issues related this section.

Video content is not available in this format.

## 4.1 Introduction; access and core networks

Because of the ubiquity of wireless communications, such as WiFi and cellular mobile communications, it is tempting to think that it is only a matter of time before fixed-line communications, such as broadband and optical fibre, are superseded by forms of wireless communication. This is unlikely to occur because in many countries wireless communication depends on an infrastructure of fixed-line communications. In addition, it is likely that fixed-line communications will generally (but not in all cases) out-perform wireless communication because of the greater amount of frequency spectrum available in fixed-line communications.

The major fixed-line infrastructure in countries with a long history of telecommunications (such as the UK) is a 'legacy' public switched telecommunications network (PSTN) initially devised for telephony. The PSTN now carries much more than telephony. Parts of it have become essential infrastructure for carrying data as well as voice. In addition, there are newer networks such as those created for mobile telephone and data services. A useful structural division of these networks is as follows.

> A core network, which is largely a fixed, high-speed, intensively used communications network. It is somewhat analogous to a network of motorways and major trunk roads. Core networks often interconnect with other core networks. For example, all the mobile operators' core networks interconnect with the PSTN core network.

An access network, which links end-users' equipment to the core network via a local exchange or local radio node. The access network is analogous to the minor roads that give access to motorways and other trunk routes.

Consumer premises equipment (CPE) consists of the devices used by subscribers for consuming data (for example, fixed-line telephones, computers and fax machines). In the mobile world, user equipment (UE) is the term used for this part of the network.

This section is concerned with three widely used access networks:

DSL broadband, which is currently the most widely used form of fixed-line broadband;

fourth generation (4G) mobile broadband;

WiFi.

Other forms of access, which this section does not look at, include:

third generation (3G) mobile broadband;

'cable', more properly known as Hybrid Fibre-Coaxial (HFC), which delivers television, broadband and telephony;

optical fibre, which, in the form of 'fibre to the premises', can be used to give access to the core network (which itself is usually based on optical fibre).

## 4.2 Orthogonal frequency division multiplexing (OFDM)

Orthogonal Frequency Division Multiplexing, (OFDM) and its close relative Orthogonal Frequency Division Multiple Access (OFDMA), are widely used forms of modulation. They are used in DSL broadband, 4G mobile communications, WiFi, digital television, powerline communications, cable television and digital audio broadcasting (DAB).

Modulation always spreads the power of a transmission around the carrier frequency. In conventional frequency division multiplexing (FDM), as used in radio broadcasting for example, carrier frequencies are well separated to prevent the intrusion of power from neighbouring carriers. Figure 4.1 shows power distribution around some modulated carriers.

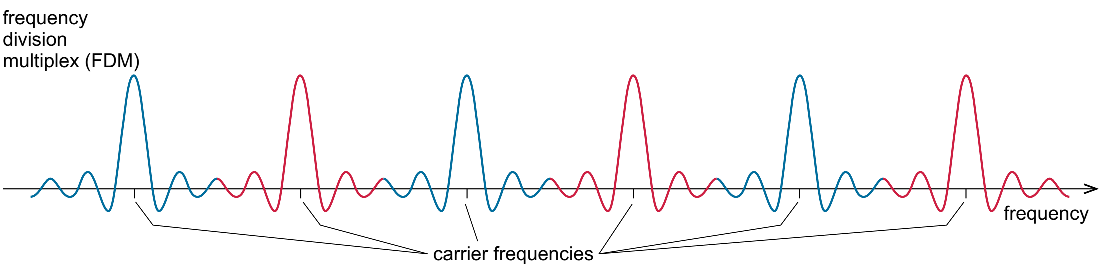



**Figure 4.1 Wide carrier separation in Frequency Division Multiplexing**

In OFDM, by contrast, all the carriers (known as subcarriers) are closely packed. As a result, their spectra overlap. The expected mutual interference doesn't occur because each subcarrier frequency coincides with zero power in all the other subcarriers, as shown in Figure 4.2. The subcarrier spacing required to make this happen is determined by a mathematical relationship between the spacing and the symbol rate of the modulation.

