Multimedia Over Wireless and Mobile Networks

17

The rapid developments in computer and communication technologies have made *ubiquitous computing* a reality. From cordless phones in the early days to later cellular phones, wireless mobile communication has been the core technology that enables anywhere and anytime information access and sharing. The new generation of smart mobile devices that emerged only in the recent years are driving the revolution further. Multimedia over wireless and mobile networks share many similarities as over the wired Internet; yet the unique characteristics of wireless channels and the frequent movement of users also pose new challenges that must be addressed.

17.1 Characteristics of Wireless Channels

Wireless radio transmission channels are far more error-prone than wire-line communications are. In this section, we briefly present the most common radio channel models to gain insight into the cause of errors and to classify the types of bit errors, the amount, and their patterns. More details can be found in [1,27,28].

Various effects cause radio signal degradation in the receiver side. They can be classified as short-range and long-range effects. Accordingly, path loss models are available for long-range atmospheric attenuation channels, and fading models are available for short-range degradation.

17.1.1 Path Loss

For long-range communication, the signal loss is dominated by atmospheric attenuation. Depending on the frequency, radio waves can penetrate the ionosphere (>3 GHz) and establish *line-of-sight* (LOS) communication, or for lower frequencies reflect off the ionosphere and the ground, or travel along the ionosphere to the receiver. Frequencies over 3 GHz (which are necessary for satellite transmissions to

Z.-N. Li et al., *Fundamentals of Multimedia*,
Texts in Computer Science, DOI: 10.1007/978-3-319-05290-8_17,

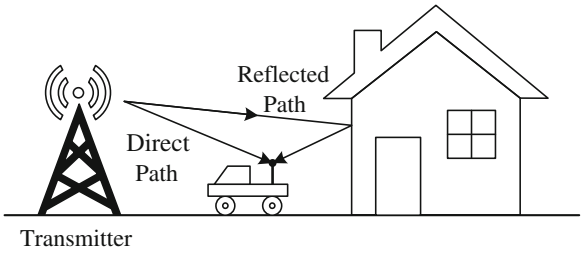


Fig. 17.1 An example of multipath

penetrate the ionosphere) experience gaseous attenuations, influenced primarily by oxygen and water (vapor or rain).

The free-space attenuation model for LOS transmission is in inverse proportion to the square of distance (d^2) and is given by the Friis radiation equation

$$S_r = \frac{S_t G_t G_r \lambda^2}{(4\pi^2) d^2 L} \tag{17.1}$$

S_r and S_t are the received and transmitted signal power, G_r and G_t are the antenna gain factors, λ is the signal wavelength, and L is the receiver loss. It can be shown that if we assume ground reflection, attenuation increases to be proportional to d^4.

Another popular medium-scale (urban city size) model is the *Hata model*, which is empirically derived based on the Okumura path loss data in Tokyo. The basic form of the path loss equation in dB is given by

$$L = A + B \cdot \log_{10}(d) + C. \tag{17.2}$$

Here, A is a function of the frequency and antenna heights, B is an environment function, and C is a function depending on the carrier frequency. Again, d is the distance from the transmitter to the receiver.

For satellite communications that are mainly affected by rain, meteorological rainfall density maps can be used to communicate with the region. Attenuation is computed according to the amount of rainfall in the area on the given date.

17.1.2 Multipath Fading

Fading is a common phenomenon in wireless (and especially mobile) communications, in which the received signal power (suddenly) drops [1]. Signal fading occurs due to reflection, refraction, scattering, and diffraction (mainly from moving objects), as illustrated in Fig. 17.1. *Multipath fading* occurs when a signal reaches the receiver via multiple paths (some of them bouncing off buildings, hills, and other objects). Because they arrive at different times and phases, the multiple instances of the signal can cancel each other, causing the loss of signal or connection. The problem becomes more severe when higher data rates are explored.

For indoor channels, the radio signal power is generally lower, and there are more objects in a small place; some are moving. Hence, multipath fading is the main

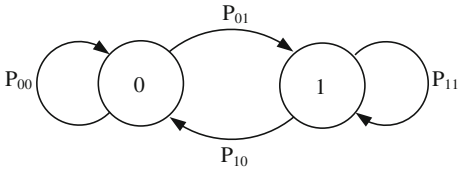


Fig. 17.2 The Gilbert-Elliott two-state Markov chain model. State 0: *Good*; State 1: *Bad*

factor for signal degradation. In outdoor environments, refraction, diffraction, and scattering effects are also the important causes of signal degradation, mostly by the ground and buildings.

A multipath model probabilistically states the received signal amplitude, which varies according to whether the signals superimposed at the receiver are added destructively or constructively. The *Doppler spread* of a signal is defined as the distribution of the signal power over the frequency spectrum (the signal is modulated at a specific frequency bandwidth). When the Doppler spread of the signal is small enough, the signal is coherent—that is, there is only one distinguishable signal at the receiver. This is typically the case for narrowband signals. When the signal is wideband, different frequencies of the signal have different fading paths, and a few distinguishable signal paths are observed at the receiver, separated in time.

For narrowband signals, the most popular models are *Rayleigh fading* and *Rician fading*. The Rayleigh fading model assumes an infinite number of signal paths with *non line-of-sight* (NLOS) to the receiver for modeling the probability density function P_r of received signal amplitude r:

$$P_r(r) = \frac{r}{\sigma^2} \cdot e^{\frac{-r^2}{2\sigma^2}} \tag{17.3}$$

where σ is the standard deviation of the probability density function. Although the number of signal paths is typically not too large, the Rayleigh model does provide a good approximation when the number of paths is over 5.

A Rayleigh fading channel can be approximated using a Markov process with a finite number of states, referred to as a *Finite State Markov Channel* [2]. The simplest form, known as the Gilbert-Elliott model [3], is with only two states, representing the *good* and the *bad* channel conditions. As illustrated in Fig. 17.2, state 0 has no error and state 1 is erroneous, and the wireless channel condition switches between them with transition probabilities P_{00} to P_{11}. It captures the short-term bursty nature of wireless errors, and has been widely used in simulations.

A more general model that assumes LOS is the Rician model. It defines a *K-factor* as a ratio of the signal power to the scattered power—that is, K is the factor by which the LOS signal is greater than the other paths. The Rician probability density function P_c is

$$P_c(r) = \frac{r}{\sigma^2} \cdot e^{\frac{-r^2}{2\sigma^2} - K} \cdot I_o\left(\frac{r}{\sigma}\sqrt{2K}\right), \quad \text{where } K = \frac{s^2}{2\sigma^2} \tag{17.4}$$

As before, r and σ are the signal amplitude and standard deviation, respectively, and s is the LOS signal power. I_o is a modified Bessel function of the first kind with 0 order. Note that when $s = 0$ ($K = 0$), there is no LOS, and the model thus reduces

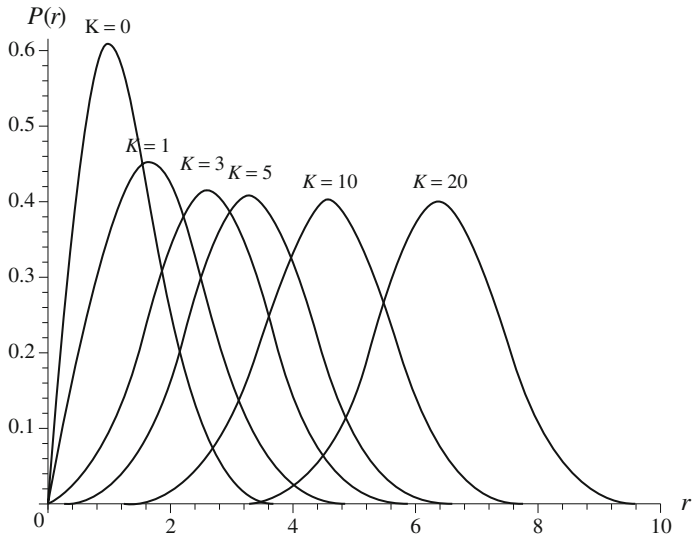


Fig. 17.3 Rician PDF plot with K-factor = 0, 1, 3, 5, 10, and 20

to a Rayleigh distribution. When $K = \infty$, the model reflects the popular *additive white Gaussian noise* (AWGN) conditions. Figure 17.3 shows the Rician probability density function for K-factors of 0, 1, 3, 5, 10, and 20, respectively, with a standard deviation of $\sigma = 1.0$.

For a wideband signal, the fading paths are more empirically driven. One way is to model the amplitude as a summation over all the paths, each having randomized fading. The number of paths can be 7 for a closed-room environment (six walls and LOS) or a larger number for other environments. An alternative technique of modeling the channel fading is by measuring the channel impulse response.

A similar technique is to use *rake receivers*, through which multiple radio receivers are tuned to signals with different phases and amplitudes, to recompose the transmission that is split to different distinguishable paths. The signal at each rake receiver is added up to achieve better SNR. To tune the rake receivers to the proper fading paths, a special *pilot channel* sends a well-known pilot signal, and the rake receivers are adjusted to recognize that symbol on each fading path.

17.2 Wireless Networking Technologies

Like wired networks, there is a large family of wireless networks, using different technologies to combat fading and pass loss and covering geographical areas of different sizes. In a wide-area cellular network, a field is covered by a number of *cells*. Each mobile terminal in a cell contacts its *Access Point (AP)* or *Base Station* (BS),

which serves as a gateway to the network. The AP themselves are connected through high-speed wired lines, or wireless networks or satellites that form the backbone network. When a mobile user moves out of the range of the current AP, a *handoff* (or *handover*, as it is called in Europe) is required to maintain the communication. The size of a cell is typically of 1,000 m in cities, but can be larger (macrocell) or smaller (microcell) depending on the location and the density of users. The whole network of cells collectively cover a city- or nation-wide area or even beyond, ensuring anywhere connection.

A *Wireless Local Area Network* (WLAN), on the other hand, covers a much shorter range, generally within 100 m. Given the short distances, the bandwidth can be very high while the access cost and power consumption can be low, making them ideal for use within a house or an office building. Many modern home entertainment systems are built around WLANs. Public WLAN accesses have also been offered by many airports, shops, restaurants, or even city-wide.

In this section, we provide an overview of the different generations of wireless cellular networks and wireless local area networks.

17.2.1 1G Cellular Analog Wireless Networks

The very early wireless communication networks were used mostly for voice communications, such as telephone and voice mail. The first-generation (1G) cellular phones used an analog technology with *Frequency Division Multiple Access* (FDMA), in which each user is assigned a separate frequency channel during the communication. Its standards were *Advanced Mobile Phone System* (AMPS) in North America, *Total Access Communication System* (TACS), and *Nordic Mobile Telephony* (NMT) in Europe and Asia, respectively. Digital data transmission users needed modems to access the network; the typical data rate was 9,600 bps.

AMPS, for example, operates at the 800–900 MHz frequency band. Each direction of the two-way communication is allocated 25 MHz, with *mobile station transmit* (MS transmit) in the band of 824–849 MHz and *base station transmit* (BS transmit) in the band of 869–894 MHz. Each of the 25 MHz bands is then divided up for two operator bands, A and B, giving each 12.5 MHz. FDMA further divides each of the 12.5 MHz operator bands into 416 channels, which results in each channel having a bandwidth of 30 kHz. The frequency of any MS transmit channel is always 45 MHz below the frequency of the corresponding BS transmit channel in communication.

Similarly, TACS operates at the 900 MHz frequency band. It carries up to 1,320 full-duplex channels, with a channel spacing of 25 kHz.

Figure 17.4 illustrates a sample geometric layout for an FDMA cellular system. A cluster of seven hexagon cells can be defined for the covered cellular area. As long as each cell in a cluster is assigned a unique set of frequency channels, interference from neighboring cells will be negligible. For clarity, the cells from the first cluster are marked with thicker borders.

The same set of frequency channels (denoted f_1 to f_7 in Fig. 17.4) will be reused once in each cluster, following the illustrated symmetric pattern. The so called

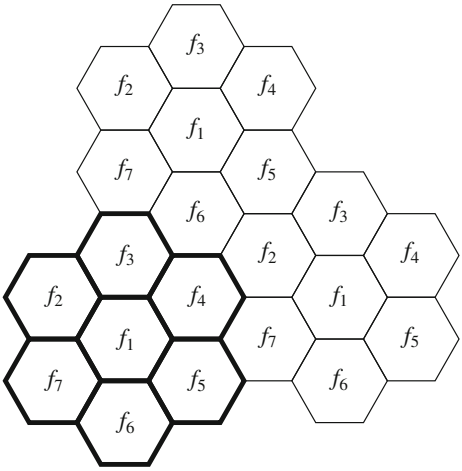


Fig. 17.4 An example of geometric layout for an FDMA cellular system with a cluster size of seven hexagon cells

reuse factor is $K = 7$. In the AMPS system, for example, the maximum number of channels (including control channels) available in each cell is reduced to $416/K = 416/7 \approx 59$.

