Video Streaming in Wireless Internet

Frank Fitzek and Patrick Seeling and Martin Reisslein

Contents

List of Figures

List of Tables

About the Authors

Dr. Frank Fitzek

Frank H. P. Fitzek received his diploma (Dipl.-Ing.) degree in electrical engineering from the University of Technology - Rheinisch-Westflische Technische Hochschule (RWTH) - Aachen, Germany, in 1997 and his Ph.D. (Dr.-Ing.) in Electrical Engineering from the Technical University Berlin, Germany in 2002. He co-founded the start-up company acticom GmbH in Berlin in 1999. At acticom GmbH he is currently involved in the development of advanced wireless IP services for WLAN and 3G. His current research interests are in the areas of QoS support for voice and video over wireless networks, robust header compression, security in self organizing networks, and the integration of mobile ad-hoc networks in cellular systems. Simultaneously, he is teaching at the University of Ferrara. Dr. Fitzek is member of the IEEE and serves on the Editorial Board of the IEEE Communications Surveys and Tutorials.

Dipl.-Ing. Patrick Seeling

Patrick Seeling is a graduate student in the Department of Electrical Engineering at Arizona State University. He received the Dipl.-Ing. degree in Industrial Engineering and Management from the Technical University of Berlin (TUB), Germany, in 2002. His research interests are in the area of video communications in wired and wireless networks. He is a student member of IEEE.

Dr. Martin Reisslein

Martin Reisslein is an Assistant Professor in the Department of Electrical Engineering at Arizona State University, Tempe. He received the Dipl.-Ing. (FH) degree from the Fachhochschule Dieburg, Germany, in 1994, and the M.S.E. degree from the University of Pennsylvania, Philadelphia, in 1996. Both in electrical engineering. He received his Ph.D. in systems engineering from the University of Pennsylvania in 1998. During the academic year 1994-1995 he visited the University of Pennsylvania as a Fulbright scholar. From July 1998 through October 2000 he was a scientist with the German National Research Center for Information Technology (GMD FOKUS), Berlin. While in Berlin he was teaching courses on performance evaluation and computer networking at the Technical University Berlin. He has served on the Technical Program Committees of IEEE Infocom and IEEE Globecom. He is Editor-in-Chief of the IEEE Communications Surveys and Tutorials. He maintains an extensive library of video traces for network performance evaluation, including frame size traces of MPEG–4 and H.263 encoded video, at http://www.eas.asu.edu/trace. He is co-recipient of the Best Paper Award of the SPIE Photonics East 2000 — Terabit Optical Networking conference. His research interests are in the areas of Internet Quality of Service, video traffic characterization, wireless networking, and optical networking.

1 Introduction

Video services are becoming an integral part of future communication systems. Especially for the upcoming 3G wireless networks such as UMTS, video may very well turn out to be the key value addition that achieves the required return of investment. While previous generations of wireless communication systems were primarily designed and used for voice services, next generation systems have to support a broad range of applications in a wide variety of settings. Novel wireless applications such as telematics and fleet management have introduced wireless networking to the enterprise domain. At the same time the private market sector is booming with the availability of low–priced wireless equipment. The early market stages were characterized by the needs of early adopters, mostly for professional use. As the market matures from the early adopters to normal users, new services will be demanded. These demands will likely converge toward the demands that exist for wired telecommunications services. These demands, also referred to as "the usual suspects", are comprised of a variety of different services — including Internet access for browsing, chatting, and gaming. In addition, entertainment services such as television, cable–TV, and pay–per view movies are demanded. With the availability of wireless services, the location is no longer of importance to private users. Thus, the demand of mobile users, connected over wireless networks will approach this mixture of services. With the omnipresence of wireless services, the usage schemes will become independent from location and connection type. One example for such a wireless service is mobile gaming. Private users, who are waiting at an airport or elsewhere, are able to join Internet–based multi-player games to bridge time gaps. Among the different entertainment services, mobile video will likely account for a large portion of the entertainment services, as are cinemas, video rentals, and television now. This application scenario covers wireless entertainment broadcasting and video on demand. A second, professional application of video in the wireless domain is telemedicine, where remote specialists are enabled to respond to emergencies. The wide area of video services in wireless environments as well as the expectations for wireless communication systems call for an understanding of the basic principles of wireless video streaming.

Generally, the video delivered to wireless users is either (*i*) *live* video, e.g., the live coverage of a sporting event, concert, or conference, or (*ii*) *prerecorded* (stored) video, e.g., a TV show, entertainment movie, or instructional video. Some videos fall in both categories. For instance, in a typical distance learning system, the lecture video is available to distance learners live, i.e., while the lecture is ongoing, and also as stored video, i.e., distance learners can request the lecture video later in the day or week from a video server.

In general, there are two ways to deliver video over a packet-switched network (including packet-oriented wireless networks): (*i*) *file download*, or (*ii*) *streaming.* With file download the entire video is downloaded to the user's terminal before the playback commences. The video file is downloaded with a conventional reliable transport protocol, such as TCP. The advantage of file download is that it is relatively simple and ensures a high video quality. This is because losses on the wireless links are remedied by the reliable transport protocol and the play–out does not commence until the video file is downloaded completely and without errors. The drawback of file download is the large response time, typically referred to as *start–up delay.* The start–up delay is the time from when the user requests the video until playback commences. Especially for large video files and small bandwidth wireless links, the start-up delay can be very large.

With video streaming, on the other hand, playback commences before the entire file is downloaded to the user's terminal. In video streaming typically only a small part of the video ranging from a few video frames to several hundreds or thousands of frames (corresponding to video play back durations on the order of hundreds of milliseconds to several seconds or minutes) are downloaded before the streaming commences. The remaining part of the video is transmitted to the user while the video playback is in progress. One of the key trade–offs in video streaming is between the start-up delay and the video quality. That is, the smaller the amount of the video that is downloaded before streaming commences, the more the continuous video playback relies on the timely delivery of the remaining video over the unreliable wireless links. The errors on the wireless links may compromise the quality of the delivered video in that only basic low quality (and low bit rate) video frames are delivered or some video frames are skipped entirely. Thus, video streaming gives the user shorter start-up delays at the expense of reduced video quality. The challenge of video streaming lies in keeping the quality degradation to a level that is hardly noticeable or tolerable while utilizing the wireless resources efficiently (i.e., supporting as many simultaneous streams as possible).

We note that file download and streaming with some start-up delay are suitable only for prerecorded video. The delivery of live video, on the other hand, requires an extreme form of video streaming with essentially no pre-playback download. Another consideration in the delivery of prerecorded video is user interaction (i.e., VCR functions such as fast forward, pause and rewind). Some of these interactions may result in a new start-up delay in video streaming.

In this book chapter we introduce video streaming in wireless environments. We first give an introduction to the world of digital video, and outline the fundamental need for compression. Next, we introduce the basics of video compression. We show how video compression is achieved by exploiting different types of redundancies in the raw (uncompressed) video stream. We do this by following one sample video sequence on its way from the raw video to the compressed video stream. This sample video sequence is called *Highway*, and is publicly available [4]. Next, we introduce the characteristics of the wireless communication channel and study the impact of the wireless link errors on the video quality. Finally, we discuss the different protocols and adaption techniques for streaming the video content over the wireless channel.

The model of a transmission chain of a general wireless communication system for video streaming is given in Figure 1. At the sender side the video source is passed to the video (source) encoder. The compressed video stream is passed to the transport process which in turn passes the stream plus some overhead information to the channel coder and modulation part for transmission over the wireless link. At the receiver side the process is reversed.

2 Basics of Video Compression

Video compression is undergoing constant changes as new coding/decoding (codec) systems are being developed and introduced to the market. Nevertheless, the internationally standardized video compression schemes (see subsection 2.5), such as the H.26x and MPEG-n standards, are based on a common set of fundamental encoding principles. In this section we give an overview of these encoding principles. For more details on video compression, we refer the interested reader to [5, 6, 7, 8, 9, 10, 11].

2.1 Digital Video

The sizes of the pictures in the current video formats are illustrated in Figure 2. Note that the ITU-R/CCIR 601 format (i.e., the common TV image format) and the CIF and QCIF have the same ratio of width to height. In contrast, the High Definition Television (HDTV) image format has a larger width to height ratio, i.e., is perceived as "wider". Each individual image is composed of picture elements (usually referred to as pixels or pels). The specific width and height (in pixels) in of the different formats are summarized in Table 1. Today, typical formats for wireless video are QCIF (176 × 144 pixel) and CIF (352 × 288 pixel).

A single pixel can be represented in two different color spaces: RGB and YUV. In the RGB color space, the pixel is composed of the three colors **R**ed, **G**reen, and **B**lue. In the YUV color space, the pixel is represented by its luminance **Y**, chrominance **U**, and saturation **V**. The U and V components often referred to as the chrominance values. The RGB color space is typically used for computer displays while the YUV color space is common for TV sets. TV sets convert the incoming YUV signal to RGB while displaying the picture. The YUV color representation was necessary to broadcast color TV signals while allowing the old black and white TV sets to function without modifications. As the luminance information is located in different frequency bands, the old tuners are capable to tune in on the Y signal alone. The conversion between these two color spaces is defined by a fixed conversion matrix. Most common video compression schemes take the YUV color space as input and we therefore focus on this color space for the remainder of this section.

In digital video, the (analog) values of the three components Y, U, and V (or R, G, and B) are quantized to 8 bit representations, i.e., there are three bytes representing each pixel. The human eye is far more sensitive to changes in luminance than to the two chrominance components. It is therefore common to reduce the information that is stored per picture by *chrominance sub–sampling.* In the unsampled YUV format, also referred to as YUV 4:4:4 format every pixel is represented by three bytes, as just noted. With sub–sampling the ratio of chrominance to luminance bytes is reduced. More specifically, sub-sampling represents a group of typically four pixels by their four luminance components (bytes) and one set of two chrominance values. Each of these two chrominance values is typically obtained by averaging the corresponding chrominance values in the group. In case the four pixels are grouped as a block of 2×2 pixels, the format is YUV 4:2:0. If the grouped pixels are forming a line of 4×1, the format is referred to as YUV 4:1:1. These two most common YUV sampling formats are illustrated in Figures 3 and 4. Thus the size of one YUV frame with 4:2:0 (or 4:1:1) chrominance sub–sampling in the QCIF format (176 pixel columns by 144 pixel rows for the luminance) is

$$176 \cdot 144 \cdot \left(8 \text{ bit} + \frac{2 \cdot 8 \text{ bit}}{4}\right) = 304,128 \text{ bit} = 38,016 \text{ byte}. \tag{1}$$

The frame sizes and data rates for the different video formats and frame rates are summarized in Table 1. Note that although chrominance sub–sampling reduces the bit rate already significantly, the sub–sampled video is commonly referred to as uncompressed (raw) video, as it is the input to the video encoding (compression). For television applications, YUV 4:2:2 is used. This format samples the chrominance for every second luminance value. Several different formats for the sequence of sampling and storage exist.

The different frame rates of 25 frames per second in PAL video and 30 frames per second in NTSC video are due to the different frequencies in the power supplies in Europe/Asia and the U.S. Given the enormous bit rates of the raw video streams and the limited bandwidths provided by wireless links, it is clear that some form of compression is required to make wireless video viable.

In general digital video is not processed, stored, compressed and transmitted on a per–pixel basis, but in a hierarchy [10] as illustrated in Figure 5. At the top of this hierarchy is the *video sequence*, which is divided into *Groups of Pictures (GoPs)*. Each GoP in turn consists of multiple *video frames*. A single frame is divided into *slices*. Each slice consists of several *macroblocks (MBs)*, each typically consisting of 4×4 *blocks*. Each block typically consists of 4×4 *pixels*.

Due to the large bit rates, digital video is almost always encoded (compressed) before transmission over a packet-oriented network. The limited bandwidths of wireless links make compression especially important. Video compression generally exploits three types of redundancies [10]. On a per–frame basis (i.e., single picture), neighboring pixels tend to be correlated and thus have *spatial* redundancy [11]. Intra-frame encoding is employed to reduce the spatial redundancy in a given frame. In addition, consecutive frames have similarities and therefore *temporal* redundancy. These temporal redundancies are reduced by inter-frame coding techniques. The result of the reduction of these two redundancies is a stream of codewords (symbols) that has some redundancy at the symbol level. The redundancy between these symbols is reduced by variable length coding before the binary code is passed on to the output channel. [1] The elimination of these redundancies is explained in the following three subsections.

2.2 Intra-frame Coding

The intra-coding (compression) of an individual video frame resembles still picture encoding. It is commonly based on the discrete cosine transformation (DCT). (Wavelet–based transformation schemes have also emerged. Studies indicate that in the field of video encoding, the wavelet–based approach does not improve the quality of the transformed video significantly [12]. However, essentially all internationally standardized video compression schemes are based on the DCT and we will therefore focus on the DCT in our discussion.)

[1] Additional compression schemes, such as the exploitation of object recognition techniques, are also in development, but not commonly applied up to now.

The intra-frame coding proceeds by partitioning the frame into blocks, also referred to as block scanning. The size of these blocks today is typically 8×8 pixels (previously, also 4×4 and 16×16 were used). The DCT is then applied to the individual blocks. The resulting DCT coefficients are quantized and zig–zag–scanned according to their importance to the image quality. An overview of these steps is given in Figure 6.

2.2.1 Block Scanning

In order to reduce the computational power required for the DCT, the original frame is subdivided into blocks, since efficient algorithms exist for a block–based DCT [13]. The utilization of block–shapes for encoding is one of the limitations for DCT–based compression systems. The typical object shapes in natural pictures are irregular and thus cannot be fitted into rectangular blocks, as illustrated in Figure 7. In order to increase the compression efficiency, different block sizes can be utilized at the cost of increased complexity [14].

2.2.2 Discrete Cosine Transformation

The DCT is used to convert a block of pixels (e.g., for the luminance component 8×8 pixels, represented by 8 bits each, for a total of 256 bits) into a block of transform coefficients. The transform coefficients represent the spatial frequency components of the original block. An example for this transformation of the block marked in Figure 7 is illustrated in Figure 8.

