1 Introduction to mobile telecommunications

Mobile phones were first introduced in the early 1980s. In the succeeding years, the underlying technology has gone through three phases, known as generations. The first generation (1G) phones used analogue communication techniques: they were bulky and expensive, and were regarded as luxury items. Mobile phones only became widely used from the mid 1990s, with the introduction of second generation (2G) technologies such as the *Global System for Mobile Communications* (GSM). These use more powerful digital communication techniques, which have allowed their cost to plummet, and have also allowed them to provide a wider range of services than before. Examples include text messaging, email and basic access to the Internet.

Third generation (3G) phones still use digital communications, but they send and receive their signals in a very different way from their predecessors. This allows them to support much higher data rates than before, and hence to provide more demanding services such as video calls and high speed Internet access. This book is about the most popular third generation technology, the *Universal Mobile Telecommunication System* (UMTS).

The first chapter lays the foundations for the subjects covered later in the book. It begins by briefly describing the architecture of a mobile telecommunication system, and continues with a more detailed look at two important aspects of its operation: the communication protocols that manage the delivery of information to and from a mobile phone, and the special techniques that are used for radio transmission and reception. It concludes with the history of mobile telecommunications, and a look at the current state of the market.

1.1 Architecture of a mobile telecommunication system

Figure 1.1 shows the architecture of a mobile telecommunication system. The system is controlled by a particular network operator such

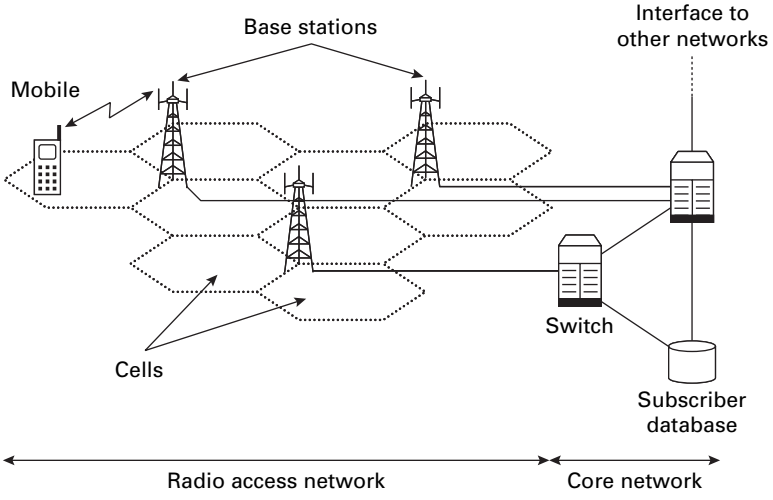

Figure 1.1 Simplified architecture of a mobile telecommunication system.

as Vodafone or O2, and is often known as a *public land mobile network* (PLMN). It has three main components: the core network, the radio access network and the mobile phone.

The *core network* has a similar role to a traditional fixed line telephone network. It sends information like voice calls or text messages from one phone to another using components that are known as *switches*. It also maintains a database containing information about the network operator's subscribers, and uses the database for tasks like preparing and distributing bills. Finally, it has a number of functions that are not required in a fixed line network; for example, it monitors the locations of the mobile phones, so that it can continue sending information to them as they move around.

The *radio access network* handles the radio communications between the core network and the mobile phone. It contains a large number of *base stations*, each of which transmits and receives radio signals to and from the mobile phones in the surrounding area. The area around a base station is often divided into multiple *sectors* by equipping the base station with multiple directional antennas, each of which communicates with the mobile phones in the corresponding sector.

If this is done, then the number of sectors per base station is usually either three (as shown in the figure) or two.

The most common word for a part of the radio access network is *cell*. Unfortunately this word is ambiguous: it can refer either to a single sector, or to the group of sectors that a base station controls. We will use the first definition throughout this book, so that the words 'cell' and 'sector' will mean exactly the same thing.

Each cell has a maximum size, which is determined by the greatest distance at which the radio signals can be successfully received. It also has a maximum capacity, which limits the number of mobile phones that can make calls within the cell at the same time. In rural areas, the population densities are low, so capacity is not a problem. The cells are therefore large, typically a few kilometres across, and are known as *macrocells*. In urban areas, the population densities are much greater. To handle this, we can introduce an extra set of *microcells*: these are only a few hundred metres across, so they greatly increase the total capacity of the network. The original macrocells are usually retained, as they are useful for fast-moving users who move quickly from one cell to another. We can also use a third set of *picocells*: these are a few tens of metres across, and provide small-scale coverage in offices, shopping centres or the home.

The use of cells is a crucial part of the system: it allows the same radio frequencies to be used in different locations with little interference, which greatly increases the number of mobile phones that can be supported. For this reason, the system is often known as a *mobile cellular network*.