In this configuration, the users in two different clusters using the same frequency f_n are guaranteed to be more than D apart geographically, where D is the diameter of the hexagonal cell. In a vacuum, electromagnetic signals decay at a rate of D^{-2} over a distance D. In real physical spaces on the earth, the decay is consistently measured at a much faster rate of $D^{-3.5}$ to D^{-5}. As such, the interference by users of the same frequency channel from other groups becomes insignificant.

17.2.2 2G Cellular Networks: GSM and Narrowband CDMA

Besides voice, digital data was increasingly transmitted for applications such as text messaging, streaming audio, and electronic publishing. Starting from the second-generation (2G) wireless networks, digital technologies had replaced the analog technologies. The digital cellular networks adopted two competing technologies since 1993: *Time Division Multiple Access* (TDMA) and *Code Division Multiple Access* (CDMA). The *Global System for Mobile communications* (GSM) [4], which was based on TDMA, is the most widely used worldwide.

TDMA and GSM

As the name suggests, TDMA creates multiple channels in multiple time slots while allowing them to share the same carrier frequency. In practice, TDMA is generally combined with FDMA —that is, the entire allocated spectrum is first divided into

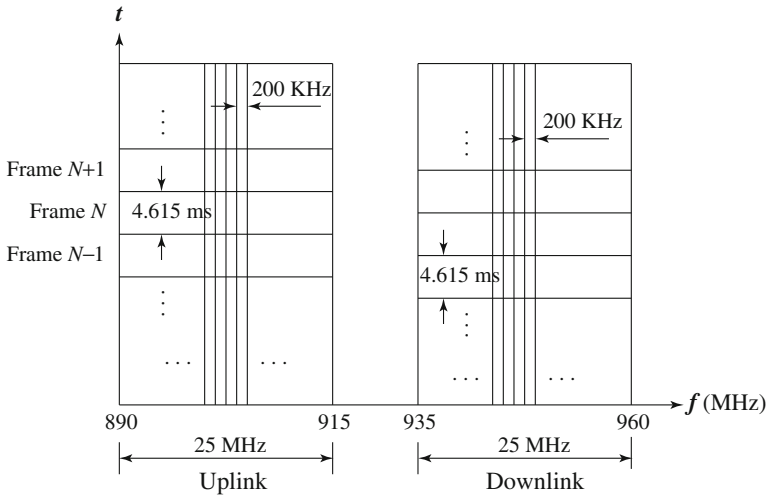


Fig. 17.5 Frequency and time divisions in GSM

multiple carrier frequency channels, each of which is further divided in the time dimension by TDMA.

GSM was established by the *European Conference of Postal and Telecommunications Administrations* (CEPT) in 1982, with the objective of creating a standard for a mobile communication network capable of handling millions of subscribers and providing roaming services throughout Europe. It was designed to operate in the 900 MHz frequency range and was accordingly named GSM 900. Europe also supported GSM 1800, which is the original GSM standard modified to operate at the 1.8 GHz frequency range.

In North America, the GSM network uses frequencies in the range of 1.9 GHz (GSM 1900).

As Fig. 17.5 shows, the uplink (mobile station to base station) of GSM 900 uses the 890–915 MHz band, and the downlink (BS to mobile station) uses 935–960 MHz. That is, each is allocated 25 MHz. The frequency division in GSM divides each 25 MHz into 124 carrier frequencies, each with a separation of 200 kHz. The time division in GSM then divides each carrier frequency into TDMA frames; 26 TDMA frames are grouped into a *Traffic Channel* (TCH) of 120 ms that carries voice and data traffic.

Each TDMA frame is approximately 4.615 ms (i.e., 120/26 ms) and consists of eight time slots of length $4.615/8 \approx 0.577$ ms. Each mobile station is given unique time slots during which it can send and receive data. The sending/receiving does not occur at the same time slot, but are separated by three slots.

GSM provides a variety of data services, through sending and receiving data to users on POTS, ISDN, and packet-switched or circuit-switched public data networks. It also supports a *Short Message Service* (SMS), in which text messages up to 160

characters can be delivered to (and from) mobile phones. Another feature is the adoption of the *subscriber identity module* (SIM), a smart card that carries the mobile user's personal number and enables ubiquitous access to cellular services. SIM is used in later generations of cellular networks too and is available virtually in all today's cellphones.

By default, the GSM network is circuit switched, and its data rate is limited to 9.6 kbps (kilobits per second), which is hardly useful for general data services (certainly not for multimedia data). *General Packet Radio Service* (GPRS), developed in 1999, supports packet-switched data over GSM wireless connections, so users are "always connected." It is also referred to as one of the 2.5G (between second- and third-generation) services. The theoretical maximum speed of GPRS is 171.2 kbps when all eight TDMA time slots are taken by a single user. In real implementations, the single-user throughput reached 56 kbps in year 2001. Apparently, when the network is shared by multiple users, the maximum data rate for each GPRS user will drop.

Preliminary multimedia content exchange was supported by GPRS through the *Multimedia Messaging Service* (MMS). It extends the basic SMS in GSM that allows exchange of text messages only up to 160 characters in length. To send a multimedia content, the sending device first encodes it with an *MMS Message Encapsulation Specification*. The encoded message is forwarded to the carrier's MMS store and to the forward server, known as the *Multimedia Messaging Service Centre* (MMSC). Once the MMSC receives the message, it first determines whether the receiver's handset is MMS-capable. If so, the content is extracted and sent to a temporary storage server with an HTTP front-end. An SMS `control message` containing the URL of the content is then sent to the recipient's handset to trigger the receiver's browser to open and receive the content from the embedded URL. Given the limited data rate of GPRS, the multimedia content is generally downloaded and then played, while not through real-time streaming.

Code Division Multiple Access

Code Division Multiple Access (CDMA) [5] is a major breakthrough in wireless communications. It is a *spread spectrum* technology, in which the bandwidth of a signal is spread before transmission. In its appearance, the spread signal might be indistinguishable from background noise, and so it has distinct advantages of being secure and robust against intentional interference (known as *jamming*). Spread spectrum is applicable to digital as well as analog signals, because both can be modulated and "spread." The earlier generation of cordless phones and cellular phones, for example, used analog signals. However, it is the digital applications, in particular CDMA, that made the technology popular in modern wireless data networks.

The foundation of CDMA is *Direct Sequence (DS) spread spectrum*. Unlike FDMA, in which each user is supposed to occupy a unique frequency band at any moment, multiple CDMA users can make use of the same (and full) bandwidth of the shared wideband channel during the entire period of transmission! A common

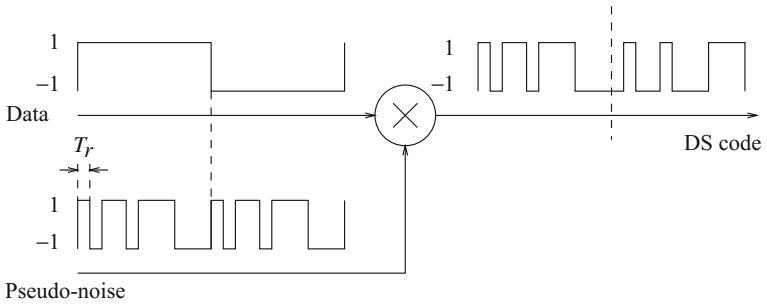


Fig. 17.6 Spreading in DS spread spectrum

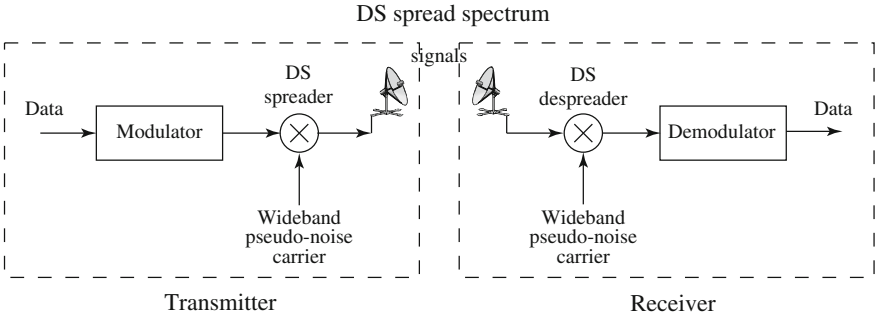


Fig. 17.7 Transmitter and Receiver of DS spread spectrum

frequency band can also be allocated to multiple users in all cells—in other words, providing a reuse factor of $K = 1$. This has the potential to greatly increase the maximum number of users, as long as the interference from them is manageable.

As Fig. 17.6 shows, for each CDMA transmitter a unique *spreading code* is assigned to a DS spreader. The spreading code (also called *chip code*) consists of a stream of narrow pulses called *chips*, with a bit width of T_r. Its bandwidth B_r is on the order of $1/T_r$.

The spreading code is multiplied with the input data by the DS spreader. When the data bit is 1, the output DS code is identical to the spreading code, and when the data bit is 0 (represented by -1), the output DS code is the inverted spreading code. As a result, the spectrum of the original narrowband data is spread, and the bandwidth of the DS signal is

$$B_{DS} = B_r. \tag{17.5}$$

The despreading process involves taking the product of the DS code and the spreading sequence. As long as the same sequence is used as in the spreader, the resulting signal is the same as the original data.

To separate the receivers for multiple access, i.e., CDMA, *orthogonal codes* can be used. As an example, consider spreading codes for two receivers: $(1,-1,-1,1)$ and $(-1,1,1,-1)$, which are orthogonal to each other (in practice, the code length

can be much longer); that is, their inner product is zero. Assume the data bit for receiver 1 is x and that for receiver 2 is y. The output DS code for receiver 1 is $x \cdot (1, -1, -1, 1)$ and that for receiver 2 is $y \cdot (-1, 1, 1, -1)$. The sender combines them together, sending $x \cdot (1, -1, -1, 1) + y \cdot (-1, 1, 1, -1)$. The decoding results at receivers 1 and 2 using their respective codes will be

$$(x \cdot (1, -1, -1, 1) + y \cdot (-1, 1, 1, -1)) \cdot (1, -1, -1, 1) = 4x \quad \text{(at receiver 1)}$$

$$(x \cdot (1, -1, -1, 1) + y \cdot (-1, 1, 1, -1)) \cdot (-1, 1, 1, -1) = 4y \quad \text{(at receiver 2)}$$

which, after normalization by 4 (the spreading code length), become x and y for receivers 1 and 2, respectively. In other words, there is no interference between them.

Because T_r is small, B_r is much wider than the bandwidth B_b of the narrowband signal.

In practice, to support more users and achieve better spectrum utilization, non-orthogonal *Pseudo-random Noise* (PN) sequences can be used as codes. This is based on the observation that in general not all users are active in a cell. Since the effective noise is the sum of all other users' signals, as long as an adequate level of "average case" interference is maintained, the quality of the CDMA reception is guaranteed. Such a *soft capacity* makes CDMA much more flexible than TDMA or FDMA with hard capacity only, accommodating more users when necessary and alleviating the undesirable dropping of ongoing calls when reaching the capacity limit.

17.2.3 3G Cellular Networks: Wideband CDMA

The 2G cellular networks were mainly designed for voice communications with circuit switching and had very limited support for Internet data access, not to mention multimedia services. Starting from the third generation (3G), multimedia services have become the core issues for the cellular network development. Applications include continuous media on demand, mobile interactive video call, remote medical service, and so on.

GPRS is considered as the first major step in the evolution of GSM networks toward 3G, which started to support the *Multimedia Messaging Service* (MMS), albeit with cumbersome operations. GPRS networks evolved to the *Enhanced Data rates for GSM Evolution* (EDGE) networks with enhanced modulation. It is a backward-compatible digital mobile phone technology that allows improved data transmission rates, as an extension over the standard GSM, known as 2.75G. Yet its support for multimedia remains very limited.

The 3G standardization process started in 1998, when the ITU called for Radio Transmission Technology (RTT) proposals for *International Mobile Telecommunication-2000* (IMT-2000). Since then, the project has been known as 3G or *Universal Mobile Telecommunications System* (UMTS).