This transformation is lossless, it merely changes the representation of the block of pixels, or more precisely the block of luminance (chrominance) values. A two–dimensional DCT for an $N \times N$ block of pixels can be described as two consecutive one–dimensional DCTs (i.e., horizontal and vertical). With $f(i,j)$ denoting the pixel values and $F(u,v)$ denoting the transform coefficients we have

$$F(u,v) = \frac{2}{N} \cdot C(u) \cdot C(v) \cdot \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) \cos\left(\frac{(2i+1)\,u\pi}{2N}\right) \cos\left(\frac{(2j+1)\,v\pi}{2N}\right), \quad (2)$$

where

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}}, & x = 0 \\ 1, & \text{otherwise.} \end{cases} \tag{3}$$

The lowest order coefficient is usually referred to as the DC–component, whereas the other components are referred to as AC–components.

2.2.3 Quantization

In a typical video frames the energy is concentrated in low frequency coefficients. That is, a few coefficients with u and v close to zero have a high significance for the representation of the original block. On the other hand, most higher frequency coefficients (i.e., $F(u,v)$'s for larger u and v) are small. In order to compress this spatial frequency representation of the block, a quantization of the coefficients is performed. Two factors determine the amount of compression and the loss of information in this quantization:

1. Coefficients $F(u,v)$ with an absolute value smaller than the *quantizer threshold* T are set to zero, i.e., they are considered to be in the so-called "dead zone".
2. Coefficients $F(u,v)$ with an absolute value larger than or equal to the quantizer threshold T are divided by twice the *quantizer step size* Q and rounded to the nearest integer.

In summary, the quantized DCT coefficients $I(u,v)$ are given by

$$I(u,v) = \begin{cases} 0 \text{ for } \mid F(u,v) \mid < T \\ \left[\frac{F(u,v)}{2Q}\right] \text{ for } \mid F(u,v) \mid \geq T, \end{cases} \tag{4}$$

where[·] denotes rounding to the nearest integer.

A quantizer with $T = Q$, as typically used in practice, is illustrated in Figure 9.

Figure 13 continues the example from Figure 8 and shows the quantized values for $T = Q = 16$. As illustrated here, typically many DCT coefficients are zero after quantization [9]. The larger the step size, the larger the compression gain — as well as the loss of information [15]. The trade–off between compression and decodable image quality is controlled by setting the quantizer step size (and quantizer threshold) [10]. This trade-off between image quality and compression (frame size in bytes after quantization) is illustrated for one frame from the *Highway* test sequence [4] in Figures 10, 11, and 12. The frame was encoded with a fixed quantizer step size Q and subsequently decoded. Notice that the quality of the video frame visibly decreases as Q increases. As can be seen from the frame sizes, the quality loss is also reflected in the amount of data needed. We also report the *peak signal to noise ratio (PSNR)* for these encodings. The PSNR is a commonly employed objective quality metric, see Subsection 3.1 for details. As illustrated here, a smaller PSNR value corresponds to a worse image quality.

The Figures 10, 11, and 12 represent the encoding result without rate control. Rate control is applied during the encoding process to adjust the resulting video frame sizes to the bandwidth available. The quantization is adjusted in a closed–loop process (i.e., the result of the quantization is measured for its size and as required encoded again with a different quantizer step size) to apply a compression in dependence of video content and the resulting frame size. The result is a constant bit rate (CBR) video stream but with varying quantization and thus quality. The opposite of CBR is variable bit rate (VBR) encoding. Here the quantization process remains constant, thus it is referred to as open–loop encoding (i.e., the result of the quantization process is no longer subject to change in order to meet bandwidth requirements). To achieve a constant quality, VBR encoding has to be used [16, 17].

2.2.4 Zig–Zag Scanning

The coefficient values obtained from the quantization are scanned by starting with the DC–component and then continuing to the higher frequency components in a zig–zag fashion, as illustrated in Figure 13. The zig–zag scanning facilitates the subsequent variable length encoding by encountering the most likely non–zero elements first. Once all non-zero coefficients are scanned, the obtained sequence of values is further encoded to reduce codeword redundancy, see Section 2.4. The scanning can be stopped before collecting all quantized non-zero coefficients to achieve further (lossy) compression.

2.3 Inter-frame Coding: Motion Estimation and Compensation

Video encoders commonly employ inter-frame coding to reduce the temporal redundancy between successive frames (images). The basic idea of inter-frame coding is that the content of a given current video frame is typically similar to a past video frame. The past frame is used as a reference frame to predict the content of the current frame. This prediction is typically performed on a macroblock or block basis [18]. For a given actual block in the current frame a *block matching algorithm* (BMA) searches for the most similar prediction block in the past frame, as illustrated in Figures 14 and 15. The goal of the search is to determine the *motion vector*, i.e., the displacement of the most similar prediction block from the actual block. This search, which is also referred to as motion estimation, is performed over a specific range of n pixels around the location of the block in the current frame. Several different matching (similarity) criteria such as the cross-correlation function, mean squared error, or mean absolute error can be applied. To find the best match by full search, $(2n + 1)^2$ comparisons are required. Several fast motion estimation schemes such as the *three step search* [19] or the *hierarchical block matching algorithm* [20] have evolved to reduce the processing. Once the motion vector is determined, the difference between the prediction block and the actual block is encoded using the intra-frame coding techniques discussed in the preceding section. These differences may be due to lighting conditions, angles, and other factors that slightly change the content of block. These differences are typically small and allow for efficient encoding with the intra-frame coding techniques (and variable length coding techniques). The quantizer step size can be set independently for the coding of these differences. The encoding of these differences accounts for the differences between the prediction block and the actual block, and is referred to as *motion compensation.* The inter–coded frame is represented by (i) the encoded error or difference between the current frame and a previously transmitted reference frame (motion compensation), and (ii) the motion vectors (motion estimation).

2.3.1 Concept of I, P, B frames

Blocks in video frames often reveal parts of the background or scene that were not visible before the actual frame [9]. Motion vectors of these areas can therefore not be found by referencing previous frames, but only by considering also future frames, as illustrated in Figure 16. For this reason, inter-frame coding often considers both prediction from past reference frames as well as future reference frames.

There are three basic methods for encoding the original pictures in the temporal domain: I (Intra), P (Inter), and B (Bi–directional), as introduced in the MPEG-1 standard [8]. These encoding methods are applied on the frame or block level, depending on the codec. The inter–coded frames use motion estimation relying on the previous inter– or intra–coded frame. The bi–directional encoded frames rely on a previous as well as a following intra– or inter–coded frame. Intra–coded frames or blocks are not relying on other video frames and thus are important to stop error propagation. The sequence of frames between two intra–coded frames is referred to as a *Group of Pictures* (GoP). The relationship between these different encoding types and how frames rely on each other in a typical MPEG frame sequence is illustrated in Figure 17.

2.4 Variable Length Coding

The purpose of variable length coding is to reduce the statistical redundancy in the sequence of codewords obtained from zig–zag scanning an intra-coded block (or block of differences for a predicted block). Short codewords are assigned to values with high probabilities. Longer codewords are assigned to less probable outcomes of the quantization. The mapping between these original values and the code symbols is done within the variable length coding (VLC). The mapping has to be known by both, the sender and the receiver. As shown before, the quantization and zig–zag scanning result in a large number of zeros. These values are encoded using run–level coding. This encoding only transmits the number of zeros instead of the individual zeros. In addition, when no other values are trailing the coefficients with zeros (and this is the most likely case), an End of Block (EOB) codeword is inserted into the resulting bitstream. Huffman coding [21] and

Arithmetic coding [22, 21] and their respective derivatives are used to implement VLC. Huffman coding is fairly simple to implement, but achieves lower compression ratios. On the other hand, arithmetic coding schemes are computationally more demanding, but achieve better compression. As processing power is abundant in many of today's systems, newer codecs mostly apply arithmetic coding [10, 23].

2.5 ISO/MPEG and ITU–T Standards for Video Encoding

We close out this section by giving a quick overview of video standards. Despite a large variety of video coding and decoding systems (e.g., the proprietary Real–Media codec etc.), standardization on an international level is performed by two major bodies: ITU–T and ISO/MPEG. The early H.261 codec of the ITU–T was focused on delivering video over ISDN–networks with a fixed bitrate of $n \times 64$kbit/s, where n denotes the number of multiplexed ISDN–lines. From this starting point, codecs were developed for different purposes such as the storage of digital media or delivery over packet–oriented networks. The latest codec development H.26L (or MPEG–4 Annex 10, H.264/AVC) is still under way[2] and performed by a joint team of the ITU–T and the ISO/MPEG bodies (JVT). The evolving standards achieved better quality (in terms of PSNR, see Subsection 3.1) with lower bit rates and thus better rate–distortion performance. Figure 18 sketches an overview of the video standards development to date.

[2] At the time of writing.

3 Traffic and Quality Characteristics of Encoded Video

Having outlined the encoding (compression) of video in the preceding section we now turn our attention to the output of the video encoder. In particular we study the traffic characteristics of the encoded video stream as well as the video quality. The video traffic is governed by the sizes (in bytes or bits) of the individual video frames and the frame periods (display times of the individual video frames). For NTSC video without any skipped frames, the frame period is the reciprocal of the fixed frame rate, i.e., the frame period is 1/30 frames/sec = 33.33 msec. For networking research purposes, the traffic of encoded video is often recorded in *video traces*, which give the frame number (index), type (e.g., I, P, or B), playout time (when frame is decoded and displayed on screen), and the frame size (length), as illustrated by the following excerpt of a video trace of an H.26L encoding [24]:

```
Frame No.  Frametype  Time [ms]  Length [byte]
0          I          0.0        7969
1          P          120.0      5542
...        ...        ...        ...
```

Video traces are available for a few existing video standards [25, 26, 27, 28]. The traces are the basis for video traffic studies and video traffic modeling. The traces are also used to drive simulations of video streaming protocols and mechanisms. (Parsers that read the traces into commonly used discrete event simulator packages (e.g. NS [29], Omnet++ [30], or Ptolemy [31]) are available [32].)

Continuing the example used throughout this chapter, Table 2 gives elementary statistics for the video frame sizes and bit rates of a QCIF *Highway* encoding. The statistical analysis of the video traffic typically includes also metrics capturing the correlations in the traffic and the long range dependence (self-similarity) properties of the traffic, which are beyond the scope of this chapter.

Figure 19 gives a plot of the frame sizes averaged over GoPs as a function of time. Notice that the frame size hovers around 5,500 byte with occasional peaks reaching up to 7,000 and 8,000 bytes. This behavior is governed by the video content, as illustrated by three screenshots. The highlighted regions represent different content dynamics. The regions highlighted by eclipses have monotone content dynamics and result in roughly constant frame sizes. In contrast, the street sign and the bridge increase the content dynamics and result in larger frame sizes. Note that the fraction of the dynamic content in the frame influences the frame sizes. The street sign is smaller in size than the bridge which crosses the entire screen — and results in a smaller frame size peak than the bridge. Video frame sizes relate to the content of the video, different encoding methods, and encoder settings. These dependencies cause the frame sizes to have long–range dependencies and complicate the modeling of video traffic.

We note that the traffic variabilities in the *Highway* sequence considered here are relatively small (peak-to-mean ratio of 3.24 and coefficient of variation of 0.25), due to the overall fairly monotone content. Many typical entertainment videos are significantly more variable with frame size peak-to-mean ratios of ten or more and coefficients of variation close to one. Accommodating this highly variable (bursty) traffic in a packet-switched network is very challenging. In wireless video streaming the second major challenge is to overcome the highly variable wireless link errors, see Section 4.

3.1 Objective Video Quality Measurements

In order to estimate the video quality, an approach is needed which compares the reconstructed frame at the receiver side to the original frame. The peak signal to noise ratio (PSNR) is the most commonly utilized metric. Other algorithms for objective video frame quality assessment exist but are not proven to achieve better results than the PSNR [33]. The PSNR represents the objective video quality for each video frame by a single number. Since the human visual systems is more sensitive to the luminance component than the chrominance components, the PSNR is typically evaluated only for the Y (luminance) component. For a video frame is composed of $N \cdot M$ pixels, the mean squared error (MSE) and the PSNR in decibels are computed as

$$
\begin{aligned}
MSE &= \frac{\sum\limits_{\forall i,j} [f(i,j) - \hat{f}(i,j)]^2}{N \cdot M} \qquad (5) \\
PSNR &= 20 \cdot \log_{10}\left(\frac{255}{\sqrt{MSE}}\right), \qquad (6)
\end{aligned}
$$

where $f(i,j)$ represents the luminance value of pixel (i,j) in the original source frame and $\hat{f}(i,j)$ the corresponding luminance value in the decoded frame. Some recent video traces include both the frame sizes as well as the PSNR frame qualities [34].

4 Wireless Channel Characteristics

Before we have a closer look at the protocols and mechanisms used for video streaming, we give a short introduction to the relevant characteristics of the wireless channel. A basic understanding of the impairments of the wireless channel is important, as the video streaming mechanisms have to compensate for (*i*) the errors on the wireless channels (discussed in this section), and (*ii*) the burstiness of the video traffic (discussed in Section 3). Wireless channels are typically time-varying and frequency selective, as illustrated in Fig. 20 and 21. These channel fluctuations are the result of a combination of attenuation, multi-path fading and shadowing [35]. (see Section 4.1). Reflections result in signals being transmitted over multiple paths between sender and receiver (see Section 4.2).

4.1 Free Space Propagation

The wireless signal is attenuated on its path from the transmitter to the sender, i.e., the power level (strength) of the received signal is lower than the transmit power level. The free space propagation model is used to predict the received signal strength at the receiver under the assumption that only one line of sight path between sender and receiver exists. In such a scenario, the power level of the received signal P_r depends on the transmitted power P_s, the distance d between sender and receiver, the antenna gain of sender and receiver, G_s and G_r, the wavelength λ, and the path loss L. The so-called free space equation [35, 36] gives the received signal power as

$$P_r(d) = \frac{P_s G_t G_r \lambda^2}{(4\pi d)^2 L}. \tag{7}$$

Note that the received power decreases with the square of the distance.