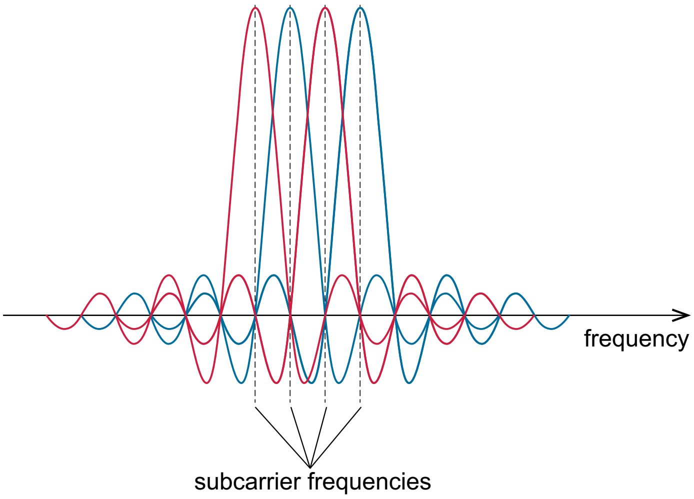



**Figure 4.2 OFDM subcarriers**

All the subcarriers operate at the same symbol rate, but each is usually modulated independently of the others. We can think of each subcarrier as being at the centre of a narrow frequency channel, as in Figure 4.3. In OFDM, these channels are called subchannels, and their width is equal to the subcarrier spacing.

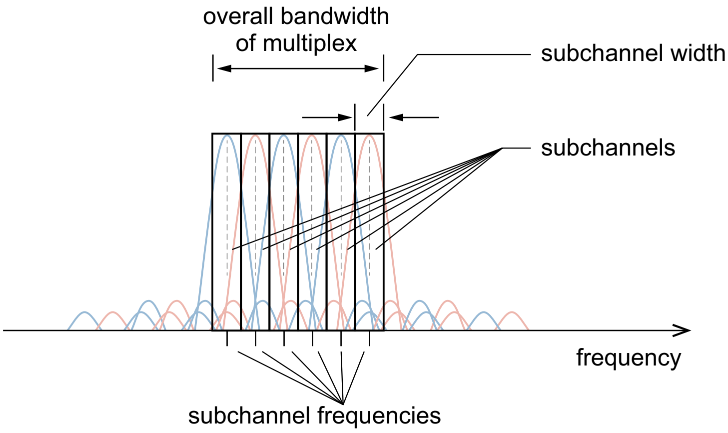
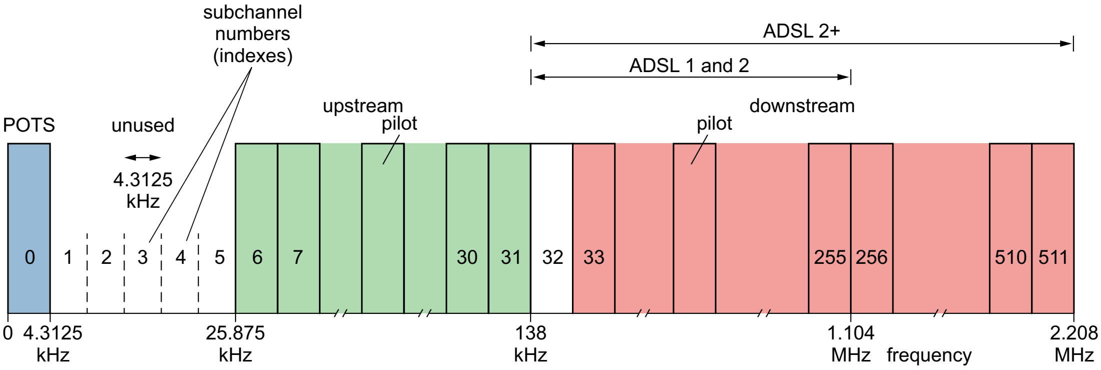



**Figure 4.3 Subchannels centred on subcarrier frequencies**

A good example of the use of subcarriers and subchannels is Digital Subscriber Line broadband, or DSL.

## 4.3 Digital Subscriber Line (DSL) broadband

In DSL an ordinary telephone line (consisting of a pair of twisted copper wires) is used to deliver broadband to homes and offices. DSL uses a variant of OFDM called Discrete Multitone (or DMT). A common form of DSL is Asymmetrical DSL, or ADSL, which itself is available as ADSL1, ADSL2 and ADSL2+. Figure 4.4 shows ADSL subchannels which are numbered from zero upwards.