While a large number of 2G wireless networks used TDMA/GSM and some CDMA, the 3G wireless networks have been predominantly using *Wideband CDMA*

(WCDMA). The key differences in WCDMA air interface from a narrowband CDMA air interface are

- To support bitrates up to 2 Mbps, a wider channel bandwidth is allocated. The WCDMA channel bandwidth is 5 MHz, as opposed to 1.25 MHz for IS-95 and other earlier standards.
- To effectively use the 5 MHz bandwidth, longer spreading codes at higher chip rates are used. The chip rate specified is 3.84 Mcps, as opposed to 1.2288 Mcps.
- WCDMA supports variable bitrates, from 8 kbps up to 2 Mbps. This is achieved using variable-length spreading codes and time frames of 10 ms, at which the user data rate remains constant but can change from one frame to the other—hence bandwidth on demand.

To achieve global standardization, the *Third Generation Partnership Project* (3GPP) was established in late 1998 to specify a global standard for the WCDMA technology, which was named *Universal Terrestrial Radio Access* (UTRA). At the same time the Telecommunication Industry Association (TIA), with major industry support, had been developing the *cdma2000* air interface recommendation for ITU. As similar work was going on in Asia, following the 3GPP example; the standards organizations decided to form a second forum called *Third Generation Partnership Project 2* (3GPP2).

The 3GPP and 3GPP2 forums, despite having some similarities in WCDMA air interface proposals, still proposed competing standards. However, in the interest of creating a global standard, the two forums are monitoring each other's progress and support recommendations by the operators harmonization group. The harmonized standard, referred to as *global 3G* (G3G), has three modes: Direct Spread (DS), Multi-Carrier (MC), and Time Division Duplex (TDD), where the DS and TDD modes are specified as in WCDMA by the 3GPP group, and the MC mode is, as in *cdma2000*, specified by 3GPP2. All air interfaces (all modes) can be used with both core networks.

A migration (or evolution) path was specified for the 2G wireless networks supporting digital communication over circuit-switched channels to the 3G networks supporting high data rates over both circuit-switched and packet-switched channels. The evolution path offers intermediate steps that are easier and cheaper to achieve, which is associated with enhanced data rates and packet data services (i.e., the addition of packet switching to 2G networks). Table 17.1 summarizes the 2G, 2.5G, and 3G standards that have been developed using the IS-41 core networks (in North America) and GSM MAP core networks (in Europe).

The bandwidth made available by 3G networks gives rise to applications not previously available to mobile phone users. Examples include online maps, online gaming, mobile TV, and instant picture/video content sharing. The multimedia nature of these 3G wireless services also calls for a rapid development of new generations of handsets, where support for high-quality video, better software and user interface, and longer battery life are key factors. These smartphones and tablets have greatly changed the way for people to interact with mobile devices and even their social behaviors.

Table 17.1 Evolution from 2G to 3G wireless networks

		Peak data rate R	Carrier spectrum W (MHz)
	IS-41 core network		
2G	cdmaOne (IS-95A)	14.4 kbps	1.25
2.5G	cdmaOne (IS-95B)	115 kbps	1.25
3G	cdma2000 1X	307 kbps	1.25
3G	cdma2000 1xEV-DO	2.4 Mbps	1.25
3G	cdma2000 1xEV-DV	4.8 Mbps	1.25
3G	cdma2000 3X	>2 Mbps	5
	GSM MAP core network		
2G	GSM (TDMA)	14.4 kbps	1.25
2.5G	GPRS (TDMA)	170 kbps	1.25
3G	EDGE (TDMA)	384 kbps	1.25
3G	WCDMA	2 Mbps	5

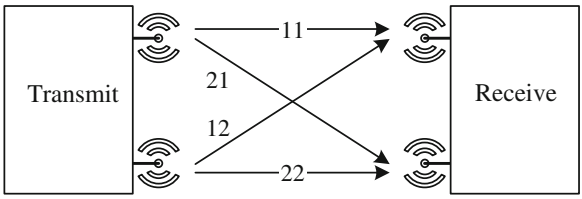


Fig. 17.8 A 2 × 2 MIMO antenna system

17.2.4 4G Cellular Networks and Beyond

Continuous improvements in semiconductor and computing technologies encourage the wireless industry and consumers to naturally anticipate 4G wireless networking [6]. Several new radio techniques are employed to achieve higher rates and lower latencies than 3G. They include *Space Division Multiplexing* via *Multiple Input/Multiple Output* (MIMO), *Space Time Coding* (STC) using higher order of modulation and encoding schemes, sophisticated beam forming and beam directionality control, and intercell interference mitigation. Of these, MIMO and beam forming are advanced antenna technologies. Using multiple sending and receiving antennas, MIMO creates multiple channels to carry user information, leading to higher capacity and less impact from interference. Figure 17.8 shows a typical 2 × 2 MIMO system. The beam-forming techniques temporarily improve gain and offer higher capacity. The properties of a beam are tuned or customized for a subscriber to achieve this capability for a limited duration. STC improves the number of bits transmitted per Hz over the available bandwidth. These techniques collectively lead to higher capacity as required by advanced networks [7].

Additionally, techniques that reduce interference are also used to further boost the capacity, most notably *Orthogonal Frequency Division Multiplexing* (OFDM).

While CDMA is well-suited for voice, OFDM can be a better transport mechanism for multimedia data. With a mix of technologies, backward compatibility is possible.

There are, however, still many debates about the definition of 4G given the breadth of technology covered under the 4G umbrella. IMT-Advanced, as defined by ITU, has been commonly viewed as the guideline for 4G standards [8]. An IMT-Advanced cellular system must fulfill the following requirements:

- Based on an all-IP packet switched network.
- Peak data rates of up to 100 Mbps for high mobility and up to 1 Gbps for nomadic/local wireless access.
- Dynamically share and use the network resources to support more simultaneous users per cell.
- Smooth handovers across heterogeneous networks.
- High quality of service for next generation multimedia support.

The pre-4G *3GPP Long-Term Evolution* (LTE) technology is often branded as 4G-LTE. The initial LTE releases had a theoretical capacity of up to 100 Mbps in the downlink and 50 Mbps in the uplink, which however still do not fully comply with the IMT-Advanced requirements.

In September 2009, a number of proposals were submitted to ITU as 4G candidates, mostly based on two technologies:

- LTE Advanced standardized by the 3GPP.
- 802.16m standardized by the IEEE (i.e., WiMAX).

These two candidate systems have been commercially deployed: the first Mobile WiMAX network in South Korea in 2006, and the first LTE network in Oslo, Norway and Stockholm, Sweden in 2009. Today LTE Advanced has largely taken the place of WiMAX, and has been considered as the standard for 4G [9].

The target of 3GPP LTE Advanced is to reach and surpass the ITU requirements through improving the existing LTE network. This upgrade path makes it more cost effective for vendors. LTE Advanced makes use of additional spectrums and multiplexing to achieve higher data speeds. It achieves peak download rates up to 299.6 Mbps and upload rates up to 75.4 Mbps depending on the user equipment category (e.g.,with 4×4 MIMO antennas using 20 MHz of spectrum). Five different terminal classes have been defined from a voice centric class up to high end terminals that support the peak data rates. It also enables lower data transfer latencies ($<$5 ms latency for small packets), and lower latencies for handover and connection setup time than with previous radio access technologies. Support for mobility is improved too. Depending on the frequency band, it allows terminals to move at speeds up to 350 km/h (220 mph) or 500 km/h (310 mph). More importantly, through macrodiversity, also known as *group cooperative relay*, high bitrates are now available in a larger portion of a cell, especially to users in an exposed position in between several BSs. All these enable high-quality multimedia services with seamless mobility, even in such extreme scenario as today's high-speed trains.

17.2.5 Wireless Local Area Networks

The increasing availability of such mobile computing devices as laptops and tablets brought about keen interest in *Wireless Local Area Networks* (WLANs), which potentially provide much higher throughput with much lower costs than the wide-area cellular wireless networks. The emergence lately of ubiquitous and pervasive computing [10] has further created a new surge of interest in WLANs and other short-range communication techniques.

Most of today's WANs are based on the 802.11 family of standards (also known as Wi-Fi), developed by the IEEE 802.11 working group. They specify Medium Access Control (MAC) and Physical (PHY) layers for wireless connectivity in a local area within a radius less than 100 m, addressing the following important issues:

- **Security**. Enhanced authentication and encryption, since the broadcast over-the-air is more susceptible to break-ins.
- **Power management**. Saves power during no transmission and handles *doze* and *awake*.
- **Roaming**. Permits acceptance of the basic message format by different AP.

The initial 802.11 standard uses the 2.4 GHz radio band, which is the globally unlicensed *Industrial, Scientific and Medical* (ISM) short-range radio frequency band. As such, it faces interferences from both of its own users and many other wireless systems, e.g., cordless phones.

Similar to Ethernet, the basic channel access method of 802.11 is *Carrier Sense Multiple Access* (CSMA). However, *Collision Detection* (CD) in Ethernet is not employed. This is because of the unique *Hidden Terminal* problem in wireless communications. As shown in Fig. 17.9, wireless terminals S1 and S3 are at a far edge of the AP (S2)'s range. Recall that, unlike that of wired signals, the strength of a wireless signal decays very quickly with distance and a receiver can hear the signal only if its strength is above a certain threshold. As such, even if both S1 and S3 are "exposed" to AP S2, i.e., can hear S2 (and vice versa), they do not necessarily hear each other given the long distance. These two terminals are therefore "hidden" to each other—if they send packets simultaneously, the two packets will collide at S2, but neither S1 nor S3 can detect the collision.

To address the hidden terminal problem, 802.11 uses *Collision Avoidance* (CA); that is, during carrier sensing, if another node's transmission is heard, the current node should wait for a period of time for transmission to finish before listening again for a free communications channel. CSMA/CA can optionally be supplemented by the exchange of a Request to Send `RTS` packet sent by the sender, and a Clear to Send `CTS` packet sent by the intended receiver. This alters all the nodes within the range of the sender, receiver, or both, to not transmit for the duration of the intended transmission. For example, before sending a message to S2, S1 can first send an `RTS` request, and S2 will then broadcast a `CTS` for S1, which will be heard by both S1 and S3. S1 can then send the message and S3 will temporarily refrain from sending, thus avoid potential collisions.

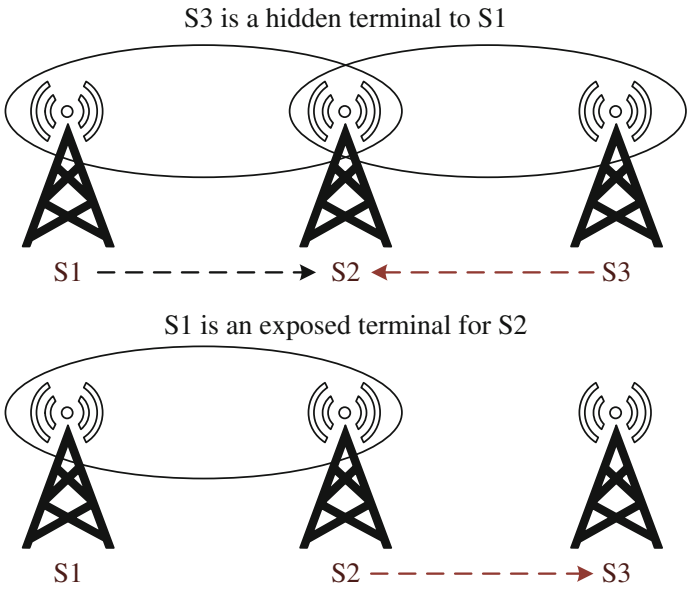


Fig. 17.9 An illustration of the hidden terminal problem. S2 is the AP; S1 and S3 are "hidden" to each other due to the long distance, but they can cause interference at S2

IEEE 802.11b/g

IEEE 802.11b is an enhancement of the basic 802.11. It uses DS spread spectrum and operates in the 2.4 GHz band. With the aid of new modulation technologies, it supports 5.5 and 11 Mbps in addition to the original 1 and 2 Mbps, and its functionality is comparable to Ethernet.

In North America, for example, the allocated spectrum for 802.11b is 2.400–2.4835 GHz. Regardless of the data rate (1, 2, 5.5, or 11 Mbps), the bandwidth of a DS spread spectrum channel is 20 MHz. Three nonoverlapped DS channels can be accommodated simultaneously, allowing a maximum of 3 APs in a local area.