4.2 Multi–path Fading

Considering a mobile and wireless communication between one sender and one receiver, the sender–side signal arrives at the receiver's antenna over multiple paths. Assuming an omni–directional antenna at the sender–side the signal is transmitted in all directions. Therefore the sender's signal arrives typically over a direct path and multiple indirect paths. The direct path is called the **L**ine **o**f **S**ight *(LoS)* path, while all indirect paths are called the **N**on **L**ine **o**f **S**ight *(NLoS)* path. NLoS paths are caused by reflection, diffraction, and scattering of the signal off of mountains, buildings, and moving obstacles in the wireless environment. *Multi–path propagation* refers to the situation where multiple copies of the unique sender–side signals arrive at the receiver. Due to the reflections and diffractions the received multi–path signals differ in amplitude, phase, and delay. The signals arriving at different times at the receiver typically reduce the received signal strength Since symbols of the same signal interfere with each other this type of interference is called **I**nter **S**ymbol **I**nterference (*ISI*). The level of interference depends on delay spread of a multi–path signal. The delay spread is the time duration between the first incoming signal (LoS) and the last incoming signal with a significant power level. A typical example for multi–path communication is illustrated in Figure 22 with one LoS path and two NLoS paths. Suppose the sender transmits a signal at some time instance, and after τ_1 the LoS signal arrives at the receiver. The NLoS signals arrive after τ_2 and τ_3 (with $\tau_3 > \tau_2$) at the receiver. The delay spread in this example is $\tau_3 - \tau_1$. Because of the propagation in free space and the possible reflections the received signals differ in their amplitudes. In real wireless systems the delay spread and the number of paths strongly depend on the environmental setting. Figure 23 gives typical delay spreads for the settings *Rural Area*, *Hilly Terrain*, and *Bad Urban* [3].

We illustrate the impact of multi–path propagation with the following example. Suppose two different signals S_N and S_H with low and high rate are transmitted by the sender over the channel. While the a symbol of the low rate signal S_N has 400 time units, the symbol length of the high data channel is only 100 time units. Both signals are given in Figure 26, where the low data rate signal is on the left and the high data rate is on the right side. The characteristics of the multi–path channel are given in Figure 27. The signal is received with three copies at the receiver side. After the LoS signal two NLoS signals are received with a delay of 75 and 90 time units. Attenuation causes the signal to be received with a lower amplitude of 70%, 60%, and 20%. The three copies of the signals are given in Figure 28. Each of the copies is delayed and attenuated as described by the channel characteristics. If we have now a closer look to the received signals, we note that these are distorted compared to the original sequences. Furthermore it is easier to recognize and restore the low data rate signal than the high rate signal. By this simple example we see that our brain or an equalizer have to work harder to recognize high data signals. Thus, there is a limit for single carrier systems using higher data rates. (The effects of the ISI are mitigated in practice by employing equalizers, RAKE sender/receiver, and directional antennas.) We close this brief intuitive discussion of the multi–path propagation by noting that multi–path propagation can severely degrade the system performance. Multipath propagation in conjunction with the *Doppler–Effect* due to the velocity of the mobile terminal the causes changes in the received power of about $30 - 40$dB compared to the mean received power [37].

4.3 Shadowing Fading

Shadow fading is caused by obstructions, such as buildings and vegetation. These obstructions may block the signal from the wireless mobile terminal. Typically, the fading effect due to shadowing has a relatively long time scale (on the order of hundreds of msec to seconds), depending of course to a large degree of the velocity of the mobile terminal.

The combined fading effects typically degrade the signal power over long time period, resulting in *bursty* or correlated errors on the wireless link [38, 39]. These correlated link errors in turn result in bursty packet drops, i.e., several consecutive packets are dropped, on the wireless links. A popular model for this effect from the perspective of the higher protocol layers in the Gilbert-Elliot two state Markov chain model [40, 41]. The two states of this Markov chain represent the “bad” channel state, where packets are dropped with high probability and the “good” channel state where the packet drop probability is small. The underlying Markov chain captures the effect of bursty errors in that a bad channel in a given time slot is more likely followed by a bad channel than a good channel in the next time slot, and vice versa. Higher order Markov models with more states provide a refined model [42].

5 Impact of Wireless Channel Errors on Video Quality

Having discussed both the variability of the video traffic and the wireless channel we now briefly illustrate their combined effects on the video quality over an error–prone channel. We simulate this by inflicting random bit errors on the encoded video (Note that we do not explicitly introduce bursty errors, which would render the outcome differently. However, as decoders differ in their error–resilience, sometimes bit errors have the same effect as a complete frame being lost on the channel. Therefore we assume the outcome of both, bursty errors and decoder–dependent frame drops, as similar in this experiment.). In this experiment, no error corrections or retransmissions were applied. Two different encoding schemes [3] were used to show the differences of quality and bit rates. The illustrated results were obtained with an early version of the forthcoming H.26L standard [23] reference encoder. Figures 30 and 32 illustrate the average video frame PSNR at the prospective receiver as a function of the mean bit error probability. Figures 31 and 33 illustrate the respective encoded video traffic on the GoP–level. The first encoding scheme shown in Figures 30 and 31 consists only of intra–coded frames (i.e., the GoP–length is only one frame, the frame type pattern thus IIIIIIIII . . .). With the lack of differential encoding, the video traffic generated is seemingly high. The objective video quality drops as the error probability increases. The second GoP is using a different encoding scheme and applies advanced features of H.26L (i.e., the GoP–length

[3] The results for additional encoding schemes are available online at http://www.acticom.info and http://trace.eas.asu.edu

is 12 frames, with a frame type pattern of IBBPBBSPBBPBBI . . .). The SP–frame types are introduced in H.26L to add additional error resilience to the encoded video. The video traffic that is generated by the encoding is visibly lower than for the mere intra–encoded sequence. However, the objective video quality obtained from the decoding process is similar if not better than for the intra–coded sequence. The comparison with Figures 32 and 33 shows a very similar traffic characteristic. This is surprising, since differentially encoded sequences should be more sensitive to errors. This result is explained as follows. As shown in Subsection 2.3.1, different frame types have certain dependencies. For successful decoding of frames that are differentially encoded, the referenced frames have to be present at the time of decoding. If referenced frames are missing, successful decoding cannot be achieved for their depending frames. This effect is also referred to as error–propagation. Thus the impact of these referenced frames is higher than the impact of frames that are depending on them. Video sequences that are only intra–coded should thus be less error–sensitive than sequences with differential encoding (in the experiment shown, the SP–frames within the encoded video stream are reducing the effect of error–propagation). Contrarily, a second effect has to be regarded. The frame sizes of intra–coded sequences are higher than those of sequences that also feature differential inter–coding. With a given bit error probability, the larger intra–coded frames are more likely to be subject to transmission errors. These effects are visible vice versa at the intra– nad inter–coded GoP–scheme.

This comparison yields the result that differential encoding of video data is not introducing a severe change in the quality at the receiver–side. Additionally, the network benefits from a reduction of the bandwidth needed for the transmission of the encoded video sequence.

To compensate for errors on the network, three general methods of error handling exist. Forward error correction uses additional parity information to allow for correction of transmission errors. As errors in wireless transmissions tend to occur in bursts, the parity information may be lost with the original information. The second method is the adaption of retransmission schemes. However, retransmissions have to be handled carefully, especially in the domain of real–time streaming. Thirdly, error concealment techniques can be applied on the receiver–side of the video stream. We will show how these techniques are applied as we regard the protocols and streaming of video in the following sections.

6 Internet Protocol Stack for Video Streaming

In this section we move up the networking protocol stack and consider the protocols commonly used for wireless video streaming at the network (IP) layer and higher. The key enabling protocols for multimedia streaming over IP networks is the Real Time Protocol (RTP) in combination with the Real Time Control Protocol (RTCP). These run on top of the User Datagram Protocol (UDP) for data transport. At the same time, the Session Announcement Protocol (SAP), Session Initiation Protocol (SIP), and the Real Time Streaming Protocol (RTSP) are used for session management, as illustrated in Figure 34.

The Session Announcement Protocol (SAP), in conjunction with the SIP and/or RTSP protocols initiate the streaming of a video. SAP announces both, multicast and unicast sessions to a group of users. SIP initiates multimedia session in a client/server manner. An open issue is how the client retrieves the destination address. Possible solutions are that the address is well known or is provided by SAP. RTSP is simply a ”remote control” used to control unicast streams in a server/client manner. SIP, SAP, and RTSP use the Session Description Protocol (SDP) to describe the media content and run either over TCP or UDP protocol. (Note that SDP is missing in the figure, as it is not a real protocol but rather a language such as HTTP.)

After establishing a session the application is able to start the data exchange using the RTP protocol. RTP is used for data transmission while QoS information between sender and receiver is exchanged using RTCP. The underlying IP network may independently provide QoS and multicasting (multicasting is a part of IP, not of RTP).

6.1 Session Description Protocol (SDP)

SDP is used to describe a multimedia session. The SDP message contains a textual coding that describes the session, more specifically it gives 1.) the transport protocol used to convey the data, 2.) a type field to distinguish the media (video, audio, etc), and 3.) the media format (MPEG4, GSM, etc). Furthermore the SDP message may contain the duration of the session, security information (encryption keys), and the session name in addition to the subject information (e.g. Arielle (c) Disney). An example of the textual coding of an SDP message is the following:

```
v=0 // Version number
o=mjh 2890844526 2890842807 IN IP4 192.16.6.202 // Originator
s=Wireless Internet Demonstrator // session title
i=A seminar on Internet multimedia //session information
u=http://www.acticom.de //URL for more information
e=fitzek@acticom.de (Frank Fitzek) //  Email address to contact
c=IN IP4 224.2.17.12/127 // connection information
a=recvonly // attribute, this telling session is receive only
m=audio 1789 RTP/AVP 0 // PCM audio using RTP on port 1789
m=application 1990 udp wb // "wb" application on port 1990
a=orient:portrait // "wb" in portrait mode
m=video 2003 RTP/AVP 31 // H.261 video using RTP on port 2003
```

SDP messages can be carried in any protocol, including HTTP, SIP, RTSP, and SAP. Originally SDP was designed for the support of multicast sessions. The information relating to the multicast session was conveyed using SAP. More recently, SDP is also used in combination with SIP and RTSP.

6.2 Session Announcement Protocol (SAP)

The SAP [43] is used for advertising multicast sessions. In brief, SAP discovers ongoing multicast sessions and seeks out the relevant information to setup a session. (In case of a unicast session the setup information might be exchanged or known by the participants.) Once all the information required for initiating a session is known, SIP is used to initiate the session.

6.3 Session Initiation Protocol (SIP)

Signaling protocols are needed to create sessions between two or more entities. For this purpose the H.323 and the SIP protocol have been standardized by two different standardization committees. H.323 was standardized by the ITU. The IETF proposed the Session Initiation Protocol (SIP) specified in RFC 3261 [44]. In contrast to other signaling protocols, SIP is textual based such as SDP.

SIP is a client/server-oriented protocol and is able to create, modify, and terminate sessions with one or multiple participants. Multi–party conferencing is enabled through IP multicast or a mesh of unicast connections. Clients generate requests and transmit them to a SIP proxy. The proxy in turn typically contacts a SIP registrar to obtain the user's current IP address. Users register with the SIP registrar whenever they start up an SIP application on a device, e.g., PDA, laptop, etc. This allows the SIP registrar to keep track of the user's current IP address. With SIP it is thus possible to reach user's that are on the move, making SIP very relevant for wireless streaming.

Using the INVITE request a connection is setup. To release an existing connection a BYE request is used. Besides these two requests, further requests are OPTIONS, STATUS, ACK, CANCEL, and REGISTER. SIP reuses HTTP header fields to ease an integration of SIP servers with web servers. In the SIP terminology the client is called *user agent.* A host can simultaneously operate as client and as server.

The call identifiers used in SIP include the current IP addresses of the users wishing to communicate and the type of media encoding used, e.g., MPEG–4 in the case of video.

6.4 Real Time Streaming Protocol (RTSP)

RTSP [45] may be thought of as a "remote control" for media streaming. More specifically, it is used to implement interactive features known from the VCR, such as pause and fast-forward. RTSP has many additional functionalities, see [45] for detail, and has been adopted by RealNetworks.

RTSP exchanges RTSP messages over an underlying transport protocol, such as TCP or UDP. The RTSP messages are ASCII text and very similar to HTTP messages. RTSP uses out-of-band signaling to control the streaming.

6.5 Real Time Protocol (RTP)

Real **T**ime **P**rotocol (*RTP*) [46] is a transport mechanism for real–time data. It consists of two components: RTP and RTCP, the RTP Control Protocol. Both, RTP and RTCP typically run over UDP but can use any other packet oriented transport protocol. In a video conference audio and video streams are separated and transmitted over different RTP sessions using different UDP port addresses.

To send multimedia content with RTP, the host packetizes the media, adds content dependent header fields, the RTP header, and then passes the message to the underlying protocol layers. The content dependent header field informs the receiver about the employed video codec and the used parameters (as explained shortly in an example). The content dependent header field has variable length, depending on the specific content and encoding used. On the other hand, the RTP header illustrated in Figure 35 has a fixed structure and is always 12 bytes long.

VERSION (V) The version number of the RTP protocol, represented by 2 bits. Currently version number 2 is used.

PADDING (P) This flag indicates that padding is used if set to one. Padding might be used for encryption.

EXTENSION (X) This flag indicates whether a content dependent header is used, which is placed between the RTP header and the payload.

PAYLOAD TYPE (PT) This field identifies the format of the RTP payload.

SEQUENCE NUMBER The initial sequence number is chosen randomly and than increments by one for each RTP packet. A random starting number was chosen to complicate plain text attacks. By means of the sequence number lost packets can be detected or disordered packets can be detected and replaced.

TIMESTAMP Four octets are used to reflect the sampling time. This information is important to assure the correct play–out at the receiver side.

The CSRC counter field (CC), the marker field (M) and the source identifiers are beyond the scope of this discussion, see [46] for details.

Next, we take a closer look at the content dependent header used to identify among other things the used video codec.