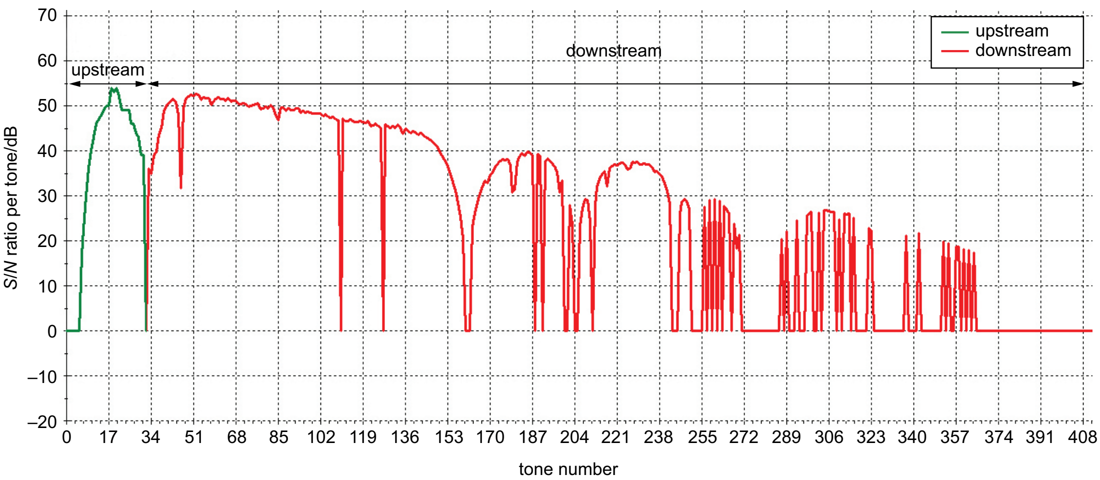



### Figure 4.4 ADSL subchannels

The subchannels in DSL are 4.3125 kHz wide. Subchannel zero is reserved for the 'Plain Old Telephony Service' (POTS), which is analogue telephony. Subchannels 6 to 31 are for upstream data (from the user to the telephone exchange), and subchannels from 32 upwards are for downstream data (from the exchange to the user).

Some subchannels are unused, for example subchannels 1 to 5 and 32. These provide a guard band between groups of subchannels. Other subchannels do not carry user data but are used for pilot tones required for the proper functioning of the system.

Downstream subchannels greatly outnumber upstream subchannels, hence the 'asymmetry' of ADSL. The downstream data rate is therefore much higher than the upstream rate.




### Figure 4.5 Signal-to-noise ratio of a 2 km long telephone line

Figure 4.5 is the signal-to-noise ratio of a 2 km long telephone line. The 'tone numbers' on the horizontal axis are the same as subchannel numbers. The line gets noisier at higher numbers, which almost always happens. There are places where the subchannels are so noisy as to be unusable, and this is commoner at the upper end.