IEEE 802.11g, an extension of 802.11b, is an attempt to achieve data rates up to 54 Mbps. It was designed to be downward compatible with 802.11b and hence still uses the 2.4 GHz band, but OFDM is used instead of DS spread spectrum. IEEE 802.11b/g has gained public acceptance and is appearing in WLANs everywhere, including university campuses, airports, conference centers, and so on.

IEEE 802.11a

IEEE 802.11a operates in the 5 GHz band and supports data rates in the range of 6–54 Mbps. It uses OFDM instead of DS spread spectrum, too, and allows 12 nonoverlapping channels, hence a maximum of 12 APs in a local area.

Because 802.11a operates in the higher frequency (5 GHz) band, it faces much less radio interference, such as from cordless phones, than 802.11 and 802.11b. Coupled with the higher data rate, it has great potential for supporting various multimedia applications in a LAN environment.

802.11a products started shipping late, lagging 802.11b products due to the 5 GHz components being more difficult to manufacture. It was then not widely adopted in the consumer space given that the less-expensive 802.11b was already dominating the market. With the arrival of less expensive early 802.11g products on the market, which were backwards-compatible with 802.11b, the bandwidth advantage of the 5 GHz 802.11a in the consumer market was further reduced. It, however, does penetrate into enterprise network environments which require increased capacity and reliability over 802.11b/g-only networks. Dual-band, or dual-mode APs and network interface cards that can automatically handle 802.11a and b/g, are now common in all the markets, and very close in price to b/g- only devices.

IEEE 802.11n and 802.11ac

The latest WLAN standard, 802.11n, improves network performance over all the past 802.11 standards, with a significant increase in the maximum net data rate to 600 Mbps with the use of four spatial streams at a channel width of 40 MHz [11]. It builds on previous 802.11 standards by adding multiple input, multiple output (MIMO), and frame aggregation to the MAC layer.

Channels operating with a width of 40 MHz are another feature incorporated into 802.11n; this doubles the channel width from 20 MHz in the previous 802.11 PHYs to transmit data, and provides twice the PHY data rate available over a single 20 MHz channel. It can be enabled in the 5 GHz mode, or within the 2.4 GHz mode if there is knowledge that it will not interfere with any other 802.11 or non-802.11 systems (such as cordless phones) using the same frequencies.

When 802.11g was released to share the band with existing 802.11b devices, it provided ways of ensuring coexistence between legacy and successor devices. 802.11n extends the coexistence management to protect its transmissions from legacy devices, including 802.11a/b/g, making its deployment much easier and smooth. It is quickly replacing the existing 802.11a/b/g devices in recent years, offering much better support for multimedia over wireless.

A newer WALN standard, 802.11ac, is under active development, with the final 802.11 Working Group approval and the publication scheduled for early 2014. It will offer multistation WLAN throughput of at least 1 Gbps and a single link throughput of at least 500 Mbps. This is accomplished by further enhancing the air interfaces in 802.11n: wider radio bandwidth (up to 160 MHz), more MIMO streams (up to 8), multiuser MIMO, and high-density modulation.

17.2.6 Bluetooth and Short-Range Technologies

It is known that proximity-based services have constituted a considerable portion of the mobile data traffic. Such services enable geographically close users to directly exchange data. *Bluetooth* (named after the tenth-century king of Denmark, Harold Bluetooth) is a protocol intended for such short-range (called *piconet*) wireless communications [12].

Bluetooth uses *Frequency Hopping* (FH), a spread spectrum technology for data transmission in the 2.4 GHz ISM short-range radio frequency band. Similar techniques have been used in the 802.11 WLAN. Bluetooth also employs a *master-slave* structure. One master may communicate with up to seven slaves in a piconet; all devices share the master's clock. Packet exchange is based on a basic clock, defined by the master.

Bluetooth provides a secure and low-cost way to connect and exchange information between devices such as faxes, mobile phones, laptops, printers, Global Positioning System (GPS) receivers, digital cameras, and video game consoles. It was principally designed as a low-bandwidth technology. However, it permits moving or still pictures to be sent from a digital camera or mobile phone, at a speed of over 700 kbps, within a distance of 10 m. Many other short-range wireless communication protocols have also been developed in recent years for direct data exchange between mobile devices, including Near Field Communication (NFC) and Wi-Fi Direct, and etc. Advanced technologies such as Ultra Wide Band (UWB) and cognitive radio are also under active development.

17.3 Multimedia Over Wireless Channels

We have studied the evolution of 2G networks to high-capacity 3G/4G networks as well as that of wireless local area networks. The main driving force toward the new generation of higher speed wireless networks are from multimedia communications over wireless. Suggested multimedia applications range from streaming video, video-conferencing, online gaming, collaborative work, and slide show presentations, to enhanced roadside assistance and online map guidance for drivers, to name but a few.

The characteristics of wireless handheld devices are worth keeping in mind when designing multimedia transmission over wireless, in particular video transmission. First, both the handheld size and battery life limit the processing power and memory of the device. Thus, encoding and decoding must have relatively low complexity. On the other hand, the smaller screen sizes well accept relatively lower resolution videos, which helps reduce the processing time. This, however, is changing given the rapid adoption of high-resolution screens in mobile devices.

Second, due to memory constraints and reasons for the use of wireless devices, as well as billing procedures, real-time communication is likely to be required. Long delays before starting to see a video are either not possible or not acceptable.

Finally, wireless channels have much more interference than wired channels, with specific loss patterns depending on the environment conditions. The bitrate for wireless channels is also much more limited, even with the 3G/4G. This implies that even though a lot of bit protection must be applied, coding efficiency has to be maintained as well. And error-resilient coding is important.

The 3G standards specify that video shall be standard compliant. Moreover, most companies will concentrate on developing products using standards, in the interest of interoperability of mobiles and networks. The video standards reasonable for use over wireless channels are MPEG-4 and H.263/264/265 and their variants, given their high efficiency in low bitrate. The 3GPP/3GPP2 group has defined the following QoS parameters for wireless videoconferencing services [13,14].

- **Synchronization**. Video and audio should be synchronized to within 20 ms.
- **Throughput**. The minimum video bitrate to be supported is 32 kbps. Video rates of 128 kbps, 384 kbps, and above should be supported as well.
- **Delay**. The maximum end-to-end transmission delay is defined to be 400 ms.
- **Jitter**. The maximum delay jitter (maximum difference between the average delay and the 95th percentile of the delay distribution) is 200 ms.
- **Error rate**. A frame error rate of 10^{-2} or a bit error rate of 10^{-3} should be tolerated.

In this section, we are concerned mainly with sending multimedia data robustly over wireless channels, particularly for video communication, the natural extension to voice communication. We will introduce solutions for error detection, error correction, error-resilient entropy coding, and error concealment in the wireless network context, although most of these techniques are also applicable to other networks.

17.3.1 Error Detection

Error detection is to identify errors caused by noise or other impairments during transmission from the sender to the receiver. Commonly used error detection tools include parity checking, checksum, and Cyclic Redundancy Check (CRC) [15,16].

Parity Checking

With binary data, errors appear as bit flips. Parity checking adds a *parity bit* to a source bitstring to ensure that the number of *set bits* (i.e., bits with value 1) in the outcome is even (called *even parity*) or odd (called *odd parity*). For example, with even parity checking, a bit 1 should be appended to bitstring 10101000, and a bit 0 should be appended to 10101100.

This is a very simple scheme that can be used to detect any single or odd number of errors on the receiver's side. An even number of flipped bits, however, will make the parity bit appear correct even though the data is erroneous.

Checksum

A checksum of an input message is a modular arithmetic sum of all the codewords in the message. The sender can append the checksum to the message, and the receiver can perform the same sum operation to check whether there is any error. It has been implemented in many network protocols, from data link and network layers, to transport and application layers. The *Internet checksum* algorithm in these protocols works as follows (see more details in RFC 1071):

1. First pair the bytes of the input data to form 16-bit integers. If there is an odd number of bytes, then append a byte of zero in the end.
2. Calculate the 1's complement sum of these 16-bit integers. Any overflow encountered during the sum will be wrapped around to the lowest bit.
3. The result serves as the checksum field, which is then appended to the 16-bit integers.
4. On the receiver's end, the 1's complement sum is computed over the received 16-bit integers, including the checksum field. Only if all the bits are 1 will the received data be correct.

To illustrate this, let the input data be a byte sequence of $D_1, D_2, D_3, D_4, \ldots, D_N$. Using the notation $[a, b]$ for the 16-bit integer $a \cdot 256 + b$, where a and b are bytes, then the 16-bit 1's complement sum of these bytes is given by one of the following (here $+'$ means 1's complement sum):

$$[D_1, D_2] +' [D_3, D_4] +' \cdots +' [D_{N-1}, D_N] \quad (N \text{ is even; no padding})$$

$$[D_1, D_2] +' [D_3, D_4] +' \cdots +' [D_N, 0] \quad (N \text{ is odd; append a zero}).$$

As an example, suppose we have the following input data of 4 bytes: 10111011, 10110101, 10001111, and 00001100. They will be grouped as 1011101110110101 and 1000111100001100.

The sum of these two 16 bit integers is

$$\begin{array}{r} 1011101110110101 \\ +1000111100001100 \\ \hline 0100101011000010 \end{array}$$

This addition has an overflow, which has been wrapped around to the lowest bit. The 1's complement is then obtained by converting all the 0s to 1s and all the 1s to 0s. Thus the 1s complement of the above sum becomes 1011010100111101, which becomes the checksum.

The receiver will perform the same grouping and summation for the received bytes, and then add the received checksum too. It is easy to see that if there is no error, then the outcome should be 1111111111111111. Otherwise, if any bit becomes 0, then errors happen during transmission.

Cyclic Redundancy Check

The basic idea behind *Cyclic Redundancy Check* (CRC) is to divide a binary input by a keyword K that is known to both the sender and the receiver. The remainder R after the division constitutes the *check word* for the input. The sender sends both the input data and the check word, and the receiver can then check the data by repeating the calculation and verifying whether the remainder is still R. Obviously, to ensure that the check word R is fixed to r bits (zeros can be padded at the highest bits if needed), the keyword K should be of $r + 1$ bits.

CRC implementation uses a simplified form of arithmetic for the division, namely, computing the remainder of dividing with modulo-2 in GF(2) (Galois field with two elements), in which we have

$$\begin{aligned} 0 - 0 &= 0 + 0 = 0 \\ 1 - 0 &= 1 + 0 = 1 \\ 0 - 1 &= 0 + 1 = 1 \\ 1 - 1 &= 1 + 1 = 0 \end{aligned}$$

In other words, addition and subtraction are identical and both are equivalent to *exclusive OR* (XOR, $\oplus$). Multiplication and division are the same as in conventional base-2 arithmetic, too, except that, with the XOR operation, any required addition or substraction is now without carries or borrows. All these make the hardware implementation much simpler and faster.

Given the message word M, and the keyword K, we can manually calculate the remainder R using conventional long division, just with modulo-2 arithmetic. We also append r zeros to M before division, which makes the later verification easier, as we will see soon. For example, for $M = 10111$ and $K = 101$, we have

$$\begin{array}{r} 10011 \\ 101\overline{)1011100} \\ 101 \\ \hline 110 \\ 101 \\ \hline 110 \\ 101 \\ \hline 11 \end{array}$$

Hence, $R = 11$, which is to be appended as the check word to the message.

It is not difficult to show that, in this case, $M \cdot 2^r \oplus R$ is perfectly divisible by K (which we leave as an exercise). Hence, instead of calculating the remainder on the receiver's side and comparing with the R from the sender, the receiver can simply divide $M \cdot 2^r \oplus R$ by K and check whether the remainder is zero or not. If it is zero, then there is no error; otherwise, the error is detected.

The keyword K indeed comes from a *generator polynomial* whose coefficients are the binary bits of the keyword K. For example, for $K = 100101$ in the binary

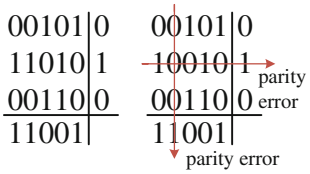


Fig. 17.10 An example of a two-dimensional even parity checking. *Left* No error; *Right* A single-bit error detected and corrected

format, its polynomial expression is $x^5 + x^2 + 1$. The keyword is, therefore, also called a *generator*. Choosing a good generator is a nontrivial job and there have been extensive studies [16]. For a well-chosen generator, two simple facts are known (more can be found in [16]):

1. If the generator polynomial contains two or more terms, all single-bit errors can be detected.
2. An r-bit CRC can detect all burst errors of length no more than r. Here the burst means that the first and last bits are in error, and the bits in between may or may not be in error.