The user's device was traditionally known as a mobile phone but, with the increased use of data communications like text messaging and email, this terminology has become rather restrictive. In UMTS, the device is officially known as the *user equipment* (UE); in this book, we normally use the simple term *mobile*. The interface between the radio access network and the mobile is known as the *air interface* or the *radio interface*. On this interface, the path from the network to the mobile is known as the *downlink* (DL) or *forward link*, and the path from mobile to network is the *uplink* (UL) or *reverse link*.

When a mobile moves from one cell to another, it has to stop communicating with its current cell and start communicating with the next

cell along. This process is known as a *handover*, and is controlled by signalling messages between the mobile and the network. A mobile can also move outside the region covered by its own network operator, for example when travelling to another country. The mobile can still make calls by using resources in two networks: the base stations in the *visited network*, the user database in the *home network*, and switches in both. This situation is known as *roaming*.

There are several books with more information about mobile telecommunication systems: reference [1] is an excellent example. In the next couple of sections, we will examine two aspects of the system in more detail: the communication software that transfers data and signalling messages across the network, and the techniques that are used for reliable transmission and reception over the air interface.

1.2 Communication networks

The role of a communication network is to connect devices like computers and phones to each other so that they can exchange data and signalling messages. The network has to accept information from a transmitting device, identify a route to the receiver, and send the information there without introducing any significant errors. Some example communication networks include fixed line telephone systems, the Internet, and the mobile communication network that we introduced in Figure 1.1.

This section is an introduction to the tasks that communication networks carry out, with some examples of the communication software that they use. There are many books that give detailed accounts, such as references [2], [3] and [4].

1.2.1 Circuit switching and packet switching

Communication networks can transport information using two very different techniques, which will both be important for this book: circuit switching and packet switching. The two techniques are illustrated in Figure 1.2. Circuit switching is generally used for voice calls, and packet switching for data.

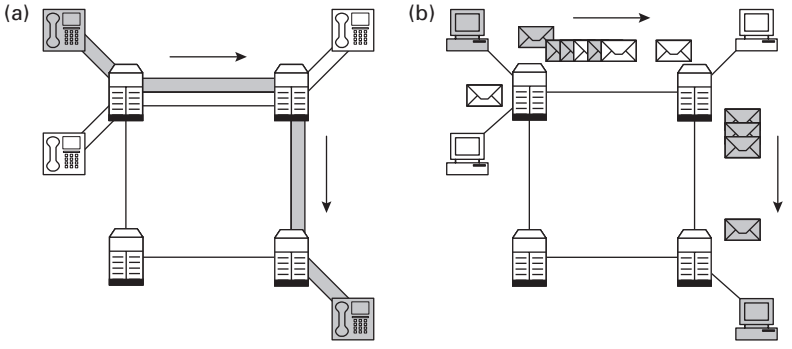

Figure 1.2 Illustration of the two main transport mechanisms in a communication network. (a) Circuit switching. (b) Packet switching.

Circuit switched (CS) networks (Figure 1.2a) use the same techniques as a traditional fixed line telephone system. At the start of a call, the network identifies a route through the switches that connect the two phones, and reserves enough resources on that route to handle the call. For example, a voice call typically requires a constant data rate of 64 000 bits per second (64 kbps). By reserving enough resources in the switches and the intervening links for transmission at 64 kbps, we can ensure that the information travels from end to end with a very low delay and with no obstruction from other calls.

Circuit switching has a big disadvantage, however: it is rather inefficient. In a phone call, each user is only speaking for half the time on average, so we have already set aside twice the resource that we actually need. The situation is worse when doing data transfers such as web browsing because these are typically very bursty, comprising short periods of activity separated by long periods when nothing is happening.

To deal with this problem, *packet switched* (PS) networks like the Internet use a different technique (Figure 1.2b). In this technique, the transmitter divides the data stream into blocks that are known as packets. It adds some extra information, known as a header, to each packet, which tells the network how the packet should be routed. It then sends each successive packet to the first switch in the network. When a packet reaches a switch, the switch looks up the packet's routing information in a routing table, reads the identity of the next switch in

the route, and forwards it there. This process is repeated switch by switch, so that eventually the packet reaches its destination.

The packets can be routed in two different ways. In the *virtual circuit* approach, each packet is labelled with a virtual circuit number that identifies the data stream. The routing table lists the virtual circuits that a particular switch is using, and identifies the next switch in the route for each one. *Permanent virtual circuits* last indefinitely, while *temporary virtual circuits* are set up for the duration of a single data stream. Temporary virtual circuits are more flexible, but they require some initial signalling at the time the data stream is set up, to identify a route through the network and set up the routing tables. The *datagram* approach is rather different. In this approach, each packet is labelled with the address of the destination device. The routing table lists all the destination addresses that the switch might have to use, and maps each destination address onto the identity of the next switch in the route. The datagram approach is the more popular of the two (it is used for routing in the Internet, for example), but we will see both approaches in this book.