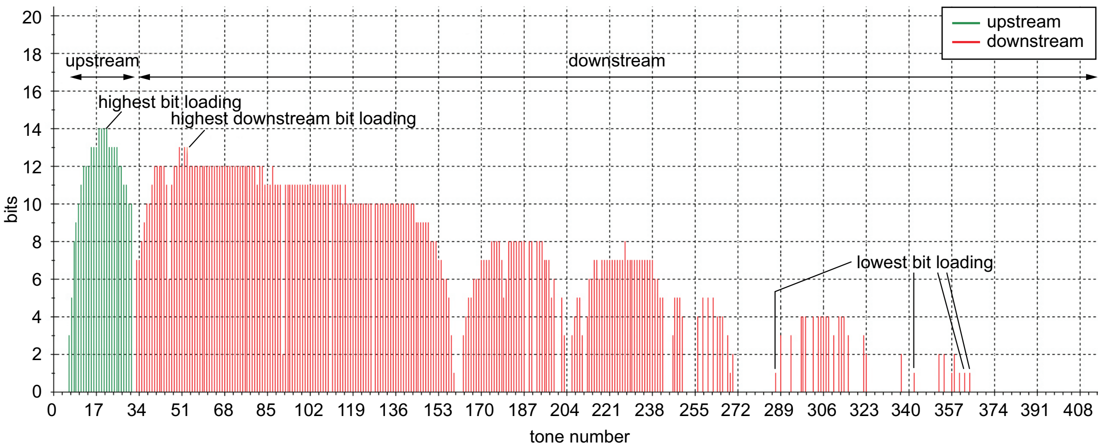



**Figure 4.6 Bit loading of a 2 km ADSL2+ line**

Figure 4.6 shows the bit loading per subchannel, which is the number of bits per symbol per subchannel. The better, less noisy subchannels (generally at lower numbers) are loaded with more bits than are the poorer subchannels. QAM modulation is used in each subchannel, enabling the bit loading to be varied. For example, 64-QAM gives 6 bits per symbol (because $2^6 = 64$), and 16-QAM gives 4 bits per symbol (because $2^4 = 16$). There are 4000 symbols per second in the widely used versions of DSL, so Figure 4.6 shows the number of bits per subchannel per symbol period.

Because each telephone line has a different noise characteristic which varies over time, a 'one size fits all' modulation scheme,(where there is a standard order of QAM for each subchannel), would not be very efficient. OFDM, however, allows optimum use of each line because the subchannel bit loading can be automatically customised for the prevailing noise at each location. The bit loading can change as noise conditions change during the day, or from day to day.

If a particular subchannel is not loaded to its capacity, it can be topped up with bits that would otherwise have exceeded the capacity of another subchannel. This process is known as **bit swapping**, and is a distinguishing feature of DMT (as opposed to OFDM).

Despite the order of QAM being chosen to suit the noise conditions, errors still occur. Reed-Solomon error correction is therefore incorporated into DSL, and on particularly poor lines interleaving of data units may also be used. This increases the latency of data transmission.

## Activity 4.1 Self assessment

If every downstream subchannel in ADSL2+ were loaded to 13 bits, what would be the data rate? Ignore unused subchannels.

Answer

There are 4000 symbols per second. Each symbol in each subchannel is loaded to 13 bits, so data rate per subchannel is

$4000 \times 13 \text{ bit s}^{-1} = 52 \text{ kbit s}^{-1}$

The highest numbered downstream subchannel is 511. The lowest is 32. The total number of subchannels is therefore 480. Therefore the overall data rate is

$480 \times 52 \text{ kbit s}^{-1} = 24.96 \text{ Mbit s}^{-1}$

## 4.4 VDSL2

VDSL2 (version 2 of Very High Bit Rate DSL) achieves higher data rates than ADSL by shortening the copper line. Shortening results from the use of a locally installed connection box like Figure 4.7.


**Figure 4.7 Fibre box used with VDSL2**

The box contains, effectively, a piece of telephone-exchange equipment which is linked to the local exchange by optical fibre. Subscribers have the copper line from their home terminated at the box – at least as far as data is concerned. (Telephony continues to use copper wire to the exchange.)

Because signal attenuation per metre of copper wire increases with increasing frequency, a major benefit of a short copper line is that higher frequencies can be used and therefore more subchannels made available. VDSL2 uses many more subchannels than any version of ADSL. Various frequency plans, or profiles, are available for VDSL2. Figure 4.8 is a common one, called ‘17a’, in which the highest frequency is just above 17 MHz.

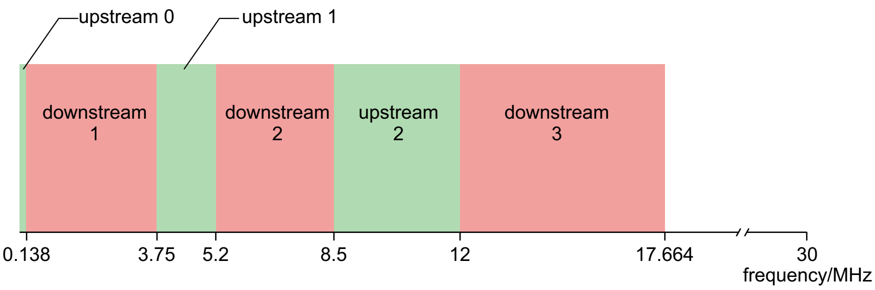



**Figure 4.8 VDSL2 profile 17a**

Downstream data rates with VDSL2 are typically around 30 to 80 Mbit $s^{-1}$, and upstream rates can be in the region of 20 Mbit $s^{-1}$, although providers often set a lower rate.