Standard generators of 8-, 12-, 16-, and 32-bits have been defined in international standards. For example, the following 32-bit generator has been used in a number of link-layer IEEE protocols, in particular, the Internet since 1975:

$$G_{CRC-32} = 100000100110000010001110110110111$$

Note that it is of 33 bits (so that the remainder is of 32 bits). It is also the CRC generator that is used in MPEG-2, and the error message "CRC failed!" often appears for a scratched DVD that is hard to read by the disk player.

17.3.2 Error Correction

Once an error is detected, a retransmission could be used to recover the error, as such reliable transport protocols as TCP does. The back channel, however, is not always available, e.g., for satellite transmission, or can be quite expensive to create, e.g., in the broadcast or multicast scenarios. For real-time multimedia streaming, the delay for retransmission can be too long, making the retransmitted packet useless.

Instead, for real-time multimedia, *Forward Error Correction* (FEC) is often used, which adds redundant data to a bitstream to recover some random bit errors in it [16]. Consider a simple extension to the parity checking, from one dimension to two dimensions [17]. We not only calculate the parity bit for each bitstring of M bits, but also group every M bitstrings to form a matrix and calculate the parity bit of each column of the matrix.

With this *two-dimensional parity checking*, we can both detect and correct errors! This is because a bit error will cause a failure of a row parity checking and a failure

of a column parity checking, which cross at a unique location—the flipped bit in the erroneous bitstring, as illustrated in the example in Fig. 17.10.

This is a very simple FEC scheme. It doubles the amount of parity bits, but the error correction capability is very limited; e.g., if two errors occur in a row, we will not be able to detect them, not to mention correcting them.

There are two categories of practical error correction codes: *block codes* and *convolutional codes* [15,16]. The block codes apply to a group of bits, i.e., a block, at once to generate redundancy. The convolutional codes apply to a string of bits one at a time and have memory that can store previous bits as well.

Block Codes

The block codes take an input of k bits and append $r = n - k$ bits of FEC data, resulting in an n-bit-long string [18]. These codes are referred to as (n, k) codes. For instance, the basic ASCII words are of 7 bits; parity checking adds a single bit ($r = 1$) for any ASCII word, and so it is an (8,7) code, with 8 bits in total ($n = 8$), of which seven are data ($k = 7$). The *code rate* is k/n, or 7/8 in the parity checking case.

Hamming Codes

Richard Hamming observed that error correction codes operate by adding space between valid source strings. The space can be measured using a *Hamming distance*, defined as the minimum number of bits between *any* coded strings that need to be changed so as to be identical to another valid string.

To detect r errors, the Hamming distance has to be at least equal $r + 1$; otherwise, the corrupted string might seem valid again. This is not sufficient for correcting r errors, however, since there is not enough distance among valid codes to choose a preferable correction. To correct r errors, the Hamming distance must be at least $2r + 1$.

This leads to the invention of first block code, the *Hamming(7,4)-code*, in 1950. It encodes 4 data bits into 7 bits by adding 3 parity bits based on a generator matrix G, say

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Given an input data p (4 bits as a vector), the output code x is obtained by taking the product $G \cdot p$ and then performing modulo 2. As an example, for bits 1001, the input vector p is $(1, 0, 0, 1)^T$; the product will be vector $(2, 2, 1, 1, 0, 0, 1)^T$, and the encoded output x will be $(0, 0, 1, 1, 0, 0, 1)^T$ after modulo 2, or a 7-bit data block of 0011001.

The Hamming(7,4)-code can detect and correct any single-bit error. To do this, a parity-check matrix H is used

$$H = \begin{pmatrix} 1\,0\,1\,0\,1\,0\,1 \\ 0\,1\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1\,1\,1 \end{pmatrix}$$

Similar to encoding, we take the product $H \cdot x$ with modulo 2, which yields a vector z of length 3. We can treat z as a 3-bit binary number. If it is zero, then there is no error; otherwise, it indicates the location of the error (by checking the corresponding column in H). For example, for $x = 0011001$, we have $z = 0$, i.e., no error; on the other hand, for $x' = 0111001$, we have $z = 010$, which corresponds to the second column of H, indicating bit 2 is erroneous.

Extended Hamming codes can detect up to 2-bit errors or correct 1-bit errors without detection of uncorrected errors. By contrast, the simple 1D parity code can detect only an odd number of bits in error and cannot correct errors.

BCH and RS Codes

More powerful *cyclic codes* are stated in terms of generator polynomials of maximum degree equal to the number of source bits. The source bits are the coefficients of the polynomial, and redundancy is generated by multiplying with another polynomial. The code is cyclic, since the modulo operation in effect shifts the polynomial coefficients. The Cyclic Redundancy Check (CRC) we have seen before belongs to this category, though it is mainly used for error detection. A widely used class of cyclic error correction codes is the *Bose-Chaudhuri-Hocquenghem* (BCH) codes. The generator polynomial for BCH is also given over a Galois Field (GF) and is the lowest degree polynomial with roots of α^i, where α is a primitive element of the field and i goes over the range of 1 to twice the number of bits we wish to correct.

BCH codes can be encoded and decoded quickly using integer arithmetic. H.261 and H.263 use BCH to allow for 18 redundant bits every 493 source bits. Unfortunately, the 18 redundant bits will correct at most two errors in the source. Thus, the packets are still vulnerable to burst bit errors or single-packet errors.

An important subclass of BCH codes that applies to multiple packets is the *Reed–Solomon* (RS) codes. The RS codes have a generator polynomial over GF(2^m), with m being the packet size in bits. RS codes take a group of k source packets and output n packets with $r = n - k$ redundancy packets. Up to r lost packets can be recovered from n coded packets if we know the erasure points.[1] Otherwise, as with all FEC codes, recovery can be applied only to half the number of packets, since error-point detection is now necessary as well.

In the RS codes, only $\lceil \frac{r}{2} \rceil$ packets can be recovered. Fortunately, in the packet FEC scenario, the packet itself often contains a sequence number and checksum or CRC in its header. In most cases, a packet with an error is dropped, and we can tell the location of the missing packet from the missing sequence number.

[1] Errors are also called *erasures*, since an erroneous packet can be useless, and has to be "erased".

The RS codes are useful for both storage and transmission over networks. When there are burst packet losses, it is possible to detect which packets were received incorrectly and recover them using the available redundancy. If the video has scalability, a better use of allocated bandwidth is to apply adequate FEC protection on the base layer, containing motion vectors and all header information required to decode video to the minimum QoS. The enhancement layers can receive either less protection or none at all, relying just on resilient coding and error concealment. Either way, the minimum QoS is already achieved.

A disadvantage of the block codes is that they cannot be selectively applied to certain bits. It is difficult to protect higher protocol layer headers with more redundancy bits than for, say, DCT coefficients, unless they are sent explicitly through different packets. On the other hand, convolutional codes can do this, which make them more efficient for data in which unequal protection is advantageous, such as videos.

Convolutional Codes

The convolutional FEC codes are defined over generator polynomials as well [15]. They are computed by shifting k message bits into a coder that convolves them with the generator polynomial to generate n bits. The rate of such code is defined to be $\frac{k}{n}$. The shifting is necessary, since coding is achieved using memory (shift) registers. There can be more than k registers, in which case past bits also affect the redundancy code generated.

After producing the n bits, some redundancy bits can be deleted (or "punctured") to decrease the size of n, and increase the rate of the code. Such FEC schemes are known as *rate compatible punctured convolutional* (RCPC) codes. The higher the rate, the lower the bit protection will be, but also the less overhead on the bitrate. A *Viterbi algorithm* with soft decisions decodes the encoded bit stream, although *turbo codes* are gaining popularity.

Given the limited network bandwidth, it is important to minimize redundancy, because it comes at the expense of bitrates available for source coding. At the same time, enough redundancy is needed so that the video can maintain required QoS under the current channel error conditions. Moreover, the data in a compressed media stream are of different importance. Some data are vitally important for correct decoding. For example, some lost and improperly estimated data, such as picture coding mode, quantization level, or most data in higher layers of a video standard protocol stack, will cause catastrophic video decoding failure. Others, such as missing DCT coefficients may be estimated or their effect visually concealed to some degree. As such, given certain channel conditions, different amount of FEC can be applied to these data to provide different level of protection, which is known as *Unequal Error Protection (UEP)*.

RCPC puncturing is done after generation of parity information. Knowing the significance of the source bits for video quality, we can apply a different amount of puncturing and hence achieve UEP. Studies and simulations of wireless radio models have shown that applying unequal protection using RCPC according to bit

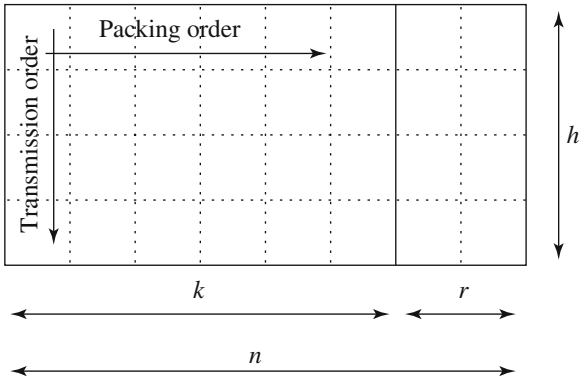


Fig. 17.11 Interleaving scheme for redundancy codes. Packets or bits are stored in rows, and redundancy is generated in the last r columns. The sending order is by columns, *top* to *bottom*, then *left* to *right*

significance information results in better video quality (up to 2 dB better) for the same allocated bitrate than videos protected using the RS codes.

Simplistically, the Picture layer in a video protocol should get the highest protection, the macroblock layer that is more localized will get lower protection, and the DCT coefficients in the block layer can get little protection, or none at all. This could be extended further to scalable videos in similar ways.

The 3G networks have also incorporated data-type-specific provisions and recognize the video standard chosen for transmission; they can adaptively apply transport coding of the video stream with enough unequal redundancy suitable to the channel conditions at the time and QoS requested.

Packet Interleaving

It is also possible to use *packet interleaving* to increase resilience to burst packet loss. As Fig. 17.11 shows, the RS codes are generated for each of the h rows of k source video packets. Instead of transmitting with the original order, we can use the column-major order, so that the first packet of each of the h rows is transmitted first, then the second, and so on. Such an interleaving can effectively convert a burst loss to a series of smaller uniform losses across the original rows, which are much easier to handle given the enough redundancy in each row. In other words, we could tolerate more than r erasures with error correction and concealment.

It is worth noting that the interleaving does not increase bandwidth overhead but introduces additional delay.

17.3.3 Error-Resilient Coding

A video stream is either packetized and transmitted over a packet-switched channel or transmitted as a continuous bitstream over a circuit-switched channel, with the former being more popular nowadays. In either case, it is obvious that packet loss or bit error will reduce video quality. If a bit loss or packet loss is localized in the video

in both space and time, the loss can still be acceptable, since a frame is displayed for a very short period, and a small error might go unnoticed.

However, digital video coding techniques involve variable-length codes, and frames are coded with different prediction and quantization levels. Unfortunately, when a packet containing variable bit length data (such as DCT coefficients) is damaged, that error, if unconstrained, will propagate all the way throughout the stream. This is called *loss of decoder synchronization*. Even if the decoder can detect the error due to an invalid coded symbol or coefficients out of range, it still cannot establish the next point from which to start decoding [19].

As we have learned in Chap. 10, this complete bitstream loss does not happen for videos coded with standardized protocol layers. The Picture layer and the Group Of Blocks (GOB) layer or Slice headers have *synchronization markers* that enable decoder resynchronization. For example, the H.263 bitstream has four layers—the Picture layer, GOB layer, Macroblock layer, and Block layer. The Picture Layer starts with a unique 22-bit picture start code (PSC). The longest entropy-coded symbol possible is 13 bits, so the PSC serves as a synchronization marker as well. The GOB layer is provided for synchronization after a few blocks rather than the entire frame. The group of blocks start code (GBSC) is 17 bits long and also serves as a synchronization marker.[2] The macroblock and the Block layers do not contain unique start codes, as these are deemed high overhead.

Slice Mode

ITU standards after H.261 (i.e., H.263 to 265) support slice-structured mode instead of GOBs (see for example H.263 Annex K), where slices group block together according to the block's coded bit length rather than the number of blocks. The objective is to space slice headers within a known distance of each other. That way, when a bitstream error looks like a synchronization marker and if the marker is not where the slice headers should be, it is discarded and no false resynchronization occurs.