RTP Payload Format Specification Example If the extension bit (X) is set in the RTP header a content dependent header is placed between the RTP header and the RTP payload. By means of an H.263 example we illustrate the usage of such a header. As illustrated in Figure 36, the H.263 payload header is placed between the RTP header and the H.263 bit-stream.

RFC 2190 specifies the payload format for encapsulating an H.263 bit stream in RTP. H.263 is an advancement of the H.261 video encoding scheme. Three different modes (namely Mode A, Mode B, and Mode C) are defined for the H.263 payload header. An RTP packet is forced to use only one of the three modes for H.263 video streams. The choice depends on the desired network packet size and H.263 encoding options employed, see [47] for details. Mode A supports fragmentation at the level of Group of Block (GOB) boundaries, while Mode B and Mode C support more fine-grained fragmentation at Macro-block boundaries. The modes have an impact on total packet size. Within this example we will refer only the Mode A. Mode A uses the payload header illustrated in Figure 37 and explained as follows.

F flag The different modes (A, B, and C) are indicated by this flag. In case the flag is not set Mode A is used, otherwise it depends on flag P which mode is used.

P flag H.263 defines the usage of PB Frames. In case this flag is set PB frames are used. Additionally, in case F is set Mode B is used if P is not set and Mode C if P is set.

SBIT field and EBIT field Specify the number of most/least significant bits that shall be ignored in the first/last data byte.

SRC field Specify the video format used. The values of the PTYPE field (bit 6 up to 8) of H.263 are used here.

I flag Flag is set in case inter coded video is used.

U flag This flag indicates whether the encoder used the unrestricted motion vector, an advanced motion compensation mechanism, see [47] for details.

S flag Using bit 11 of the PTYPE field.

A flag Using bit 12 of the PTYPE field.

R field Four bits are reserved and set to zero

DBQ field In case PB frames are not used this field contains only zeros. Otherwise the values of the DBQUANT field defined by H.263 are copied here.

TRQ field Temporal reference for B frames. This field contains only zeros if PB frames are not used.

TR field Temporal reference for P frames. This field contains only zeros if PB frames are not used.

As illustrated by this example, the content dependent header (payload header) contains information that the receiver uses to decode and display the video.

Video related RFCs Besides the RFC2190 multiple RTP packetization schemes for multimedia applications are given in various RFCs. To name only a few, we give a short overview of each of the related RFCs in the following. For more detailed information we refer to the individual RFCs.

The RTP packetization scheme for the CellB video encoding is described in RFC 2029. The Cell image compression algorithm supports variable bit-rate video coding [48]. CellB, derived from CellA, has been optimized for network-based video applications.

A description how to packetize an H.261 video stream for RTP transmission is given in RFC 2032. The ITU–T Recommendation H.261 specifies the encodings used by ITU–T compliant video-conference codecs. H.261 was originally specified for fixed data rate ISDN circuits with multiple of 64kbit/s, but H.261 can also be used over packet–switched networks such as the Internet and wireless packet networks.

While the H263 Standard from 1996 is referred to as H.263, the updated version is called H.263+. One of the major improvements of H263+ is the introduction of bit stream scalability. Temporal, signal-to-noise ratio, and spatial scalability are used in H.263+. Numerous coding options were changed to improve coding performance. RFC 2429 considers the changes in the encoder.

RFC 2435 (replaces RFC 2035) gives the RTP payload format for JPEG-compressed encoded video streams. The Joint Photographic Experts Group (JPEG) standard defines still image compression. By combining a set of still images it is considered as motion JPEG. The packet format was optimized for real-time environments assuming that changes in the codec parameter are rare.

RFC 2250 (revision of RFC 2038) gives the packetization of MPEG1/MPEG2 audio and video for RTP. In the RFC two different approaches are presented. One approach is focusing on the compatibility with other RTP-encapsulated media streams, while a second approach is targeting maximum interoperability with MPEG system environments.

RFC 2431 specifies the RTP payload format for encapsulating ITU Recommendation BT.656–3 (digital television equipment operating on the 525-line or 625-line standards) video streams in RTP. RTP packets lengths depend on the number of scan lines. To rebuild the video frame each RTP packet contains information about fragmentation, decoding and positioning information.

RFC 3016 specifies the RTP payload format for MPEG4 based audio and video streams. This proposal for packet encapsulating allows the direct mapping of MPEG4 streams onto RTP packets.

6.6 Real Time Control Protocol (RTCP)

The companion control protocol for RTP is RTCP. It is introduced together with RTP in RFC 1889. Sender and receiver exchange RTCP packets to exchange QoS information periodically.

Five types of messages exist:

1. Sender Reports (SR)
2. Receiver Reports (RR)
3. Source Descriptions (SDES)
4. Application Specific Information (APP)
5. Session Termination Packets (BYE).

Each report type serves a different function. The SR report is sent by any host, which generated RTP packets. The SR includes the amount of data that was sent so far, as well as some timing information for synchronization process. Hosts that receive RTP streams generate the Receiver Report. This report includes information about the loss rate and the delay jitter of the RTP packets received so far. In addition the last timestamp and delay since the last SR was received, is included. This allows the sender to estimate the delay and jitter between sender and receiver. The rate of the RTCP packets is adjusted in dependency of the number of users per multicast group.

In general RTCP provides the following services: 1.) QoS monitoring and congestion control: This is the primary function of RTCP. RTCP provides feedback to an application about the quality of data distribution. The control information is useful to the senders, the receivers and third-party monitors. The sender can adjust its transmission based on the receiver report feedback. The receivers can determine whether congestion is local, regional or global. Network managers can evaluate the network performance for multicast distribution. 2.) Source identification: In RTP data packets, sources are identified by randomly generated 32-bit identifiers. These identifiers are not convenient for human users. RTCP SDES (source description) packets contain textual information called canonical names as globally unique identifiers of the session participants. It may include user's name, telephone number, email address and other information. 3.) Intermedia synchronization: RTCP sender reports contain an indication of real-time and the corresponding RTP timestamp. This can be used in inter-media synchronization like lip synchronization in video. 4.) Control information scaling: RTCP packets are sent periodically among participants. When the number of participants increases, it is necessary to balance between getting up-to-date control information and limiting the control traffic. In order to scale up to large multicast groups, RTCP has to prevent the control traffic from overwhelming network resources. RTP limits the control traffic to at most 5% of the overall session traffic. This is enforced by adjusting the RTCP generating rate according to the number of participants.

7 Video Streaming Mechanisms

The combination of the stringent Quality of Service (QoS) requirements of encoded video and the unreliability of wireless links make the real–time streaming over wireless links a very challenging problem. For uninterrupted video playback in a real-time streaming scenario the client has to decode and display a new video frame periodically (typically every 33 msec). This imposes tight timing constraints on the transmission of the video frames, once the playback at the wireless client has commenced. If a video frame is not completely delivered in time, the client loses a part or all of the frame. Generally, a small probability of frame loss (play back starvation) on the order of 10^{-5} – 10^{-2} is required for good perceived video quality. In addition, the video frame sizes (in byte) of the more efficient Variable Bit Rate (VBR) video encodings are highly variable, typically with peak–to–mean ratios of 4 – 10 as noted in Sec. 3, see also [25]. Wireless links, on the other hand, are typically highly error prone, as discussed in Sec. 4. They introduce a significant number of bit errors which may render a transmitted packet undecodable. The tight timing constraints of real–time video streaming, however, allow only for limited re–transmissions [49]. Moreover, the wireless link errors are typically time–varying and bursty. An error burst (which may persists for hundreds of milliseconds) may make the transmission to the affected client temporarily impossible. All of these properties and requirements make real–time streaming of video over wireless networks a very challenging problem. (We note that real–time video streaming is a significant challenge even for wired networks [50]). In this section we give an overview of the different strategies that can be employed to overcome the challenges of video streaming.

We organize our discussion according to the Internet protocol stack. First we discuss mechanisms that are employed at the channel coder level. We outline the basic concept of Forward Error Correction (FEC), a general technique to combat the bit errors on wireless links and then discuss some adaptive FEC techniques that have been specifically designed for wireless video streaming. Next, we move up to the link layer. We describe the basic ideas behind the concept of Automatic Repeat reQuest (ARQ) and then explore a number of ARQ schemes that have been tailored for video streaming. Strategies for video streaming often combine mechanisms that work at different levels of the protocol stack in *cross-layer designs*. We will discuss a few of the strategies that combine ARQ and FEC mechanisms; these strategies are often termed hybrid ARQ. Moving up to the network and transport layers, we will explore the issue of video streaming over the User Datagram Protocol (UDP) vs. streaming over the Transmission Control Protocol (TCP). Finally, we move up to the application layer, i.e., the video application and in particular the video (source) coder. We discuss techniques that adapt the video encoding in response to variations in the wireless channel. We will also describe scalable coding techniques which generate encoded video streams that can be adapted to the channel conditions at the lower protocol layers.

7.1 Forward Error Control Mechanisms

Shannon's channel coding theorem states that if the channel capacity is larger than the data rate, a coding scheme can be found to achieve small error probabilities. The basic idea behind **F**orward **E**rror **C**orrection (*FEC*) is to add redundancy to each information (payload) packet at the transmitter. At the receiver this redundancy is used to detect and/or correct errors within the information packet.

The binary (n, k) **B**ose–**C**haudhuri–**H**ocquenghem (*BCH*) code is a common FEC scheme based on block coding [51]. This code adds redundancy bits to payload bits to form code words and can correct a certain number of bit errors, see [51] for details. An important subclass of non–binary BCH codes are the **R**eed **S**olomon (*RS*) codes. An RS code groups the bits into symbols and thus achieves good burst error suppression capability.

An advantage of FEC is constant throughput with fixed (deterministic) delays independent of errors on the wireless channel. To achieve error–free (or close to error-free) communication, however, FEC schemes must be implemented for the worst case channel characteristics. This results in unnecessary overhead on the typically highly variable wireless links [52]. In particular, when the channel is currently good, the FEC dimensioned for the worst case conditions results in inefficient utilization of the wireless link. In addition, complex hardware structures may be required to implement the powerful, long codes required to combat the worst case error patterns. We also note that by adding redundancy bits the latency of all packets increases by a constant value. This additional delay may not be acceptable for streaming with very tight timing constraints.

In order to overcome the outlined drawbacks of FEC, *adaptive* FEC was introduced. Adaptive FEC adds redundancy as a function of the current wireless channel characteristics. Adaptive FEC for wireless communication has been studied extensively over the past five years or so. A number of adaptive FEC techniques have been specifically designed and evaluated for video streaming. The scheme [53], for instance, estimates the long term fading on the wireless channel and adapts the FEC to proactively protect the packets from loss.

Generally, adaptive FEC is an important component of an error control strategy and is often used in conjunction with the ARQ mechanisms discussed next, to form hybrid error control schemes. We note that adaptive FEC may increase the hardware complexity and possibly the signaling overhead. We note in closing this section on physical layer techniques for video streaming that some recent approaches adjust the transmit power level to ensure the successful transmission of video packets, see for instance [54, 55].

7.2 Automatic Repeat Request Mechanisms

The basic idea of **A**utomatic **R**epeat Re**Q**uest (*ARQ*) is to detect erroneous packets and to retransmit these packets until they are correctly received. Error are typically detected by applying an FEC code that has good error detection capabilities (and may have weak error correction performance). A wide variety of ARQ mechanisms have been studied over the years. The studied approaches differ in complexity and efficiency of the retransmission process. The simplest ARQ protocol is called *Send and Wait.* The basic idea behind this approach is to send a new packet only if the previous packet is transmitted successfully and has been acknowledged. In case an acknowledgment is missing, the packet is retransmitted until it is successfully received. Improved ARQ protocols are *Go–back–N* and *Selective Repeat.* ARQ achieves reasonable throughput levels if the error probability on the wireless link is not very large. More recently, ARQ techniques that employ the features of modern Code Division Multiple Access (CDMA) systems have been developed. The Simultaneous MAC Packet Transmission (SMPT) scheme [56], for instance, retransmits dropped packets on additional (parallel) CDMA codes. Roughly speaking, the scheme ramps up the number of used CDMA codes when losses occur. This avoids excessive delays and stabilizes the throughput at the link layer, thus reducing the probability of frame drop. We note that in a similar manner, the scheme [57] ramps up the transmit power.

7.3 Hybrid ARQ Techniques

The area of hybrid ARQ techniques that combine FEC and ARQ for more effective video streaming has been a very active area for the last couple of years. We discuss here a few of the major lines of work. One of the first studies to examine hybrid ARQ for wireless video is [49]. In the following many refined hybrid adaptive ARQ schemes for streaming low-bit rate video over wireless links have been developed, see for instance [58].

Some of the developed schemes are designed for particular wireless networks. Wireless LANs based on 802.11, for instance are considered in [59, 60]. Some schemes exploit the features of the encoded video for increasing the efficiency of the streaming. The scheme [61] exploits the fact that the different frame types of the MPEG encoded video have typically different sizes. The streams are coordinated such that the I frames of one video stream do not coincide with the I frames of another video stream. One fundamental aspect of this approach is the central coordination of the ongoing video streams. Central coordination generally allows for more efficient utilization of the wireless resources, but requires a central entity. In a cellular wireless system, for instance, the coordination of the streaming in the forward (downlink) direction in the base station can significantly improve efficiency. The scheme [62] exploits this central coordination. In addition the scheme exploits buffers in the wireless clients and the fact that streamed video is typically stored (prerecorded) for the prefetching of video data. During periods of good wireless channel conditions and low activity of the video traffic, the scheme transmits future video frames to the client buffer. The prefetched reserve of video frames allows the client to continue uninterrupted playback during periods of adverse wireless channel conditions or high-bit rare video segments. Along the same lines, [63] uses the receiver buffer to smooth an individual video stream (i.e., without exploiting central coordination).

7.4 Transport Layer: TCP or UDP?

Generally it is widely accepted that video streaming applications run over the User Datagram Protocol (UDP). UDP provides a connection-less and unreliable datagram service with a minimum of overhead. In addition, UDP allows streaming applications to transmit as fast as they desire. UDP is thus generally well suited for video streaming; a few refinements to improve its performance in conjunction with error resilient video coders have been developed, see for instance [64, 65].