Figure 4.9 is a setting screen from a domestic VDSL2 broadband router. A range of VDSL2 profiles is shown; and because the symbol rate and subchannel width are standard across VDSL2 and ADSL the router can also cope with ADSL (as evident in the 'Modulation' row). 'G.DMT' is an informal name for the first ADSL standard, now usually referred to as ADSL1. 'G.lite' is a very basic form of ADSL that did not catch on. 'SRA' is Seamless Rate Adaptation and is one way of dynamically changing the speed of the connection according to noise conditions.

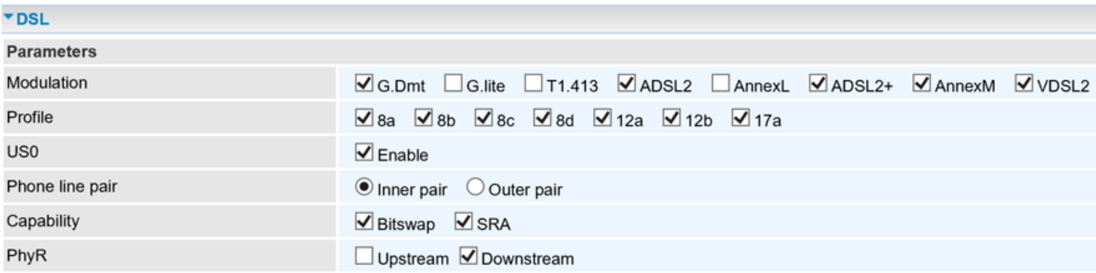



**Figure 4.9 Settings screen of a domestic DSL router**

# 4.5 4G mobile broadband

Fourth generation (4G) mobile communication is regarded as part of a so-called Long Term Evolution from 3G, and is often referred to as LTE, or 4G LTE. It was the first mobile communication system designed only for data (as opposed to 3G's voice-and-data design, and 2G's voice-only design on to which a data service was grafted). The radio bands used are generally in the region of 800 MHz, 1.8 GHz, 2.1 GHz and 2.6 GHz.

4G uses OFDMA in the downlink (from the base station to the user). The uplink (from user to base station) uses the related Single Carrier Frequency Division Multiple Access (SCFDMA), which is more energy-efficient. (SCFDMA is not covered here.) The uplink and downlink generally occupy different frequency bands, in common with the widely adopted versions of 2G and 3G.

All mobile data communication uses the idea of a unit of resource in the downlink. A resource unit is a stream of data allocated exclusively to a user for, typically, a short time. Figure 4.10 represents a resource block used in 4G. Subchannel index numbers are placed up the vertical axis (unlike the earlier diagrams for DSL where they were on the horizontal axis). There are 12 subchannels in the block, each 15 kHz wide, so a block occupies 12 × 15 kHz = 180 kHz of spectrum. A base station might typically use a channel

20 MHz wide, in which case the number of resource blocks available would be in the region of 110.

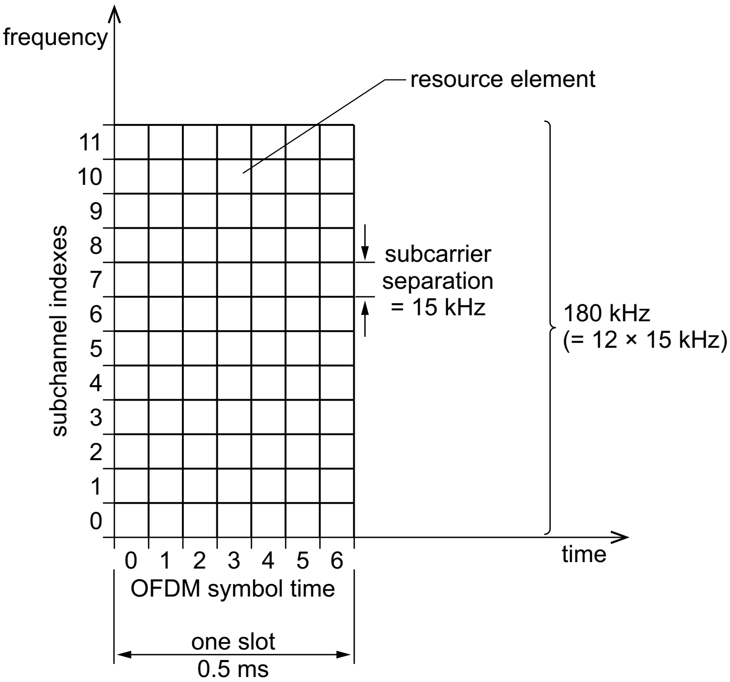



**Figure 4.10 Resource block for 4G**

The horizontal axis in Figure 4.10 represents time, but in units of symbol periods (i.e. symbol duration). Symbols 1 to 6 each have a symbol period of slightly under 71.4 ms. Symbol 0 is given a slightly longer symbol period of 71.8 ms just to make the total slot time up to 0.5 ms. For poorer signal conditions the slot time of 0.5 ms is divided into six symbol periods rather than seven, giving a lower data rate but greater resilience to noise.