Since slices need to group an integral number of macroblocks together, and macroblocks are coded using VLCs, it is not possible to have all slices the same size. However, there is a minimum distance after which the next scanned macroblock will be added to a new slice. We know that DC coefficients in macroblocks and motion vectors of macroblocks are differentially coded. Therefore, if a macroblock is damaged and the decoder locates the next synchronization marker, it might still not be able to decode the stream.

To alleviate the problem, slices also reset spatial prediction parameters; differential coding across slice boundaries is not permitted. The ISO MPEG standards (and H.264 as well) specify slices that are not required to be of similar bit length and so do not protect against false markers well.

[2] Synchronization markers are always larger than the minimum required, in case bit errors change bits to look like synchronization markers.

Other than synchronization loss, we should note that errors in prediction reference frames cause much more damage to signal quality than errors in frames not used for prediction. That is, a frame error for an I-frame will deteriorate the quality of a video stream more than a frame error for a P- or B-frame. Similarly, if the video is scalable, an error at the base layer will deteriorate the quality of a video stream more than in enhancement layers.

Reversible Variable-Length Code

Another useful tool to address the loss of decoder synchronization is *Reversible Variable-Length Code* (RVLC) [20,21]. An RVLC makes instantaneous decoding possible both in the forward and backward directions. With the conventional VLC, a single-bit error can cause continuous errors in reconstructing the data even if no further bit error happens. In other words, the information carried by the remaining correct bits become useless. If we can decode from the reverse direction, then such information could be recovered. Another potential use of RVLC is in the random access of a coded stream. The ability to decode and search in two directions should halve the amount of indexing overhead with the same average search time as compared to the standard one-directional VLC.

An RVLC, however, must satisfy the prefix condition for instantaneous forward decoding (as we have seen in Chap. 7) and also a suffix condition for instantaneous backward decoding. That is, each code word must not coincide with any suffix of a longer code word. A conventional VLC, say, Huffman coding, satisfies only the prefix condition and can only be decoded from left to right.

As an example, consider the symbol distribution in Table 17.2. For input ACDBC, the Huffman coded bit stream (C_1 in Table 17.2) is 10010011101, which cannot be decoded instantaneously in the backward direction (right to left) because the last two bits 10 might be either symbol "C" or the suffix of "D".

To ensure both the prefix and the suffix conditions, we can use a VLC composed entirely of symmetrical code words, e.g., the second column (C_2) in Table 17.2. Each symmetric code is clearly reversible, and a bit stream formed by them is reversible too. For example, ACDBC will be coded as 0010101011101, which is uniquely decodable from both directions. Compared to Huffman coding (average code length of 2.21), this symmetric RVLC has a slightly longer average code length (2.44). More efficient asymmetric RVLC can also be systematically constructed (C_3 in the table), which has an average code length of 2.37. Though it is still higher than that of Huffman coding, the overhead is acceptable given the potential benefit of bidirectional decoding.

RVLC has been used in MPEG-4 Part 3. To further help with synchronization, a data partitioning scheme in MPEG-4, groups and separates header information, motion vectors, and DCT coefficients into different packets and puts synchronization markers between them. Such a scheme is also beneficial to unequal protection.

Additionally, an adaptive intraframe refresh mode is allowed, where each macroblock can be coded independently of the frame as an inter- or intrablock according

Table 17.2 Huffman code (C_1), Symmetric RVLC (C_2), and Asymmetric RVLC (C_3)

Symbol	Probability	C_1	C_2	C_3
A	0.32	10	00	11
B	0.32	11	11	10
C	0.15	01	101	01
D	0.13	001	010	000
E	0.08	000	0110	00100

to its motion, to assist with error concealment. A faster moving block will require more frequent refreshing—that is, be coded in intramode more often. Synchronization markers are easy to recognize and are particularly well suited to devices with limited processing power, such as cell phones and mobile devices.

For interactive applications, if a back channel is available to the encoder, a few additional error control techniques are available with the feedback information. For example, according to the bandwidth available at any moment, the receiver can ask the sender to lower or increase the video bitrate (transmission rate control), which combats packet loss due to congestion. If the stream is scalable, it can ask for enhancement layers as well. Annex N of H.263+ also specifies that the receiver can notice damage in a reference frame and request that the encoder uses a different reference frame for prediction—a reference frame the decoder has reconstructed correctly. Unfortunately, for many real-time streaming applications with tight delay constraints or multicast/broadcast scenarios, such a backchannel for each receiver may not be available.

Error-Resilient Entropy Coding

The main purpose of GOBs, slices, and synchronization markers are to re-establish synchronization in the decoder as soon as possible after an error. In Annex K of H.263+, the use of slices achieves better resilience, since they impose further constraints on where the stream can be synchronized. *Error-Resilient Entropy Coding* (EREC), further achieves synchronization after every *single* macroblock, without any of the overheads of the slice headers or GOB headers. It takes entropy-coded, variable-length macroblocks and rearranges them in an error-resilient fashion. In addition, it can provide graceful degradation.

EREC takes a coded bitstream of a few blocks and rearranges them so that the beginning of all the blocks is a fixed distance apart. Although the blocks can be of any size and any media we wish to synchronize, the following description will refer to macroblocks in videos. The algorithm proceeds as in Fig. 17.12.

Initially, EREC slots (rows) of fixed bit length are allocated with a total bit length equal to (or exceeding) the total bit length of all the macroblocks. The number of slots is equal to the number of macroblocks, except that the macroblocks have varying bit length and the slots have a fixed bit length (approximately equal to the average bit length of all the macroblocks). As shown, the last EREC slot (row) is shorter when the total number of bits does not divide evenly by the number of slots.

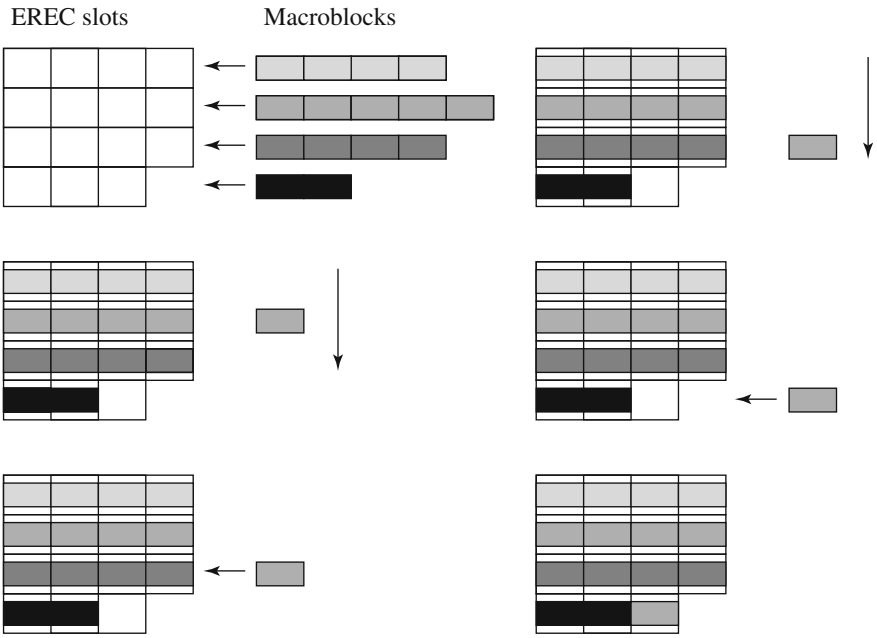


Fig. 17.12 Example of macroblock encoding using EREC

Let k be the number of macroblocks, which is equal to the number of slots, l be the total bit length of all the macroblocks, $mbs[\ ]$ be the macroblocks, and $slots[\ ]$ be the EREC slots; the procedure for encoding the macroblocks is shown below.

Procedure 17.1 Macroblock encoding using EREC

BEGIN
$j = 0$;
Repeat until $l = 0$
{
for $i = 0$ to $k - 1$
{
$m = (i + j) \bmod k$;
// m is the macroblock number corresponding to slot i;
Shift as many bits as possible (without overflow) from $mbs[i]$ into $slots[m]$;
$sb =$ number of bits successfully shifted into $slots[m]$ (without overflow);
$l = l - sb$;
}
$j = j + 1$; // shift the macroblocks downwards
}
END

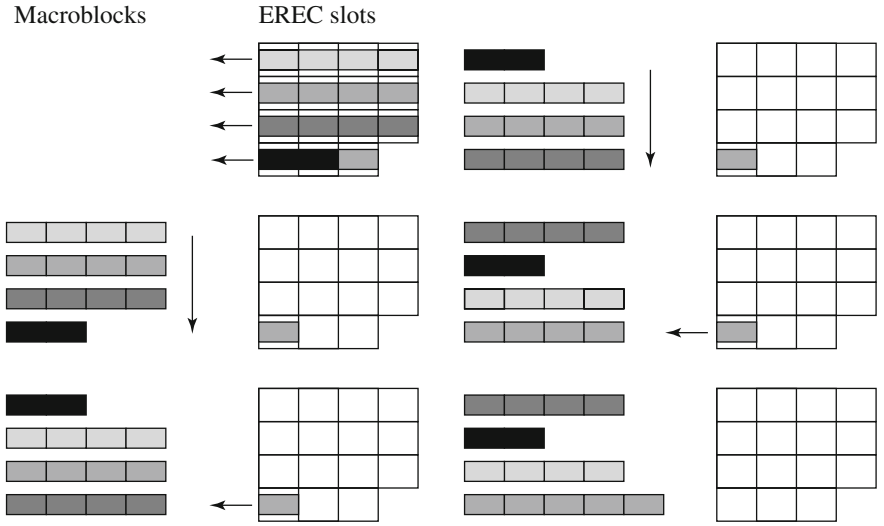


Fig. 17.13 Example of macroblock decoding using EREC

The macroblocks are shifted into the corresponding slots until all the bits of the macroblock have been assigned or remaining bits of the macroblock don't fit into the slot. Then the macroblocks are shifted down, and this procedure repeats.

The decoder side works in reverse, with the additional requirement that it has to detect when a macroblock has been read in full. It accomplishes this by detecting the end of macroblock when all DCT coefficients have been decoded (or a block end code). Figure 17.13 shows an example of the decoding process for the macroblocks coded using EREC in Fig. 17.12.

The transmission order of the data in the slots is row-major—that is, at first the data in slot 0 is sent, then slot 1, and so on, left to right. It is easy to see how this technique is resilient to errors. No matter where the damage is, even at the beginning of a macroblock, we still know where the next macroblock starts—it is a fixed distance from the previous one. In this case, no synchronization markers are used, so the GOB layer or slices are not necessary either (although we still might want to restrict spatial propagation of error).

When the macroblocks are coded using a data partitioning technique (such as the one for MPEG-4 described in the previous section) and also bitplane partitioning, an error in the bitstream will destroy less-significant data while receiving the significant data. It is obvious that the chance for error propagation is greater for bits at the end of the slot than at the beginning. On average, this will also reduce visual deterioration over a nonpartitioned encoding. This achieves graceful degradation under worsening error conditions.

17.3.4 Error Concealment

Despite all the efforts to minimize occurrences of errors and their significance, errors can still happen unless with persistent retransmission, which however is not practical for continuous media with delay constraints. The residual error will be acoustically or visually annoying. *Error concealment* techniques are then introduced to approximate the lost data on the decoder side, so as to mitigate their negative audio or visual impact.

Error concealment techniques apply in the spatial, temporal, or frequency domains, or a combination of them. For the case of video, these techniques use neighboring frames temporally or neighboring macroblocks spatially. The transport stream coder interleaves the video packets, so that in case of a burst packet loss, not all the errors will be at one place, and the missing data can be estimated from the neighborhood.

Error concealment is necessary for wireless audio/video communication, since the error rates are higher than for wired channels and might even be higher than can be transmitted with appropriate bit protection. Moreover, the error rate fluctuates more often, depending on various mobility or weather conditions. Decoding errors due to missing or wrong data received are also more noticeable on devices with limited resolution and small screen sizes. This is especially true if the macroblock size remains large, to achieve encoding efficiency for lower wireless bitrates. Here we summarize the common techniques for error concealment, particularly for video [22].

Dealing with Lost Macroblocks

A simple and popular technique for concealment can be used when DCT blocks are damaged but the motion vectors are received correctly. The missing block coefficients are estimated from the reference frame, assuming no prediction errors. Since the goal of the motion-compensated video is to minimize prediction errors, this is an appropriate assumption. The missing block is hence temporally masked using the block in the reference frame.