The drawback of UDP is that it does not enforce any congestion control. Thus, the excessive use of UDP in the Internet can result in congestion collapse [66]. TCP provides a reliable connection between two terminals on top of an unreliable communication network, which may include wireless links. TCP enforces (*i*) congestion control to avoid excessive congestion in the network, and (*ii*) flow control to avoid overwhelming the receiver with data. TCPs congestion control employs an addittive-increase-multiplicative-decrease congestion window mechanisms which provides fair bandwidth allocation to all ongoing flows. One key limitation of employing TCP over wireless links is that by design TCP interprets lost packets as a sign of congestion, causing TCP to throttle the transmission rate. When wireless links are involved, however, packets are typically dropped due to errors on the wireless link and not due to congestion. Thus, cutting back the transmission rate may not be the best strategy and result in poor performance of TCP transmissions over wireless links. This issue has been studied in some detail in the general context of wireless data transmissions and a number of remedies have been explored, see for instance [67]. In the context of video streaming this issues has received relatively less interest so far, see for instance [68].

Another limitation of video streaming with TCP is that TCPs congestion window may limit the transmission rate to below the video bit rate. This is an issue both in streaming over wired networks and wireless networks and has also received only limited attention so far, see for instance [69].

A final consideration is that UDP supports multi-casting whereas TCP does not. As discussed in some detail in Sec. 7.6 multicast may be one way to efficiently utilize the scarce wireless bandwidth for video streaming.

7.5 Adaptation in the Video Coding

The adaptation schemes at the application layer, i.e., the level of the video (source) coder fall into two broad categories:

1. Adaptive encoding on-the-fly, and
2. Scalable off-line encoding.

With on-the-fly adaptive encoding the video coder changes the encoding parameters (such as quantization scale and thus produced bit rate or error resilience in the encoding) to adapt to the fluctuations on the wireless link. This technique is suitable for transmissions of live video, as well as for re-encoding or transcoding stored video for the transmission over the wireless medium. Scalable off-line encoding techniques, on the other hand, produces an encoded bit stream that can be scaled (adapted) to the current conditions on the wireless link, without requiring any on-the-fly encoding or re-encoding. Typically, scalable encoding techniques produce a base layer and one or several enhancement layers. The base layer provides basic video quality. Adding the enhancement layers (when the wireless link conditions allow for it) improves the video quality.

In this section we first discuss streaming mechanisms which incorporate on-the-fly adaptive encoding. We then outline the different scalable encoding techniques and discuss streaming mechanisms for scalable video.

7.5.1 Streaming Mechanisms with Adaptive Encoding

The approaches for adaptive encoding may be broadly classified into approaches that adjust the transmission bit rate to adapt to the wireless channel and approaches that adapt the error resilience of the encoding. The goal of both approaches is to minimize the distortion of the video delivered over the wireless link.

An example of the rate adaptation approach is the scheme developed in [70]. In this scheme, the acknowledgments for successfully received packets from the receiver are used in conjunction with a wireless channel model to predict the future wireless channel conditions. The predicted channel conditions and the playout deadlines of the video frames are translated into bit rate constraints for the video encoder. A variety of similar approaches has been explored, see for instance [71, 72].

An approach that adapts the error resilience for transmission over the wireless link is developed in [73]. This approach is tailored for transcoding, i.e., it takes an encoded video stream that is delivered over a wired network as input. Depedending on the current conditions on the wireless link, the transcoder injects error resilience information into the encoded bit stream. Both temporal resilience, which prevents bit errors from propagating to future frames, and spatial concealment, which limits the loss of synchronization when decoding the variable length codes (see also [74]) are added.

7.5.2 Streaming Mechanisms for Scalable Encoded Video

With scalable encoding the encoder produces typically multiple layers. The base layer provides a basic quality (e.g., low spatial or temporal resolution video) and adding enhancement layers improves the video quality (e.g., increases spatial resolution or frame rate). A variety of scalable encoding techniques have been developed, which we will introduce in this section. We will then discuss mechanisms for streaming scalable encoded video over wireless links. These approaches control the transmission of the layers so as to compensate for the variations of the wireless channel. Aside from adapting to the wireless channel, scalable encoding is a convenient way to adapt to the wide variety of video–capable hardware in wireless scenarios (e.g., PDAs, Laptops). Each of these devices has different constraints due to processing power, viewing size, and so on. Scalable encoding can satisfy these different constraints with one encoding of the video. We briefly note that an alternative to scalable encoding is to encode the video into different versions, each with a different quality level, bit rate, spatial/temporal resolution. The advantage of having different versions is that it does not require the more sophisticated scalable encoders and does not incur the extra overhead due to the scalable encoding. The drawback is that the multiple versions take up more space on servers and possibly need to be stream all together (simulcast) over the wired network to be able to choose the appropriate version at any given time over the wireless hop. Transcoding, as already noted above is another alternative to scalable encoding. Transcoding can be used to adapt to the wireless link conditions as in [73], or to adapt to different desired video formats [75]. This transcoding approach requires typically a high performance intermediate node.

Having given an overview of the general issues around scalable video encoding, we now introduce the different approaches to scalable encoding, we refer the interested reader to [76] for more details.

Data Partitioning Though not explicitly a scalable encoding technique, data partitioning divides the bitstream of non–scalable video standards such as MPEG–2 [7] into two parts. The base–layer contains critical data such as motion vectors and low–order DCT coefficients while the enhancement layer contains for example the higher order DCT coefficients [5]. The *priority break point* determines where to stop in the quantization and scanning process for the base–layer coefficients to be further encoded [10] as shown in

Figure 38. The remaining coefficients are then encoded by resuming the zig–zag scanning pattern at the break point and are stored in the enhancement layer.

Temporal Temporal scalability reduces the number of frames in the base layer. The removed frames are encoded into the enhancement layer and reference the frames of the base layer. Different patterns of combination of frames in base and enhancement layer exist [6]. In Figure 39 an enhancement layer consisting of all B–frames is given as already used in video trace evaluations [34]. No other frames depend on the successful decoding of B–frames. If the enhancement layer is not decodable, the decoding of the other frame types is not affected. Nevertheless, since the number of frames that are reconstructed changes, the rate of frames per second is to be adjusted accordingly for viewing and quality evaluation methods (e.g., the last successfully decoded frame is displayed for a longer period, also called *freezing*). This adjustment is inflicting the loss in viewable video quality.

Spatial Scalability Scalability in the spatial domain is applying different resolutions to the base and the enhancement layer. If, for example, the original sequence is in the CIF format (352 × 288), the base layer is downsampled into the QCIF format (176 × 144) prior to the encoding. Spatial scalability is therefore also known as *pyramid coding*. In addition to the application of different resolutions, different GoP structures are used in the two layers. The GoP pattern in the enhancement layer is referencing the frames in the base layer. An exemplary layout of the resulting dependencies is illustrated in Figure 40. The content of the enhancement layer is the difference between the layers, as well as the frame–based reference of previous and following frames of the same layer. A study of the traffic and quality characteristics of temporal and spatial scalable encoded video is given in [34].

SNR Scalability SNR scalability provides two (or more) different video layers of the same resolution but with different qualities. The base layer is coded by itself and provides a basic quality in terms of the (P)SNR (see Subsection 3.1). The enhancement layer is encoded to provide additional quality when added back to the base layer. The encoding is performed in two consecutive steps: first the base layer is encoded with a low quality, then the difference between the decoded base layer and the input video is encoded with higher quality settings in a second step [76] as illustrated in Figure 41. At the receiver–side the base quality is obtained simply by decoding the base layer. For enhanced quality, the enhanced layer is decoded and the result is added to the base layer. There is no explicit need for both layers to be encoded by the same video compression standard, though for ease of use it is advisable to do so.

Object Scalability Another scalability feature is possible within video standards that support the composition of video frames out of several different objects, such as MPEG–4 [6]. The base layer contains only the information that could not be fitted or identified as video objects. The enhancement layer(s) are made up of the respective information for the video objects, such as shape and texture. The example shown in Figure 42 presents a case, where the background (landscape) and an object (car) were separated. In this case, the background is encoded independently from the object.

Fine Grain Scalability Fine Grain Scalability (FGS) is a relatively new form of scalable video encoding [77]. With FGS the video is encoded into a base layer and one enhancement layer. This enhancement layer has the special property that it can be cut at any bit rate and all the bits that are transmitted contribute towards improving the decoded video quality. FGS thus removes the restriction of conventional layered encoding where an enhancement layer must be completely received for successful decoding. The flexibility of FGS makes it attractive for video streaming, but this flexibility comes at the expense of reduced coding efficiency. Efforts are currently under way to improve the coding efficiency while maintaining the FGS flexibility.

Multiple Descriptions Multiple Description (MD) coding is yet another form of scalable coding of images and videos. With MD coding, an image or video is coded into multiple descriptions (which can be thought of as layers or streams). These descriptions have no explicit hierarchy, i.e., there is no notion of a base description and enhancement descriptions. Instead, any of the descriptions can be combined and decoded [78, 79, 80].

The more of the encoded descriptions are available at the decoder, the higher the decoded image or video quality.

Having given a brief introduction to scalable video coding we now discuss streaming mechanisms for scalable video in wireless networks. General frameworks for streaming scalable video in wireless networks have been proposed in [81, 82, 83] A number of schemes for efficiently streaming the layers of scalable encoded video have been developed, see for instance [84]. These approaches focus on the packet scheduling over the wireless links. A number of other approaches incorporate multiple priorities for transmission (e.g., [85]) or hybrid ARQ/FEC and power control (e.g., [86]) for the efficient transmission of scalable video over wireless links. The emergence of the FGS coding technique has also prompted studies of the traffic characteristics of FGS encoded video [87] and transport schemes for wireless networks [88, 89].

Yet another direction of research develops specialized scalable encoding schemes for wireless video streaming, see for instance [90, 91, 92, 93]

In this context, a number of studies explore the joint optimization of the video (source) coding and the (channel) coding for the transmission over the wireless links, see for instance [94, 95, 96].

7.6 Wireless Multicast of Video Streams

Multicast is gaining a lot of interest in the wireless sector, because it allows introducing video streaming services in a multicast fashion at a favorable price. Considering the scare bandwidth and the price paid for the frequency bands for 3G networks the introduction of video services using unicast connection would be very hard. So it is likely that the first video services in 3G networks will be based on multicast.

UMTS networks distinguish between two kinds of channels: dedicated channels and common channels. The dedicated channel is a point to point channel with power control. Using the dedicated channel each wireless terminal has a unique channel. The common channel is a point to multi–point channel without power control. Multiple wireless terminals may connect to the common channel. Therefore, multicast can be applied in two different fashions such as a mesh of unicast connection over dedicated channels or using the common channel. It depends on the distance of the user and the base station, which solution is chosen to achieve high bandwidth efficiency. Generally for a large number of wireless terminals the common channel seems to be preferable. On top of these channels the IP protocol stack shown in Figure 34 is used.

8 Conclusion

In this chapter we have given an overview of wireless video streaming. We have given an intuitive introduction to the basic principles of video encoding. We have reviewed the properties of the wireless channel and how these properties affect video transmissions. We gave an overview of the Internet protocols involved in wireless video streaming. Finally, we have discussed the currently available strategies for video streaming over wireless links. We note in closing that this is an active area of research with many fundamental advances to be expected over the next few years. One trend in this ongoing research is to incorporate the application's or user's perspective more into the streaming [97, 98].

9 Acronyms

3G	Third generation mobile communication systems
AC	Alternating current
ARQ	Automatic repeat request
BCH	Bose–Chaudhuri–Hocquenghem
BER	Bit error rate
BMA	lock matching algorithm
CIF	Picture and video format, dimensions are 352 columns and 288 rows
CW	Congestion window
DC	Direct current
DCT	Discrete cosine transform
DLL	Data link layer
EOB	End of block
FEC	Forward error correction
GoP	Group of Pictures
GSM	Global system for mobile communication
H.26x	Video standards of the ITU: H.261, H.263, H.263+, H.26L
HDTV	High definition television
HTTP	Hypertext transfer protocol
IETF	Internet engineering task force
IP	Internet protocol
ISDN	Integrated services digital network
ISI	Intersymbol interference
ISO	International Standardization Organisation
ITU	International Telecommunication Union
JPEG	Joint photographic experts group
JVT	Joint Video Team
LoS	Line of sight
LPDU	Link layer packet data unit

MPEG	Moving Picture Experts Group
MSE	Mean squared error
NLos	Non–Line of sight
PDA	Personal data assistant
PSNR	Peak signal to noise ratio
QCIF	Picture and video format, dimensions are 256 columns and 144 rows
QoS	Quality of service
RFC	Request for comment
RGB	Color space format consisting of the values of red, green, and blue
RS	Reed–Solomon
RSVP	Resource reservation protocol
RTO	Retransmission time out
RTP	Real time protocol
RTCP	Real time control protocol
RTSP	Real time streaming protocol
RTT	Round trip time
SAP	Session announcement protocol
SDES	Source description
SDP	Session description protocol
SIP	Session initiation protocol
SNR	Signal to noise ratio
TCP	Transmission control protocol
TV	Television
UDP	User datagram protocol
UMTS	Universal Mobile Telecommunications System
VLC	Variable length coding
YUV	Color space format consisting of the values for luminance, chrominance, and saturation

Acknowledgment

This work was supported in part by the National Science Foundation through grants Career ANI-0133252 and ANI-0136774.

References

[1] J. Gross, F. Fitzek, A. Wolisz, B. Chen, and R. Gruenheid, "Framework for combined optimization of dl and physical layer in mobile ofdm systems," in *Proc. of 6th Int. OFDM-Workshop 2001*, Hamburg, Germany, Sept. 2001, pp. 32–1–32–5.

[2] V. Kravcenko, H. Boche, F. Fitzek, and A. Wolisz, "No need for signaling: Investigation of capacity and quality of service for multi–code cdma systems using the WBE++ approach," in *Proceedings of the Fourth IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002)*, Stockholm, Sweden, Sept. 2002, pp. 110–114.

[3] Commission of the European Communities, "Digital Land Mobile Radio Communications - COST 207," Office for Official Publications of the European Communities, Luxembourg, Tech. Rep., 1989, final Report.