## Activity 4.2 Self assessment

a. If 16 QAM is used throughout the resource block in Figure 4.10, how much data does this block convey?

b. Hence what data rate is represented is represented by a continuous allocation of 1 resource block to a user?

### Answer

a. Each resource element in Figure 4.10 is one symbol in one channel. In 16 QAM a symbol carries 4 bits of data. There are 7 × 12 = 84 resource elements in the block. Each is loaded with 4 bits, so total data is 84 × 4 bits = 336 bits.

b. Each block lasts 0.5 ms, so there are 2000 blocks per second. A continuous allocation of one block to a user gives a data rate of 2000 × 336 bit $s^{-1}$ = 672 kbit $s^{-1}$

The smallest number of resource blocks that can be allocated to a user at any moment (apart from zero) is 6. The shortest duration of an allocation is two slots, or 1 ms. Overall, therefore, the smallest allocation possible is conceptually six resource blocks high and two resource blocks (slots) wide, giving a minimum resource allocation of 12 resource blocks in total (Figure 2.10). The six resource blocks are not necessarily adjacent in frequency.

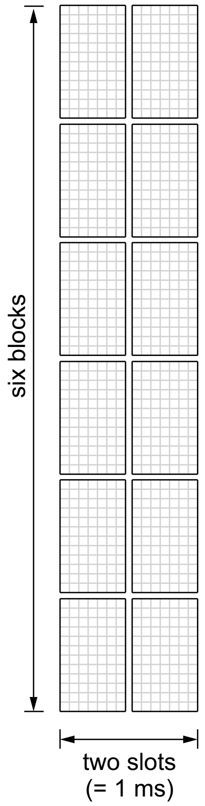



**Figure 4.11 Smallest resource-block allocation in 4G**

The 1 ms duration of the minimum allocation is the scheduling interval. At intervals of 1 ms the allocation of resources to users is reviewed, and the resource allocation and modulation methods changed if necessary, for example, in order to take account of changing link quality between base station and user.

## Activity 4.3 Self assessment

What data rate does a continuous minimum allocation give if 64 QAM is used?

### Answer

Activity 4.2 shows that allocating a single resource block using 16-QAM gives 672 kbit $s^{-1}$. Whereas 16-QAM gives 4 bits per symbol, 64-QAM gives 6 bits per symbol, or 3/2, or 1.5, times as many as 16-QAM. The minimum allocation is 6 resource blocks, so the data rate is 672 kbit $s^{-1}$ × 1.5 × 6 = 6.048 Mbit $s^{-1}$

A user would not get the data rate calculated in the last example for two reasons. Firstly, not all the resource elements in a block can carry user data. Some are reserved for signalling and pilot data, which are needed by the system itself to ensure that it functions properly. Secondly, error control is used, which always results in some redundancy.

The allocation of resources to multiple users on demand is ‘multiple access’, and is a feature of mobile communications system. For example, although 3G used a very different system from OFDMA (called Wideband Code Division Multiple Access, or WCDMA), it too depends on a limited resource being shared out among multiple users.

The rapid review and revision of resource allocation as conditions change is a feature of mobile communications and known as scheduling.

## 4.6 WiFi

WiFi, unlike mobile communication, has very limited mobility management, so is not regarded as mobile communication. It also operates at much lower power levels than mobile, giving a reduced range, and operates in unlicensed spectrum as opposed to the licensed spectrum allocated to mobile operators for their exclusive use. Anyone can use unlicensed spectrum (subject to certain restrictions), so WiFi has no guarantee of quality or reliability. Table 4.1 summarises some of the more significant versions of WiFi in chronological order.