We can achieve even better results if the video is scalable. In that case, we assume that the base layer is received correctly and that it contains the motion vectors and base layer coefficients that are most important. Then, for a lost macroblock at the enhancement layer, we use the motion vectors from the base layer, replace the DCT coefficients at the enhancement layer, and decode as usual from there. Since coefficients of less importance are estimated (such as higher frequency coefficients), even if the estimation is not too accurate due to prediction errors, the concealment is more effective than in a nonscalable case.

If the motion vector information is damaged as well, this technique can be used only if the motion vectors are estimated using another concealment technique (to be discussed next). The estimation of the motion vector has to be good, or the visual quality of the video could be inauspicious. To apply this technique for intraframes, some standards, such as MPEG-2, also allow the acquisition of motion vectors for intracoded frames (i.e., treating them as intra- as well as interframes). These motion vectors are discarded if the block has no error.

Combining Temporal, Spatial, and Frequency Coherences

Instead of just relying on the temporal coherence of motion vectors, we can combine it with spatial and frequency coherences. By having rules for estimating missing block coefficients using the received coefficients and neighboring blocks in the same frame, we can conceal errors for intraframes and for frames with damaged motion vector information. Additionally, combining with prediction using motion vectors will give us a better approximation of the prediction error block.

Missing block coefficients can be estimated spatially by minimizing the error of a smoothness function defined over the block and neighboring blocks. For simplicity, the smoothness function can be chosen as the sum of squared differences of pairwise neighboring pixels in the block. The function unknowns are the missing coefficients. In the case where the motion information is available, prediction smoothness is added to the objective function for minimization, weighted as desired.

The simple smoothness measure defined above has the problem that it smoothes edges as well. We can attempt to do better by increasing the order of the smoothing criterion from linear to quadratic or cubic. This will increase the chances of having both edge reconstruction and smoothing along the edge direction. At a larger computational cost, we can use an edge-adaptive smoothing method, whereby the edge directions inside the block are first determined, and the smoothing is not permitted across edges.

Smoothing High-Frequency Coefficients

Although the human visual system is more sensitive to low frequencies, it would be disturbing to see a checkerboard pattern where it does not belong. This will happen when a high-frequency coefficient is erroneously assigned a high value. The simplest remedy is to set high-frequency coefficients to 0 if they are damaged.

If the frequencies of neighboring blocks are correlated, it is possible to estimate lost coefficients in the frequency domain directly. For each missing frequency coefficient in a block, we estimate its value using an interpolation of the same frequency coefficient values from the four neighboring blocks. This is applicable at higher frequencies only if the image has regular patterns. Unfortunately that is not usually the case for natural images, so most of the time the high coefficients are again set to 0. Temporal prediction error blocks are even less correlated at all frequencies, so this method applies only for intraframes.

Estimating Lost Motion Vectors

The loss of motion vectors prevents decoding of an entire predicted block, so it is important to estimate motion vectors well. The easiest way to estimate lost motion vectors is to set them to 0. This works well only in the presence of very little motion. A better estimation is obtained by examining the motion vectors of reference

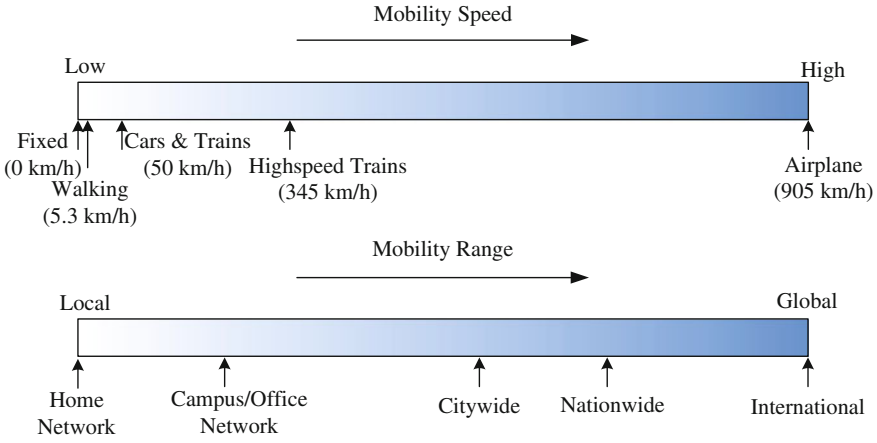


Fig. 17.14 Real-world mobility range and mobility speed

macroblocks and of neighboring macroblocks. Assuming the motion is also coherent, it is reasonable to take the motion vectors of the corresponding macroblock in the reference frame as the motion vectors for the damaged target block.

Similarly, assuming objects with consistent motion fields occupy more than one macroblock, the motion vector for the damaged block can be approximated as an interpolation of the motion vectors of the surrounding blocks that were received correctly. Typical simple interpolation schemes are weighted-average and median. Also, the spatial estimation of the motion vector can be combined with the estimation from the reference frame using weighted sums.

17.4 Mobility Management

Mobility is another distinct feature of wireless portable devices. The traditional TCP/UDP/IP networks were originally designed for communications between fixed ends. There are many issues that need to be resolved to support mobility, which has long been a research topic in the Internet community, particularly in recent years when the number of mobile terminals dramatically increases [23,24]. There is a broad spectrum of device and user mobility, in terms of both range and speed, as illustrated in Fig. 17.14.

The deep penetration of modern wireless accesses has made network connectivity anywhere and anytime a reality, which urges network operators/adminstrators to deploy mobility management protocols for ubiquitous accesses. From a network operator/administrator's view, a network usually covers a large geographical area (or administrative domain) consisting of several subnetworks. Mobility of a user in a network can be broadly classified into three categories:

- *Micromobility* (intra-subnet mobility), where movement is within a subnet.
- *Macromobility* (intradomain mobility), where movement is across different subnets within a single domain.
- *Global mobility* (interdomain mobility), where movement is across different domains in various geographical regions.

Global mobility involves longer timescales, where the goal is to ensure that mobile devices can re-establish communication after a move rather than provide continuous connectivity. Early studies on Mobile IP have addressed the simple scenario of global mobility that a computer is unplugged from a network, transported to another network, and then replugged. With the support of modern wireless mobile networks, such as 3G/4G and WLAN, the mobility can be much more frequent with complex patterns. It is, therefore, important to ensure continuous and seamless connectivity during micro- and macromobility, together with secure authentication, authorization, and accounting. The short timescales here call for joint effort across multiple layers. This is further complicated with streaming media applications that expect uninterrupted data transfer during the movement.

To avoid interruption during communication, *handoff* (also known as *handover*) management is required, by which a mobile terminal keeps its connection active when it moves from one network access point to another. Another important function needed to support mobility is *location management*, which tracks the locations of mobile terminals, and provides such popular location-based services as searching nearby users or media content related to the locations of interest.

17.4.1 Network Layer Mobile IP

We start our discussion on mobility management from the network layer support for global mobility. The most widely used protocol for this purpose is Mobile IP, whose initial version was developed by IETF in 1996. IETF released Mobile IPv4 (RFC3220) and Mobile IPv6 (RFC3775) standards in 2002 and 2004, respectively. There are certain differences in their details, but the overall architectures and the high-level designs are similar for both versions.

The key support offered by Mobile IP is to assign a mobile host two IP addresses: a *home address* (HoA) that represents the fixed address of the *mobile node* (MN) and a *care-of-address* (CoA) that changes with the IP subnet to which the MN is currently attached. Each mobile node has a *home agent* (HA) in its home network, from which it acquires its HoA. In Mobile IPv4, the foreign network where the MN currently is attached should have a *foreign agent* (FA), which is replaced by an *access router* (AR) in Mobile IPv6. The mobile node obtains its CoA from its current FA or AR.

When the mobile node MN is in its home network, it acts like any other fixed node of that network with no special mobile IP features. When it moves out of its home network to a foreign network, the following steps are to be followed (see Fig. 17.15):

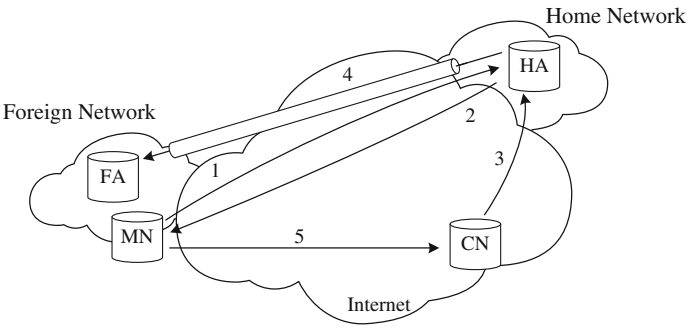
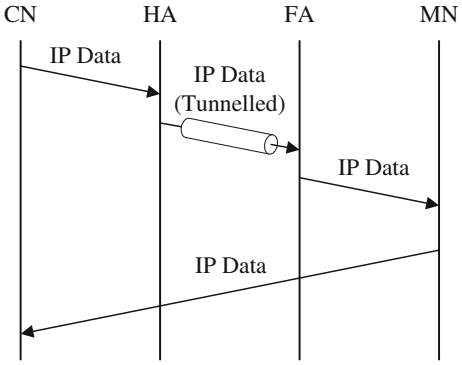


Fig. 17.15 The operations in mobile IP

Fig. 17.16 The data path in mobile IP



1. The MN obtains the CoA and informs its HA of the new address by sending a `Registration Request` message to the HA.
2. The HA, upon receiving the message, shall reply to the MN with a `Registration Reply` message. The HA keeps the binding record of the MN, which is transparent to a *correspondent node* (CN) that intends to communicate with the MN.
3. Once a packet from the CN to the MN arrives at the home network, the HA will intercept the packet.
4. The HA then forward it to the FA by a tunnel that encapsulates the original packets (with the HoA in the headers) into packets with the CoA in the headers. Once the FA receives the tunneled packets, it removes the extra header and delivers it to the MN.
5. When the MN wishes to send data back to the CN, the packets are sent directly from the MN to the destination since the CN's IP address is known by the MN.

This data path for mobile IP is further illustrated in Fig. 17.16.

Such a simple implementation that involves MN, HA, and CN can cause *triangular routing*, that is, the communication between the MN and CN now has to go through the HA using tunneling. It can be quite efficient if the MN and CN are very close;

in an extreme case, both MN and CN can be in the same network while the HA is far away. To alleviate triangular routing, the CN can also keep the mapping between the mobile's HoA and CoA, and accordingly send packets to the mobile directly, without going through the HA. In this case, the mobile node must update its CoA to CNs as well.

Even with route optimization, Mobile IP can still introduce significant network overhead in terms of increased delay, packet loss, and signaling when the MNs change their point of attachment to network frequently or the number of MNs grows dramatically. Hierarchical Mobile IP (HMIP) (RFC 4140) is a simple extension that improves the performance by using a *Mobility Anchor Point* (MAP) to handle the movement of a MN in a local region. The MN, if supporting HMIP, obtains a *Regional CoA* (RCoA) and registers it with its HA as its current CoA; while RCoA is the locator for the mobile in Mobile IP, it is also its regional identifier used in HMIP. At the same time, the MN obtains a *Local CoA* (LCoA) from the subnet it attaches to. When moving within the region, the MN only updates the MAP with the mapping between its RCoA and LCoA. It reduces the burden of the HA by reducing the frequency of updates. The shorter delay between the MN and the MAP also improves response time.

17.4.2 Link-Layer Handoff Management

Link-layer handoff or handover occurs when a mobile device changes its radio channels to minimize interference under the same AP or BS (called *intracell handoff*) or when it moves into an adjacent cell (called *intercell handoff*). For example, in GSM, if there is strong interference, the frequency or time slot can be changed for a mobile device, which remains attached to the same BS transceiver. For intercell handoff, there are two types of implementations, namely, *hard handoff* and *soft handoff*.

Hard Handoff

As illustrated in Fig. 17.17, a hard handoff is triggered when the signal strength from the existing BS, perceived by the MN moving out of the cell, is below a threshold before connecting to the new BS. The MN occupies only one channel at a time: the channel in the source cell is released and only then the channel in the target cell is engaged. Hence, the connection to the source is broken before or as the connection to the target is made. For this reason, hard handoff is also referred to as *break before make*. To minimize the impact of the event, the operations have to be short that causes almost no user-perceptible disruption to the session. In the early analog systems, it could be heard as a click or a very short beep; in modern digital systems it is generally unnoticeable.