[4] "Video traces for network performance evaluation," Online Website. [Online]. Available: http://www.eas.asu.edu/trace

[5] T. Sikora, "MPEG Digital Video Coding Standards," in *Digital Electronics Consumer Handbook.* McGraw Hill, 1997.

[6] F. Pereiera and E. Touradj, *The MPEG–4 Book.* Prentice Hall, 2002.

[7] MPEG–2, "Generic coding of moving pictures and associated ausio information," ISO/IEC 13818-2, 1994, draft International Standard.

[8] MPEG–1, "Coding of moving pictures and associated audio for digital storage media at up to 1.5 mbps," ISO/IEC 11172, 1993.

[9] M. Riley and I. Richardson, *Digital Video Communications.* Artech House, 1997.

[10] M. Ghanbari, *Video Coding – an introduction to standard codecs.* The Institution of Electrical Engineers, 1999.

[11] A.N. Netravali and B.G. Haskell, *Digital Pictures: Representation, Compression, and Standards.* Plenum Press, 1995, ch. 3.

[12] Z. Xiong, K. Ramchandran, M.T. Orchard, and Y.–Q. Zhang, “A Comparative Study of DCT– and Wavelet–Based Image Coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 5, August 1999.

[13] W. Chen, C. Smith, and S. Fralick, “A fast computational algorithm for the discrete cosine transform,” *IEEE Transcript on Communications*, pp. 1004–1009, Sept. 1977.

[14] I. Dinstein, K. Rose, and A. Heimann, “Variable block–size transform image coder,” *IEEE Trans. on Communications*, pp. 2073–2078, Nov. 1990.

[15] D. Turaga and T. Chen, “Fundamentals of video compression: H.263 as an example,” in *Compressed Video over Networks*, A. R. Reibman and M.-T. Sun, Eds. Marcel Dekker, 2001, pp. 3–34.

[16] T. Lakshman, A. Ortega, and A. Reibman, “VBR video: Tradeoffs and potentials,” *Proceedings of the IEEE*, vol. 86, no. 5, pp. 952–973, May 1998.

[17] A. Ortega, “Variable bit rate video coding,” in *Compressed Video Over Networks*, M.-T. Sun and A. Reibman, Eds., November 2001, pp. 343–383.

[18] Y. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering.* CRC Press, 2000.

[19] T. Koga et. al., “Motion compensated interframe coding for video conferencing.” New Orleans, LA: National Telecommunication Conference, 1981, pp. G5.3.1–G5.3.5.

[20] M. Bierling, “Displacement estimation by hierarchical block matching,” *Visual Communications and Image Processing*, vol. 1001, pp. 942–951, 1988.

[21] D. Lelewer and D. Hirschberg, “Data compression,” *ACM Computing Surveys (CSUR)*, vol. 19, no. 3, pp. 261–296, 1987.

[22] P. Howard and J. Vitter, “Analysis of arithmetic coding for data compression,” *Information Processing and Management*, vol. 28, no. 6, pp. 749–764, 1992.

[23] T. Wiegand, “H.26L Test Model Long–Term Number 9 (TML-9) draft0,” ITU-T Study Group 16, Dec. 2001.

[24] F. Fitzek, P. Seeling, and M. Reisslein, “H.26l pre–standard evaluation,” acticom GmbH, Tech. Rep., Nov. 2002. [Online]. Available: http://www.acticom.de/

[25] F. Fitzek and M. Reisslein, “MPEG–4 and H.263 video traces for network performance evaluation,” *IEEE Network*, vol. 15, no. 6, pp. 40–54, November/December 2001.

[26] M. Krunz, R. Sass, and H. Hughes, “Statistical characteristics and multiplexing of MPEG streams,” in *Proceedings of IEEE Infocom ’95*, April 1995, pp. 455–462.

[27] O. Rose, “Statistical properties of MPEG video traffic and their impact on traffic modelling in ATM systems,” University of Wuerzburg, Institute of Computer Science, Tech. Rep. 101, Feb. 1995.

[28] W.-C. Feng, *Buffering Techniques for Delivery of Compressed Video in Video–on–Demand Systems*. Kluwer Academic Publisher, 1997.

[29] “The network simulator — ns–2.” [Online]. Available: www.isi.edu/nsnam/ns/index.html

[30] A. Varga, “Omnet++,” *IEEE Network Interactive*, vol. 16, no. 4, 2002. [Online]. Available: www.hit.bme.hu/phd/vargaa/omnetpp

[31] J. Buck, E. L. S. Ha, and D. Messerschmitt, “Ptolemy: A Framework for Simulating and Prototyping Heterogeneous Systems,” *Int. Journal of Computer Simulation*, vol. 4, pp. 155–182, Apr. 1994. [Online]. Available: ptolemy.eecs.berkeley.edu/index.htm

[32] F. Fitzek, P. Seeling, and M. Reisslein, “Using network simulators with video traces,” Arizona State University, Dept. of Electrical Eng, Tech. Rep., mar 2003. [Online]. Available: http://www.eas.asu.edu/trace

[33] A. Marie and et. al., "Video quality experts group: Current results and future directions," no. 4067, Perth, Australia, 2000, pp. 742–453.

[34] M. Reisslein, J. Lassetter, S. Ratnam, O. Lotfallah, F. Fitzek, and S. Panchanathan, "Traffic and quality characterization of scalable encoded video: A large-scale trace-based study," Tech. Rep.

[35] T. Rappaport, S. Seidel, and K. Takamizawa, "Statistical Channel Impulse Response Models for Factory and Open Plan Building Radio Communication System Design," *IEEE Transactions on Communications*, vol. 39, no. 5, pp. 794–807, May 1991.

[36] J. D. Gibson, *Mobile Communications - Handbook.* IEEE Press, 1996, vol. 2.

[37] M. Aldinger, "Die simulation des mobilfunk–kanals auf einem digitalrechner," in *FREQUENZ*, vol. 36, no. 4/5, 1982, pp. 145–152.

[38] L. Kittel, "Analoge und diskrete Kanalmodelle fr die Signalbertragung beim beweglichen Funk," in *FREQUENZ*, vol. 36, no. 4/5, 1982, pp. 152–160.

[39] M. Zorzi and R. Rao, "Error–Constrained Error Control for Wireless Channels," *IEEE Personal Communications*, pp. 27–33, dec 1997.

[40] R. R. Rao, "Higher layer perspectives on modeling the wireless channel," in *Proceedings of IEEE ITW*, Killarney, Ireland, June 1998, pp. 137–138.

[41] M. Zorzi, R. R. Rao, and L. B. Milstein, "On the accuracy of a first–order markovian model for data block transmission on fading channels," in *Proceedings of IEEE International Conference on Universal Personal Communications*, Nov. 1995, pp. 211–215.

[42] A. Elwalid, D. Heyman, T. Lakshman, D. Mitra, and A. Weiss, "Fundamental bounds and approximations for ATM multiplexers with application to video teleconferencing," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1004–1016, Aug. 1995.

[43] M. Handley, C. Perkins, and E. Whelan, “RFC 2974: SAP: Session announcement protocol,” Oct. 2000.

[44] J. Rosenberg and et. al., “RFC 3261: SIP: Session initiation protocol,” Feb. 1999.

[45] H. Schulzrinne, A. Rao, and R. Lanphier, “RFC 2326: Real time streaming protocol (RTSP),” Apr. 1998.

[46] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RFC 1889: RTP: A transport protocol for real-time applications,” Jan. 1996.

[47] C. Zhu, “RFC 2190: RTP payload format for H.263 video streams,” Sept. 1997.

[48] M. Speer and D. Hoffman, “RFC 2029: RTP payload format of Sun’s CellB video encoding,” Oct. 1996.

[49] H. Liu and M. E. Zarki, “Performance of H.263 video transmission over wireless networks using hybrid ARQ,” *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 9, pp. 1775–1786, Dec. 1997.

[50] G. Karlsson, “Asynchronous transfer of video,” *IEEE Communications Magazine*, vol. 34, no. 8, pp. 106–113, Feb. 1996.

[51] S. Lin, *Error Control Coding: Fundamentals and Applications.* Prentice–Hall, 1983.

[52] H. Liu, H.Ma, M. E. Zarki, and S. Gupta, “Error control schemes for networks: An overview,” *Mobile Networks and Applications*, no. 2, pp. 167–182, 1997.

[53] W. Kumwilaisak, J. Kim, and C. Kuo, “Reliable wireless video transmission via fading channel estimation and adaptation,” in *Proceedings of IEEE WCNC*, Chicago, IL, Sept. 2000, pp. 185–190.

[54] I.-M. Kim and H.-M. Kim, “Efficient power management schemes for video service in CDMA systems,” *IEEE Electronics Letters*, vol. 36, no. 13, pp. 1149–1150, June 2000.

[55] ——, “An optimum power management scheme for wireless video service in CDMA systems,” *IEEE Transactions on Wireless Communications*, vol. 2, no. 1, pp. 81–91, jan 2003.

[56] F. H. P. Fitzek, M. Reisslein, and A. Wolisz, “Uncoordinated real-time Video Transmission in Wireless Multicode CDMA Systems: An SMPT–Based Approach,” *IEEE Wireless Communications*, vol. 9, no. 5, pp. 100–110, Oct. 2002.

[57] S.-H. Hwang, B. Kim, and Y.-S. Kim, “A hybrid ARQ scheme with power ramping,” in *Proceedings of the 54th IEEE Vehicular Technology Conference*, vol. 3, 2001, pp. 1579–1583.

[58] D. Qiao and K. G. Shin, “A two–step adaptive error recovery scheme for video transmission over wireless networks,” in *Proceedings of IEEE Infocom*, Tel Aviv, Israel, Mar. 2000.

[59] A. Majumdar, D. G. Sachs, I. V. Kozintsev, K. Ramchandran, and M. Yeung, “Multicast and unicast real-time video streaming over wireless LANs,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 524–534, June 2002.

[60] R. Manghavamanid, M. Demirhan, R. Raikumar, and D. Raychaudhuri, “Size–based scheduling for delay–sensitive variable bit rate traffic over wireless channels,” Real–time and Multimedia Systems Lab. Carnegie Mellon Univ., Tech. Rep., Jan. 2003.

[61] P.-R. Chang and C.-F. Lin, “Wireless ATM–based multicode CDMA transport architecture for MPEG–2 video transmission,” *Proceedings of the IEEE*, vol. 87, no. 10, pp. 1807–1824, Oct. 1999.

[62] F. Fitzek and M. Reisslein, “A prefetching protocol for continuous media streaming in wireless environments,” *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2015–2028, Oct. 2001.

[63] C. Iskander and P. T. Mathiopoulos, “Rate–adaptive transmission of H.263 video for multicode DS/CDMA cellular systems in multipath fading,” in *Proc. of IEEE Vehicular Technology Conference*, Spring 2002, pp. 473–477.

[64] A. Singh, A. Konrad, and A. D. Joseph, “Performance evaluation of UDP Lite for cellular video,” in *Proceedings of The 11th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Port Jefferson, NY, June 2001.

[65] H. Zheng and J. Boyce, “An improved UDP protocol for video transmission over internet–to–wireless networks,” *IEEE Transactions on Multimedia*, vol. 3, no. 3, pp. 356–365, Sept. 2001.

[66] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the internet,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, Aug. 1999.

[67] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, “A Comparison of Mechanisms for Improving TCP Performance over Wireless Links,” *IEEE ACM Transactions on Networking*, December 1997.

[68] F. H. P. Fitzek, R. Supatrio, A. Wolisz, M. Krishnam, and M. Reisslein, “Streaming video applications over TCP in CDMA based networks,” in *Proceedings of the 2002 International Conference on Third Generation Wireless and Beyond (3Gwireless 2002)*, San Francisco, CA, May 2002, pp. 755–760.

[69] C. Krasic, K. Li, and J. Walpole, “The case for streaming multimedia with TCP,” in *Proceedings of 8th International Workshop on Interactive Distributed Multimedia Systems (IDMS)*, Lancaster, UK, Sept. 2001.

[70] C. Hsu, A. Ortega, and M. Khansari, “Rate control for robust video transmission over burst–error wireless channels,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, pp. 756–773, May 1999.

[71] P.-C. Hu, Z.-L. Zhang, and M. Kaveh, “Channel condition ARQ rate control for real-time wireless video under buffer constraints,” in *Proceedings of IEEE International Conference on Image Processing*, 2000, pp. 124–127.

[72] A. Tosun and W. Feng, “On improving quality of video for H.263 over wireless CDMA networks,” in *Proceedings of IEEE WCNC*, Chicago, IL, Sept. 2000, pp. 1421–1426.

[73] G. Reyes, A. R. Reibman, S.-F. Chang, and J. Chuang, “Error–resilient transcoding for video over wireless channels,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1063–1074, June 2000.

[74] G. Cote, F. Kossentini, and S. Wenger, “Error resilience coding,” in *Compressed Video over Networks*, M.-T. Sun and A. R. Reibman, Eds. Marcel Dekker, 2001, pp. 309–341.

[75] T. Shanableh and M. Ghanbari, “Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats,” *IEEE Transactions on Multimedia*, vol. 2, no. 2, pp. 101–110, June 2000.

[76] M. Ghanbari, “Layered coding,” in *Compressed Video over Networks*, A. R. Reibman and M.-T. Sun, Eds. Marcel Dekker, 2001, pp. 251–308.

[77] W. Li, “Overview of Fine Granularity Scalability in MPEG–4 Video Standard,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301–317, March 2001.

[78] V. Goyal, “Multiple description coding: Compression meets the network,” *IEEE Signal Processing Magazine*, vol. 18, pp. 74–91, sept 2001.

[79] L. Gao, L. Karam, M. Reisslein, and G. Abousleman, “Error–resilient image coding and transmission over wireless channels,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, Scottsdale, AZ, may 2002, pp. 629–632.

[80] S. Ekmekci and T. Sikora, "Unbalanced quantized multiple description video transmission using path diversity," in *IS&T/SPIE's Electronic Imaging 2003*, 2003, santa Clara, CA.