Packet switching is more efficient than circuit switching, because if a source stops transmitting, then its resources are immediately available for other data streams. It has a disadvantage, however: if several sources decide to transmit at once, then their total data rate can exceed the capacity of the intervening links, and the network can become congested. If this happens, then the packets are held in queues in the network's switches, which leads to delays.

Because packet switched networks are more efficient than circuit switched ones, there is currently a trend among telecommunication operators to work round the problems noted above, and to introduce packet switched transport for all services, voice as well as data. (The use of packet switched networks for voice calls is often known as *voice over internet protocol* (VoIP).) We will see this trend reflected at various points in the book.

1.2.2 Communication protocols

Routing is just one of the functions of a communication network. Other functions include controlling the electrical signals on each interface,

encrypting the information if it has to be transmitted securely, and possibly retransmitting the information if an error occurs. To keep these functions separate, each of them is handled by a software component known as a *protocol*, and the individual protocols are arranged into a *stack* that has several different *layers*. In the transmitter, the information is processed first by the higher layer protocols and then by the lower layer ones, before sending it into the communication network. The process is reversed in the receiver, to recover the original information.

There are different ways to arrange the layers in a protocol stack, but the most common is the seven-layer OSI (*open systems interconnection*) model shown in Figure 1.3. The figure just shows the processes in the transmitter and the receiver: we will cover what happens inside the network in a few moments. The stack will be described by reference to a packet switched network, although many of the issues apply to a circuit switched network as well.

Above the protocol stack, the application software is something like a web browser or an email client. The *application layer* (layer 7) acts as an

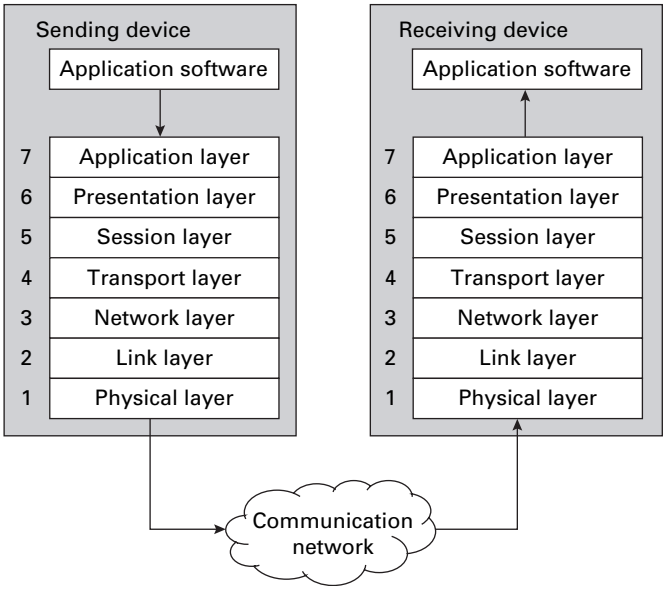

Figure 1.3 Organisation of communication protocols in the OSI protocol stack.

interface between the application and the lower layer protocols, by providing software functions for tasks such as setting up a data stream and sending a data packet. Some well-known application layer protocols are the *hypertext transfer protocol* (HTTP) and the *simple mail transfer protocol* (SMTP), which handle web pages and emails respectively. The *presentation layer* (6) represents the information being exchanged between the two end devices using a common syntax that both can understand, while the *session layer* (5) sets up, manages and tears down the communication path between them. Layers 5 and 6 are less important than the others, and we will not consider them further.

The *transport layer* (4) manages end-to-end transfers from the transmitter to the receiver, without worrying about the intervening route. There are two main types of transport layer protocol. Connection-oriented protocols, like the *transmission control protocol* (TCP), use signalling communications between the transmitter and receiver as well as the actual data transfer. This brings a number of benefits, for example it allows the receiver to request retransmissions of data that have arrived incorrectly. However, it also slows the data transfer down. Connectionless protocols, like the *user datagram protocol* (UDP), just send data to the receiver without any extra signalling. They are suitable for information like streaming video, for which timely arrival is more important than perfect accuracy.