### Table 4.1 Some versions of 802.11

| **Standard** | **Year** | **Band** /GHz | **Channel (s) /MHz** | **Modulation method** | **Highest modulation order** | **Highest code rate** | **MIMO streams** | **Max transmission rate per MIMO stream /Mb s$^{-1}$** |
|---|---|---|---|---|---|---|---|---|
| 802.11a | 1999 | 2.4 | 22 | OFDM | 64 QAM | 3/4 | N/A | 54 |
| 802.11b | 1999 | 2.4 | 22 | CCK | QPSK | 1/2 | N/A | 11 |
| 802.11g | 2003 | 2.4 | 20 | OFDM | 64 QAM | 3/4 | N/A | 54 |
| 802.11n | 2009 | 2.4/5 | 20/40 | OFDM | 64 QAM | 5/6 | up to 4 streams total | 65<br>(in 20 MHz) |
| 802.11ad | 2012 | 60 | 2160 | OFDM | 64 QAM | 13/16 | N/A | 6756.75 |
| 802.11ac | 2014 | 5 | 20/40/ 80/160 | OFDM | 256 QAM | 5/6 | up to 8 streams | 180 (in 40 MHz; highest coding rate in 20 MHz = 3/4) |

The ‘modulation method’ column shows that OFDM is standard in all versions apart from 802.11b. Not only has OFDM become standard, but the specification of OFDM has remained consistent. Subchannel width, is 312.5 kHz across all WiFi versions except 802.11ad.

The maximum transmission rates shown in Table 2.3 need to be read in conjunction with the number of channels used, as the 802.11n and ac standards allow channels to be combined. Combining two adjacent 20 MHz channels to make a 40 MHz channel yields slightly more than double the benefit, because the guard band between the channels can be used for data. Similarly, an 80 MHz channel gives slightly more than double the benefit of a 40 MHz channel.

802.11n and ac can use Multiple Input Multiple Output (MIMO), which takes advantage of the multiple routes radio waves can take in a reflective environment (such as indoors) to enable multiple streams of data to be transmitted on the same frequency. Using MIMO increases the transmission rate in proportion to the number of spatial streams used. Table 2.3 gives maximum transmission rates for a single spatial stream.

802.11ad is clearly exceptional, both in the band it occupies (60 GHz) and in its channel width (2160 MHz). The appeal of the 60 GHz band lies in the large amount of licence-free spectrum available – hence the prospect of channels that are about 100 times wider than the 20 MHz channel width of, for example, 802.11g. With such wide channels, transmission rates of several gigabits per second become possible; thus 802.11ad is also known as ‘WiGig’ or gigabit-rate WiFi.

A basic principle of the 802.11 standards is backwards compatibility, whereby later variants are compatible with earlier variants. For example, a device equipped for 802.11ac – which is a 5 GHz-only standard – will ‘fall back’ to earlier 5 GHz standards, namely 802.11a and n if there is a WiFi access point using one of those earlier standards. In those circumstances an 802.11ac device will perform no better than a device designed for the earlier standards.

## 4.7 WiFi maximum transmission rate

The 20 MHz channel width of common versions of WiFi is divided into 64 OFDM sub-channels, each 312.5 kHz wide. However, not all 64 subchannels are used. For example in 802.11n the three subcarriers at the lower end of the channel and four subcarriers at the upper end are nulled (see Figure 4.12).

This results in power being transmitted across a band approximately 17.8 MHz wide within the 20 MHz band. This is done because OFDM is noted for the high level of power that spills out on either side of the transmission band. Restricting transmission to a 17.8 MHz band ensures that most transmitted power stays within the 20 MHz channel.

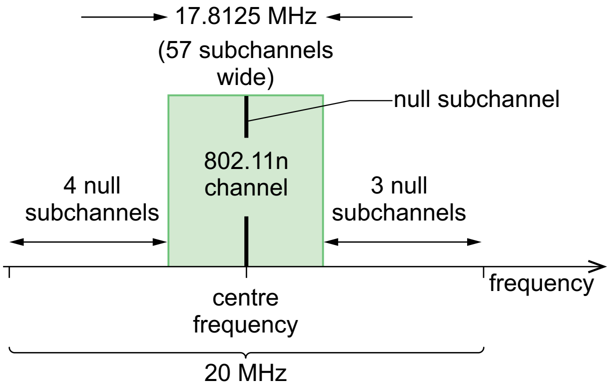



**Figure 4.12 Subchannel nulling used in an 802.11n channel**

The subcarrier on the central frequency is also nulled, which enables receivers to locate the centre of the transmission band and for other reasons related to the way subcarriers are modulated and demodulated.