The implementation of hard handoff is relatively simple as the hardware does not need to be capable of receiving two or more channels in parallel. In GSM, the decision is done by the BS with mobile's assistance, which reports the signals strength back to the BS. The BS also knows the availability of channels in the nearby cells through

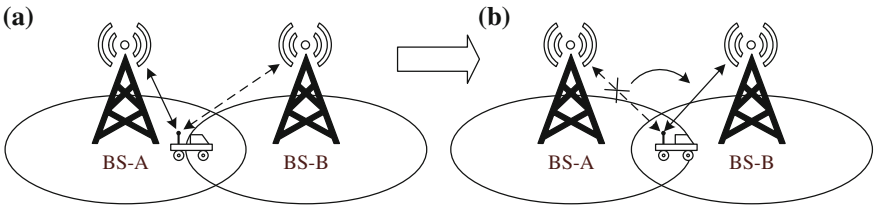


Fig. 17.17 An illustration of hard handoff. Before handoff, the mobile node is connected to BS-A, but the signal strength becomes weaker when it is moving toward BS-B. At a certain time when the signal strength of BS-B is above a threshold (and that of BS-A becomes very weak), a hard handoff decision will be made, such that the connection to BS-A is first broken and the connection to BS-B is then established. **a** Before hand off and **b** After hand off

information exchange. If the network decides that it is necessary for the mobile to hand off, it assigns a new channel and time slot to the mobile, and then informs the BS and the mobile of the change.

The ongoing session, however, can be temporarily disrupted if the hard handoff fails. Reestablishment is then needed, which may be noticeable to the users and sometimes could fail, causing a session to be terminated abnormally. Also when the mobile stays between BSs, it can bounce back and forth, causing an undesirable *ping-pong* phenomenon.

Soft Handoff

In a soft handoff, the channel in the source cell is retained and used for a while in parallel with the channel in the target cell. In this case, the connection to the target is established before the connection to the source is broken. The interval, during which the two connections are used in parallel, may be brief or substantial. For this reason, the soft handoff is also referred to as *make before break*, and is perceived by network engineers as a state of the call, rather than a instant event as in hard handoff.

One advantage of the soft handoff is that the connection to the source cell is broken only when a reliable connection to the target cell has been established, and therefore the chances that the call is terminated abnormally due to failed handoffs are lower. A soft handoff may involve using connections to more than two cells—connections to three, four, or more cells can be maintained at the same time; the best of these channels can be used for the call at a given moment or all the signals can be combined to produce a clearer copy of the signal. Since fading and interference in different channels are not necessarily correlated, the probability of them taking place at the same moment in all channels is very low. Thus the reliability of the connection becomes higher.

Soft handoff permits a smooth handoff that is critical to continuous media data flows. This advantage comes at the cost of more complex hardware in the device, which must be capable of receiving and processing several channels in parallel. This can be realized in CDMA or WCDMA through different transmission codes on different physical channels.

Vertical Handoff

A more interesting and complex handoff is between different types of networks, known as *vertical handoff* [25]. A typical example is between Wi-Fi and cellular networks, as the former is cheaper and fast and the latter is of broader and ubiquitous coverage. Switching between them, therefore, combines their advantages [26].

A typical vertical handoff consists three steps, namely, *system discovery*, *handoff decision*, and *handoff execution*. During the discovery phase, the mobile terminal determines which networks can be used. These networks may also advertise the supported data rates and the QoS parameters. In the decision phase, the mobile terminal determines whether the connections should continue using the current network or be switched to the target network. The decision may depend on various parameters or metrics including the type of the application (e.g., conversational or one-way streaming), the minimum bandwidth and delay required by the application, the transmit power, and the user's preferences. During the execution phase, the connections in the mobile terminal are rerouted from the existing network to the target network in a seamless manner.

The 3G networks support multimedia transmissions with a bitrate of 384 kbps for fast mobility to 2 Mbps for slow mobility, and the 4G achieves even higher rates up to 100 Mbps. It also allows global roaming across multiple networks with vertical handoffs, e.g., from a cellular network to a high-speed wireless LAN. To enable smooth transitions, besides the mobility solutions in the data-link and network layers, additional support from the transport and application layers with cross-layer optimizations are also needed. For example, if the transport layer or application layer is aware of a potential handoff, then prefetching could be executed by the BS in the target cell, which can avoid the potential service interruption, thus enabling continuous streaming.

17.5 Further Exploration

Rappaport [1], Goldsmith [27], and Tse and Viswanath [28] offer comprehensive and in-depth tutorials on the foundations of wireless communication. Viterbi [5] provides a solid analysis on spread spectrum and the foundation of CDMA. Wang et al. [29] give an in-depth discussion on error control in video communications.

17.6 Exercises

1. In the implementations of TDMA systems such as GSM, an FDMA technology is still in use to divide the allocated carrier spectrum into smaller channels. Why is this necessary?

2. We have seen a geometric layout for a cellular network in Fig. 17.4. The figure assumes hexagonal cells and a symmetric plan (i.e., that the scheme for splitting the frequency spectrum over different cells is uniform). Also, the reuse factor is $K = 7$. Depending on cell sizes and radio interference, the reuse factor may need to be different. Still requiring hexagonal cells, can all possible reuse factors achieve a symmetric plan? Which ones can? Can you speculate on a formula for general possible reuse factors?
3. Consider the hard handoff and soft handoff for mobile terminals moving across cells.
   (a) Why is a soft handoff possible with CDMA. Is it possible with TDMA or FDMA?
   (b) Which type of handoff works better with multimedia streaming?
4. Most of the schemes for channel allocation discussed in this chapter are fixed (or uniform) channel assignment schemes. It is possible to design a dynamic channel allocation scheme to improve the performance of a cellular network. Suggest such a dynamic channel allocation scheme.
5. The Gilbert-Elliott two-state Markov model has been widely used in simulations to characterize wireless errors, as illustrated in Fig. 17.2.
   (a) Given the state transition probabilities p_{00}, p_{11}, p_{10}, and p_{01}, calculate steady-state probability P_0 and P_1 that the wireless channel is in state 0 and state 1, respectively.
   (b) Write a simple program to simulate the process. Run it for a long enough time and calculate the average length of error bursts. Discuss how it would affect multimedia data transmission.
6. Consider a wireless network whose signal does not decay dramatically, i.e., within the network range, the signal strength is always high enough. However, the signal can be blocked by physical barriers. Will this network have the hidden terminal problem? Briefly explain your answer.
7. In today's networks, both the transport layer and link-layer implement error detection mechanisms. Why do we still need error detection in the link layer given that the transport layer protocol, say TCP, assumes that the lower layers of a network is unreliable and seeks to guarantee reliable data transfer using error detection and retransmission? Hint: Consider the performance gain.
8. Discuss the error detection and correction capability of the two-dimensional parity check.
9. Calculate the Internet checksum of the following message: 10101101 01100001 10001000 11000001
10. Consider Cyclic Redundancy Check (CRC).
    (a) Assume the keyword, K, is 1001, and the message M is 10101110. What is the width (in bits) of the CRC bits, R? What is the value of R? Please give detailed calculations.
    (b) Prove that $M \cdot 2^r \oplus R$ is perfectly divisible by K, and verify it using the M, K, and R values above.

11. Discuss why interleaving increases the delay in decoding? Will interleaving be effective if the loss is uniformly distributed?
12. H.263+ and MPEG-4 use RVLCs, which allow decoding of a stream in both forward and backward directions from a synchronization marker.
    (a) Why is decoding from both directions preferred?
    (b) Why is this beneficial for transmissions over wireless channels?
    (c) What condition is necessary for the codes to be reversibly decodable? Are these two set of codes reversible: (00, 01, 11, 1010, 10010) and (00, 01, 10, 111, 110)?
    (d) Why are RVLCs usually applied only to motion vectors?
13. Suggest two error concealment methods for audio streaming over wireless channels.
14. There is a broad spectrum of device and user mobility, in terms of both range and speed, as illustrated in Fig. 17.14. Discuss the challenges in the different mobility scenarios, and the potential solutions.
15. To alleviate triangular routing, a CN can also keep the mapping between the mobiles HoA and CoA, and accordingly encapsulate packets to the mobile directly, without going through the HA.
    (a) In which scenario does this *direct routing* solution work best?
    (b) Discuss any potential problem with the direct routing solution.
    (c) Propose another solution that addresses the triangular routing problem. Discuss its pros and cons.

References

1. T.S, Rappaport, *Wireless Communications: Principles and Practice*, 2nd edn. (Pearson Education, Upper Saddle River, 2010)
2. H.-S. Wang, N. Moayeri, Finite-state markov channel-a useful model for radio communication channels. IEEE Trans. Veh. Technol **44**(1), 163–171 (1995)
3. E.N. Gilbert, Capacity of a burst-noise channel. Bell Syst. Tech. J. **29**, 147 (1960)
4. M. Rahnema, Overview of GSM system and protocol architecture. IEEE Commun. Mag. **31**(4), 92–100 (1993)
5. A.J. Viterbi, *CDMA: Principles of Spread Spectrum Communication* (Addison Wesley Longman, Redwood City, 1995)
6. M. Baker, From LTE-advanced to the future. IEEE Commun. Mag. **50**(2), 116–120 (2012)
7. C. Zhang, S.L. Ariyavisitakul, M. Tao, LTE-advanced and 4g wireless communications. IEEE Commun. Mag. **50**, 102–103 (2012)
8. M.2134 - requirements related to technical performance for IMT-advanced radio interface(s). Technical report, ITU-R (2008)
9. Agilent Technologies, M. Rumney, *LTE and the Evolution to 4G Wireless: Design and Measurement Challenges* (Wiley, 2013)
10. J. Burkhardt et al., *Pervasive Computing: Technology and Architecture of Mobile Internet Applications* (Addison Wesley Professional, 2002)

11. E. Perahia, R. Stacey, *Next Generation Wireless LANs: 802.11n and 802.11ac* (Cambridge University Press, New York, 2013)
12. L. Harte, *Introduction to Bluetooth*, 2nd edn. (Althos, 2009)
13. Third Generation Partnership Project 2 (3GPP2). Video conferencing services - stage 1. *3GGP2 Specifications*, S.R0022 (2000)
14. Third Generation Partnership Project (3GPP). QoS for speech and multimedia codec. *3GPP Specifications*, TR-26.912 (2000)
15. A. Houghton, *Error Coding for Engineers* (Kluwer Academic Publishers, Boston, 2001)
16. T.K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms* (Wiley-Interscience, 2005)
17. J.F. Kurose, K.W. Ross. *Computer Networking: A Top-Down Approach*, 6th edn. (Pearson, New York, 2012)
18. E.K. Wesel, *Wireless Multimedia Communications: Networking Video, Voice, and Data* (Addison-Wesley, Reading city, 1998)
19. K.N. Ngan, C.W. Yap, K.T. Tan, *Video Coding For Wireless Communication Systems* (Marcel Dekker Inc, New York, 2001)
20. Y. Takishima, M. Wada, H. Murakami, Reversible variable length codes. IEEE Trans. Commun. **43**(2–4), 158–162 (1995)
21. C.W. Tsai, J.L. Wu, On constructing the Huffman-code-based reversible variable-length codes. IEEE Trans. Commun. **49**(9), 1506–1509 (2001)
22. Y. Wang, Q.F. Zhu, Error control and concealment for video communication: a review. Proc. IEEE **86**(5), 974–997 (1998)
23. D. Le, X. Fu, A review of mobility support paradigms for the Internet. IEEE Commun. Surv. Tutorials **8**(1), 38–51 (2006)
24. D. Saha, A. Mukherjee, I.S. Misra, M. Chakraborty, Mobility support in ip: a survey of related protocols. IEEE Netw. **18**(6), 34–40 (2004)
25. J. McNair, F. Zhu, Vertical handoffs in fourth-generation multinetwork environments. IEEE Wireless Commun. **11**(3), 8–15 (2004)
26. J. Sommers, P. Barford. Cell vs. wifi: on the performance of metro area mobile connections. In Proceedings of the 2012 ACM Conference on Internet Measurement Conference (IMC '12), pp. 301–314, New York, 2012
27. A. Goldsmith, *Wireless Communications*. (Cambridge University Press, 2005)
28. D. Tse, P, Viswanath, *Fundamentals of Wireless Communication* (Cambridge University Press, New York, 2005)
29. Y. Wang, J. Ostermann, Y.Q. Zhang, *Video Processing and Communications* (Prentice Hall, Upper Saddle River, 2002)