[81] M. Naghshineh and M. LeMair, "End–to–end QoS provisioning in multimedia wireless/mobile networks using an adaptive framework," *IEEE Communications Magazine*, vol. 35, no. 11, pp. 72–81, Nov. 1997.

[82] G. Fankhauser, M. Dasen, N. Weiler, B. Plattner, and B. Stiller, "The WaveVideo system and network architecture: Design and implementation," Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Tech. Rep., June 1998.

[83] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Scalable video coding and transport over broad–band wireless networks," *Proceedings of the IEEE*, vol. 89, no. 1, pp. 6–20, Jan. 2001.

[84] Z. Jiang and L. Kleinrock, "A packet selection algorithm for adaptive transmission of smoothed video over a wireless channel," *IEEE Transactions on Parallel and Distributed Systems*, vol. 60, no. 4, pp. 494–509, Apr. 2000.

[85] H. Gharavi and S. M. Alamouti, "Multipriority video transmission for third–generation wireless communication systems," *Proceedings of the IEEE*, vol. 87, no. 10, pp. 1751–1763, Oct. 1999.

[86] S. Zhao, Z. Xiong, and X. Wang, "Joint error control and power allocation for video transmission over CDMA networks with multiuser detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 425–437, June 2002.

[87] P. de Cuetos, M. Reisslein, and K. W. Ross, "Analysis of a large library of MPEG–4 FGS rate–distortion traces for streaming video," Institut Eurecom, Tech. Rep., Dec. 2002, traces available at `http://www.eas.asu.edu/trace`.

[88] M. van der Schaar and H. Radha, "Adaptive motion-compensation fine–granular–

scalability (AMC-FGS) for wireless video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 360–371, June 2002.

[89] T. Stockhammer, H. Jenkac, and C. Weiss, "Feedback and error protection strategies for wireless progressive video transmission," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 465–482, June 2002.

[90] U. Horn, B. Girod, and B. Belzer, "Scalable video coding for multimedia applications and robust transmission over wireless channels," in *7th International Workshop on Packet Video*, Mar. 1996. [Online]. Available: citeseer.nj.nec.com/horn96scalable.html

[91] F. Steinbach, N. Färber, and B. Girod, "Standard compatible extension of h.263 for robust video transmission in mobile environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 872–881, dec 1997.

[92] Y. Yu and C. Chen, "SNR scalable transcoding for video over wireless channels," in *Proceedings of IEEE WCNC*, Chicago, IL, Sept. 2000, pp. 1398–1402.

[93] L. Kondi, S. Batalama, D. Pados, and A. Katsaggelos, "Joint source-channel coding for scalable video over DS-CDMA multipath fading channels," in *Proceedings of the 2001 International Conference on Image Processing*, 2001, pp. 994–997.

[94] M. Srinivasan and R. Chellappa, "Adaptive source–channel subband video coding for wireless channels," *IEEE Journal for Selected Areas in Communication*, vol. 16, pp. 1830–1839, dec 1998.

[95] P.-R. Chang, "Spread spectrum CDMA systems for subband image transmission," *IEEE Transactions on Vehicular Technology*, vol. 46, pp. 80–95, feb 1997.

[96] E. Iun and H.-M. Khandani, "Combined source–channel coding forthe transmission of still images over a code division multiple access (cdma) channel," *IEEE Communic. Letters*, vol. 2, pp. 168–170, jun 1998.

[97] X. Lu, R. O. Morando, and M. ElZarki, "Understanding video quality and its use in feedback control," in *Proceedings of International Packet Video Workshop 2002*, Pittsburgh, PA, 2002.

[98] X. Meng, H. Yang, and S. Lu, "Application–oriented multimedia scheduling over lossy wireless links," in *Proceedings of the Eleventh International Conference on Computer Communications and Networks*, Miami, FL, Oct. 2002, pp. 256–261.

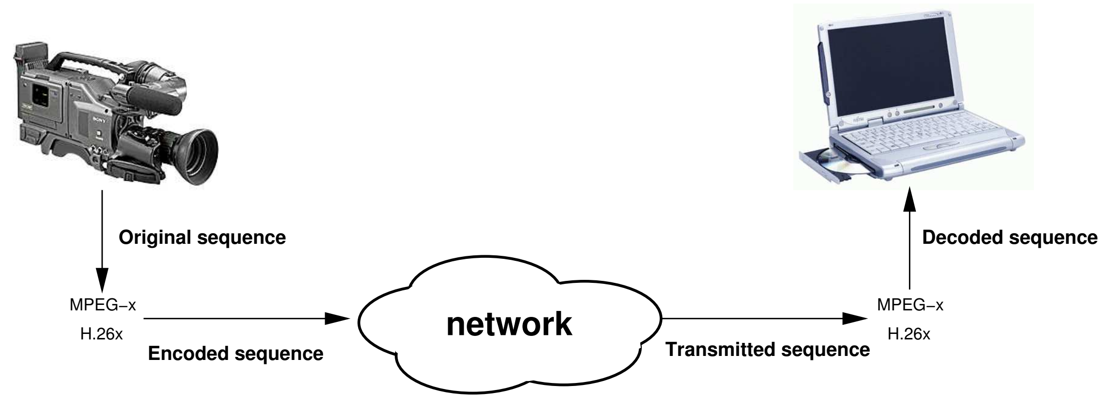


Figure 1: Generic wireless video streaming system.

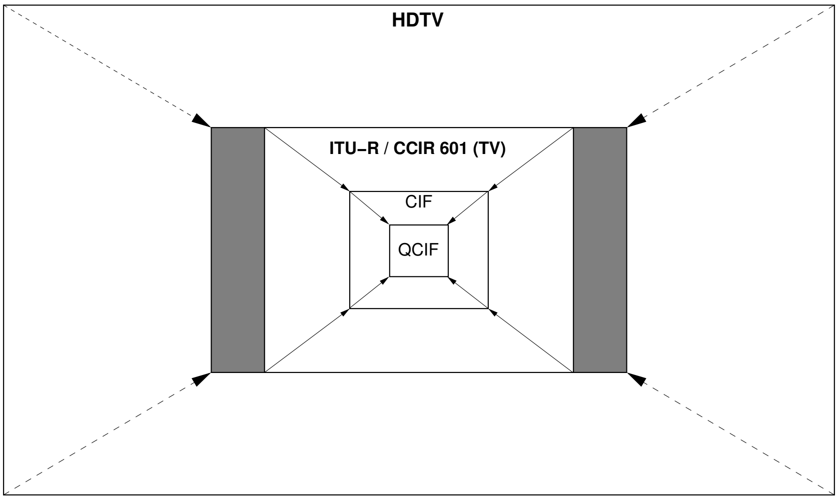


Figure 2: Illustration of image formats.

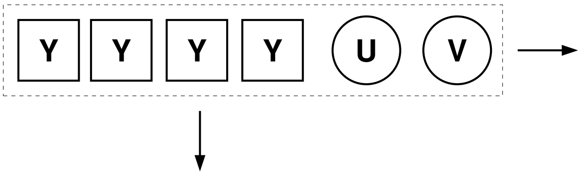


Figure 3: YUV 4:1:1 subsampling

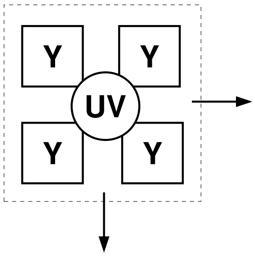


Figure 4: YUV 4:2:0 subsampling

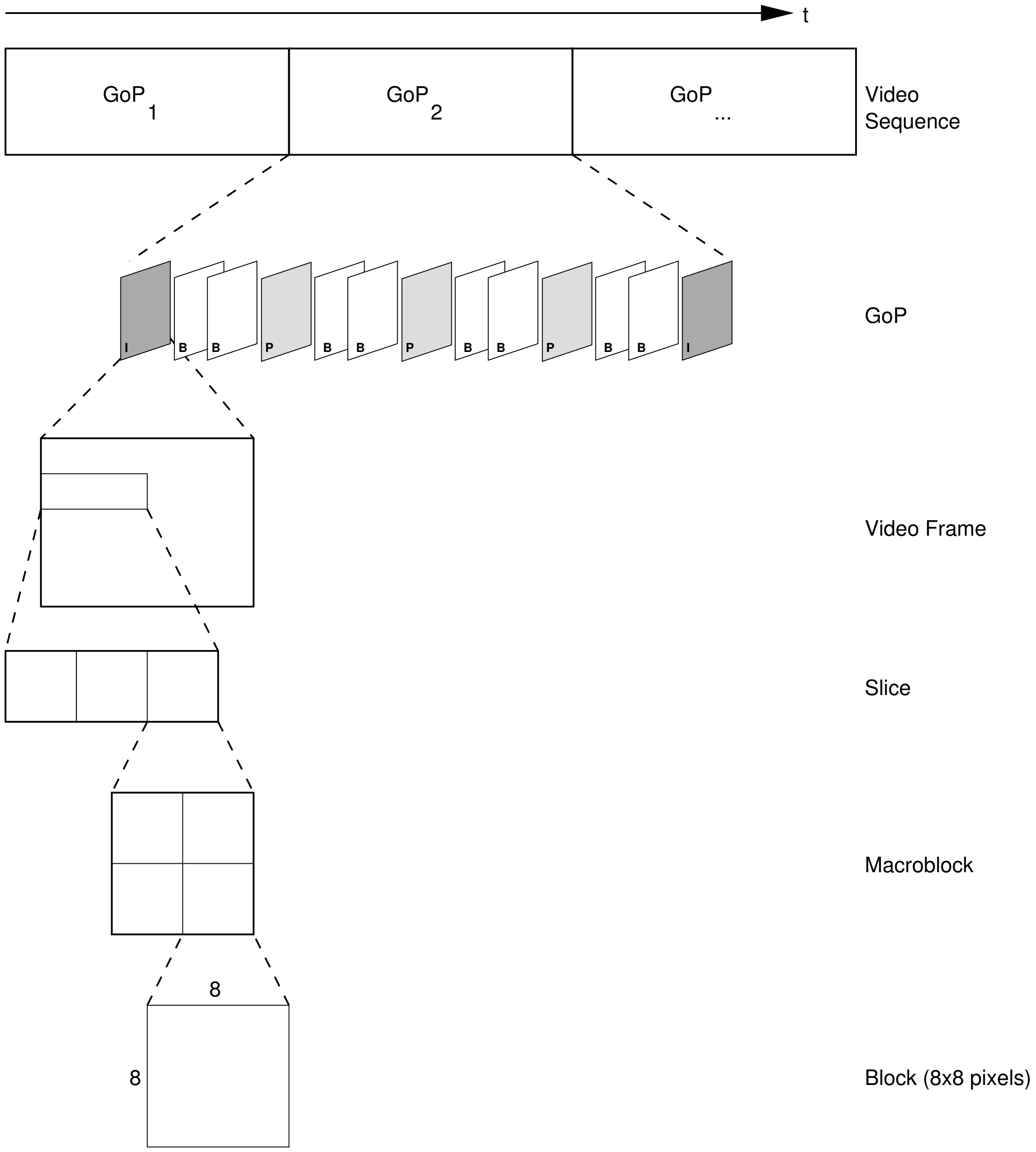


Figure 5: Typical composition of a video sequence.

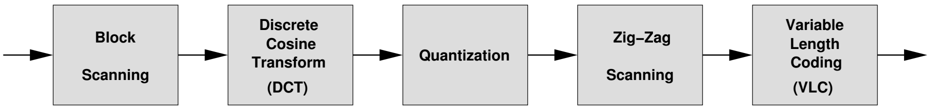


Figure 6: DCT coding concept.

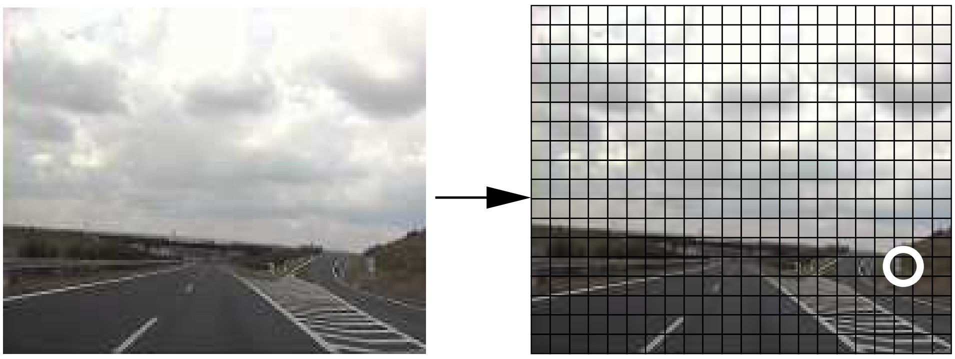
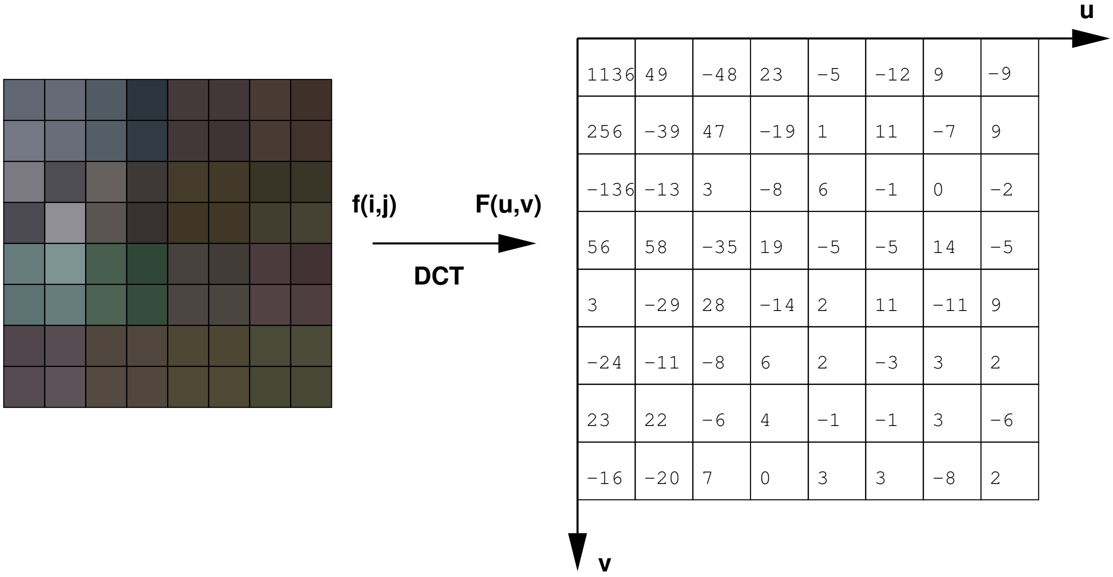


Figure 7: Video frame subdivision into blocks (QCIF format into 22×18 blocks of 8×8 pixels each).