The *network layer* (3) ensures that data are sent on the correct route from transmitter to receiver. The network layer protocol used on the Internet is the *Internet protocol* (IP), which uses the datagram approach and carries out routing using the IP address of the destination device. The *link layer* (2) sends data on a single link from one switch to another. Like the transport layer, the link layer can be connection-oriented or connectionless: the difference is that any layer 2 retransmissions are on a link-by-link basis, while layer 4 retransmissions are made end-to-end. Two common link layer protocols are *Ethernet* and the *point-to-point protocol* (PPP). The link layer also manages the underlying *physical layer* (1): this transmits and receives the actual signals, using a transmission medium such as copper wire, optical fibre or radio.

We can think of the interactions between different layers in two ways. These are shown in Figure 1.4, using the link layer as an example. The

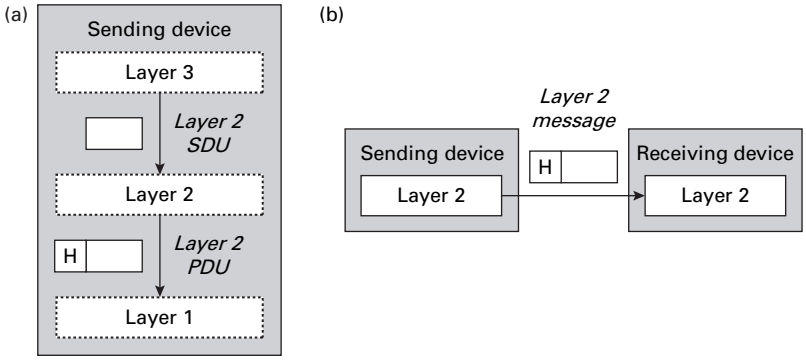

Figure 1.4 Illustration of how data packets are transferred between communication protocols. (a) Transfer between the protocol layers in a single device. (b) Transfer between two different devices.

first way is to think of the vertical interactions inside a single device (Figure 1.4a). In the transmitter, layer 3 sends a packet to layer 2, which is known as a layer 2 *service data unit* (SDU). Layer 2 processes the packet and adds a header, denoted H in the figure. (The processing might include link-by-link encryption, for example, in which case the header might indicate that encryption has been used.) It then sends the new packet down to layer 1 for further processing. The new packet is known as a layer 2 *protocol data unit* (PDU), although it immediately becomes a layer 1 SDU. In the receiver, layer 2 receives a PDU from layer 1. It detaches and inspects the header, undoes the effect of the transmit processing (here by decryption), and passes the resultant SDU up to layer 3.

The second way (Figure 1.4b) is to think of the horizontal interactions between devices. From this point of view, the transmitter's link layer sends a message to the receiver's link layer, which contains the header and the processed data. It uses the layer 1 protocol to do this, but the details of that protocol are hidden from the link layer and can be thought of as a black box. The effect is that the details of each layer can be isolated from the other layers in the protocol stack.

What happens inside an individual switch? A typical answer is shown in Figure 1.5. In this figure, the switch is receiving packets on an Ethernet link from the source device, and has to send them to the

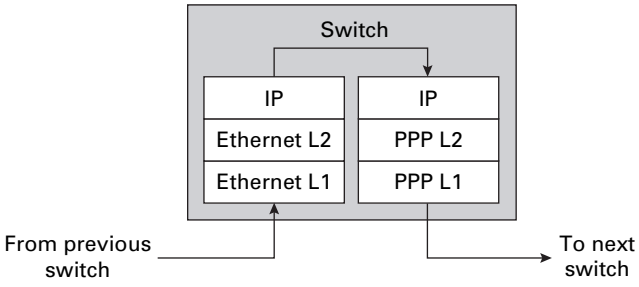

Figure 1.5 Example operation of the layer 1, 2 and 3 protocols inside a switch.

destination device using PPP. To do this, it unwraps the received packets as far as the layer 3 PDU, reads the routing information there, and makes any changes that are needed to the layer 3 header. It then wraps the packets up again using PPP, and sends them on the correct route to the destination device. In a packet switched network, a layer 3 switch like the one in the figure is often known as a *router*.

In a complex system like UMTS, the individual layers are often subdivided into more than one protocol, each of which handles one aspect of that layer's functions. In addition, UMTS needs a lot of extra signalling to carry out tasks such as roaming and handover. These signalling functions are often handled by a separate protocol stack, which we can think of as an extra version of Figure 1.3. If this is done, then the signalling functions are collectively known as the *control plane*, while the data transfer part is known as the *user plane*.

1.2.3 Example communication protocols

To illustrate the points discussed above, Figure 1.6 shows three protocol stacks that we will use later on in the book: the Internet protocol stack, ATM and SS7.

The Internet protocol stack uses several protocols that we have already introduced, such as TCP, UDP and IP. It carries out routing by datagram-based packet switching, while layers 1 and 2 can use any mechanism at all, such as Ethernet.