The 8 nulled subchannels shown in Figure 4.12 leave 56 usable subchannels from the total of 64. Four of the 56 are used for control data, leaving 52 subchannels for user data. A symbol rate of $250 \times 10^3$ symbols per second per subchannel is standard across the common WiFi versions.

From this information the maximum transmission rate across all 52 subchannels of 802.11n to be calculated. The calculation is based on the use of the highest order of modulation, which is 64-QAM in 802.11n. This gives 6 bits per symbol. The number of bits per second across the 52 usable subchannels is:

$$52 \times (250 \times 10^3 \text{ symbols s}^{-1}) \times 6 \text{ bits symbol}^{-1} = 78 \text{ Mbit s}^{-1}$$

The highest code rate for 802.11n is 5/6. That is, only 5 bits in 6 are 'useful' data, the sixth bit being redundancy for error control. Hence the maximum transmission rate is:

$(5/6) \times 78 \text{ Mbit s}^{-1} = 65 \text{ Mbit s}^{-1}$

This is the figure given in Table 2.1. A rate of 600 Mbit $\text{s}^{-1}$ is usually claimed for 802.11n. This is based on the use of 40 MHz of spectrum (slightly more than doubling the 65 Mbit $\text{s}^{-1}$ calculated above) and four MIMO streams (quadrupling the transmission rate relative to a single stream).

## 4.8 WiFi multiple access

Although WiFi uses OFDM, making it similar in some respects to 4G, the method of providing multiple access to the radio channel is very different from 4G's. WiFi doesn't use resource blocks. Also, the two directions of data traffic (towards and away from the user) are not differentiated by frequency band. The following activity shows how multiple access is provided in WiFi.

### Activity 4.4 Animation

The following animation will explain how WiFi allocates access to the radio channel among its various users. Open it by clicking on the image or 'View' link below, then watch and listen to the six sections of the animation.

Interactive content is not available in this format.

## 4.9 Summary

Orthogonal frequency division multiplexing (OFDM), which divides a communication channel into narrow subchannels (each with its own subcarrier) provides a spectrally efficient and flexible way to use the channel. OFDM has become an almost ubiquitous part of modulation, both in wired and wireless communication.

The use of QAM modulation for the subcarriers of OFDM is also ubiquitous. Using various orders of QAM (depending on noise conditions) further adds to the flexibility of OFDM.

In DSL broadband, a version of OFDM called Digital Multitone (DMT) is used both because of its spectral efficiency and its adaptability to unpredictable and variable noise conditions.

In 4G, Orthogonal Frequency Division Multiple Access (OFDMA) is used as a flexible way to share access to a radio channel among multiple, transient users. This is done by grouping subchannels into units of resource which are allocated to users dynamically as fluctuating demand requires.

In WiFi, OFDM is exploited as a way of making very efficient use of the available channels. The subchannels are not used for access (as in 4G), but selective nulling of subchannels is used to ensure that transmission power is properly restricted. Multiple access in WiFi is based a 'listen before transmitting' protocol known as Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA).

# Conclusion

The development of communication technology is, as much as anything, a story of the strategies for coping with adversity. ‘Adversity’ in this context means, for example, noise and interference, which are unavoidable and which cause error. It also refers to the unpredictable and uncontrollable characteristics of many communication channels, such as telephone lines and radio environments. Strategies for dealing with adversity here include the use of compression to minimise the amount of data that needs to be transmitted, the use of subchannels which enable particular parts of the frequency spectrum to be treated independently of others, and the variable orders of QAM, which enable data throughput to be tailored to prevailing conditions.

All the topics in this free course, *Exploring communications technolgy*, and more, are covered in greater length and depth in the Open University course TM355 *Communications technology*, which can be studied as a stand-alone course or as part of the University’s Computing and IT BSc degree.

# Acknowledgements

This free course was written by Allan Jones, David Chapman, Helen Donelan, Laurence Dooley, and Adrian Poulton.

This free course includes adapted extracts from the course TM355 *Communications technology*. If you are interested in this subject and want to study formally with us, you may wish to explore other courses we offer in Technology.



**Don't miss out**

If reading this text has inspired you to learn more, you may be interested in joining the millions of people who discover our free learning resources and qualifications by visiting The Open University – www.open.edu/openlearn/free-courses.