1136	49	-48	23	-5	-12	9	-9
256	-39	47	-19	1	11	-7	9
-136	-13	3	-8	6	-1	0	-2
56	58	-35	19	-5	-5	14	-5
3	-29	28	-14	2	11	-11	9
-24	-11	-8	6	2	-3	3	2
23	22	-6	4	-1	-1	3	-6
-16	-20	7	0	3	3	-8	2

Figure 8: 8 × 8 block of luminance values (visual representation and numerical values) and the resulting DCT transform coefficients (decimal places truncated).

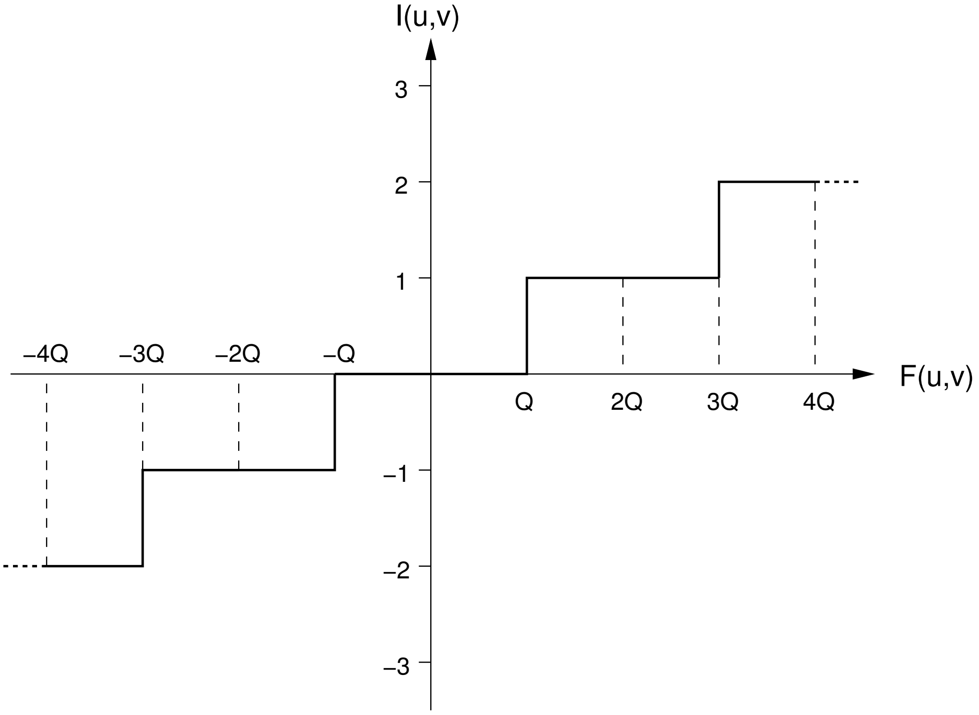


Figure 9: Illustration of quantization, $T = Q$.

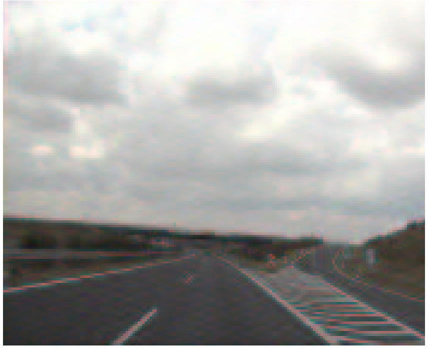


Figure 10: Small quantization parameter ($Q = 16, PSNR = 44.9579$ dB, frame size= 4640 byte).

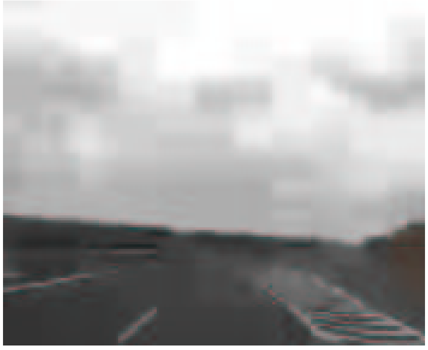


Figure 11: Medium quantization parameter ($Q = 35, PSNR = 33.0344$ dB, frame size= 135 byte).

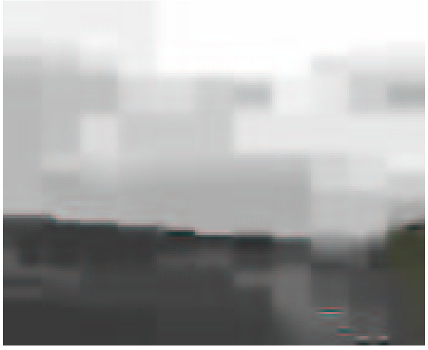
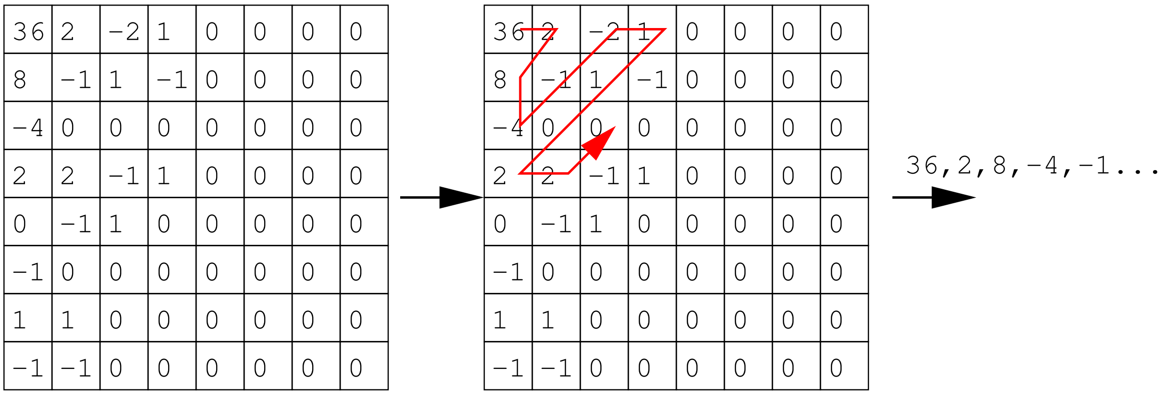


Figure 12: Large quantization parameter ($Q = 40, PSNR = 30.2765$ dB, frame size= 79 byte).



36	2	−2	1	0	0	0	0
8	−1	1	−1	0	0	0	0
−4	0	0	0	0	0	0	0
2	2	−1	1	0	0	0	0
0	−1	1	0	0	0	0	0
−1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
−1	−1	0	0	0	0	0	0

36	2	−2	1	0	0	0	0
8	−1	1	−1	0	0	0	0
−4	0	0	0	0	0	0	0
2	2	−1	1	0	0	0	0
0	−1	1	0	0	0	0	0
−1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
−1	−1	0	0	0	0	0	0

36,2,8,−4,−1...

Figure 13: Quantized DCT coefficients ($Q = 16$) and zig–zag scanning pattern.

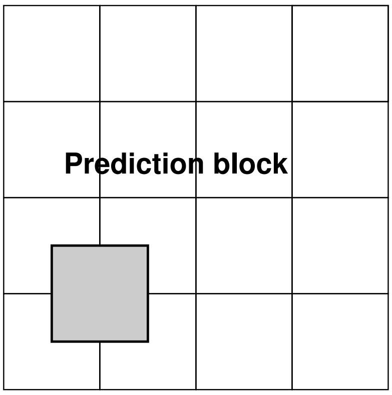


Figure 14: Reference (past) frame.

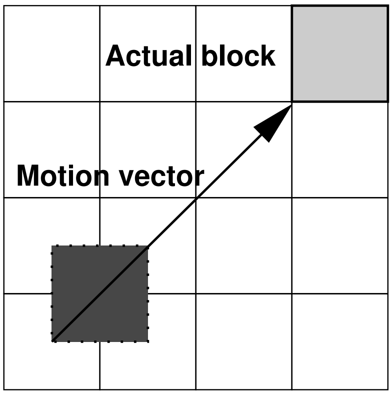


Figure 15: Predicted (current) frame.

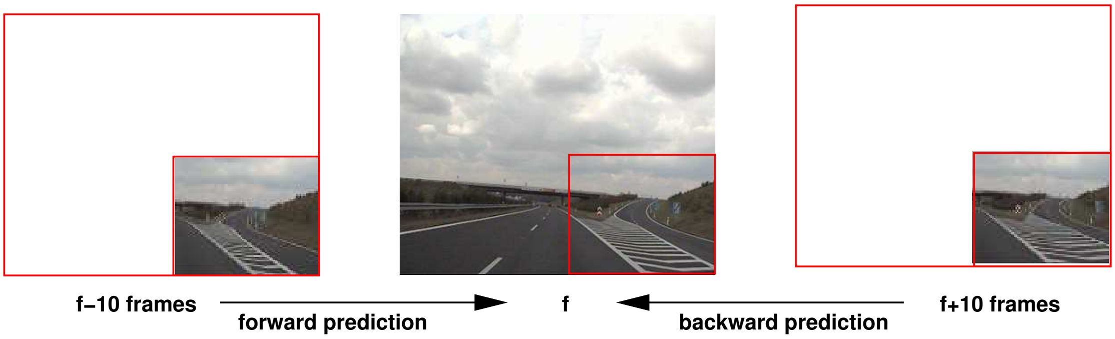


Figure 16: Changing video frame content.

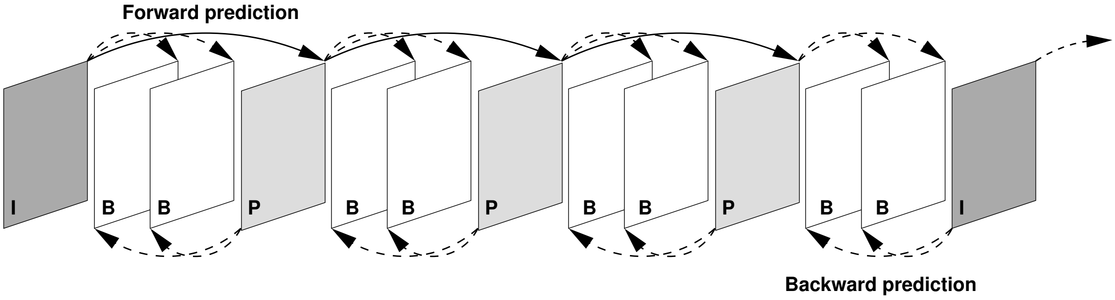


Figure 17: Typical MPEG Group of Pictures (GoP) (frames 1–12).

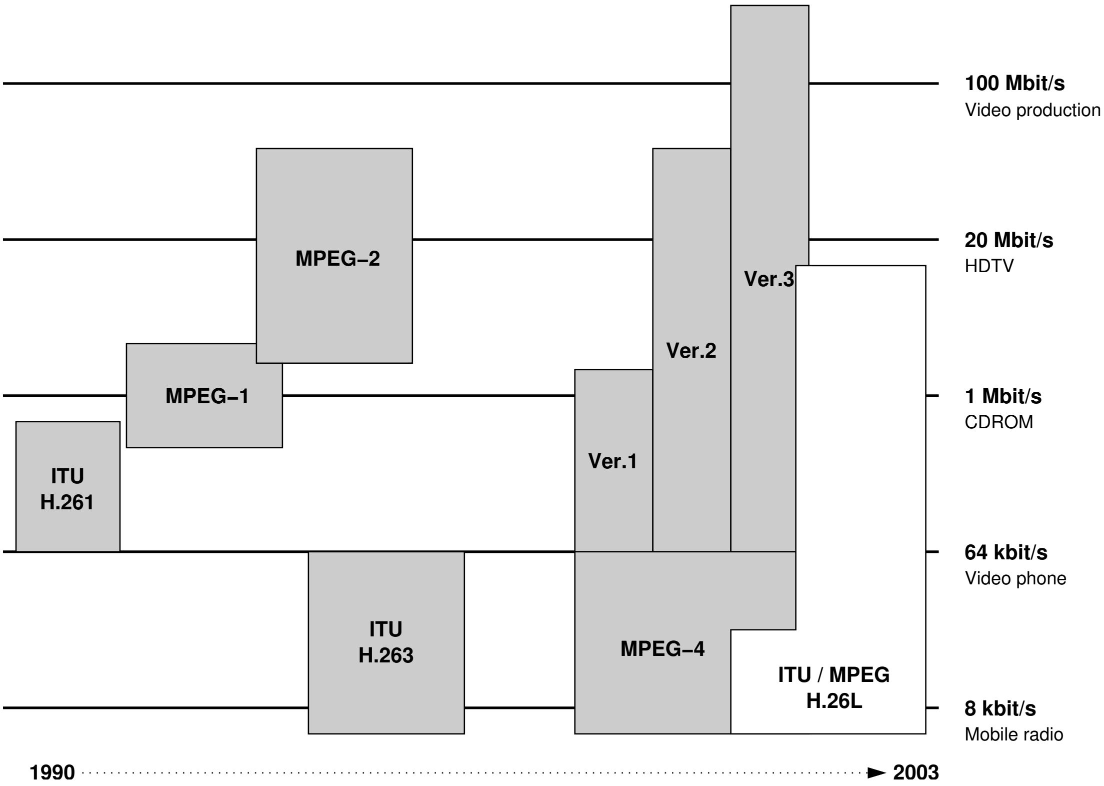


Figure 18: Video coding standards of ITU–T and ISO/MPEG.

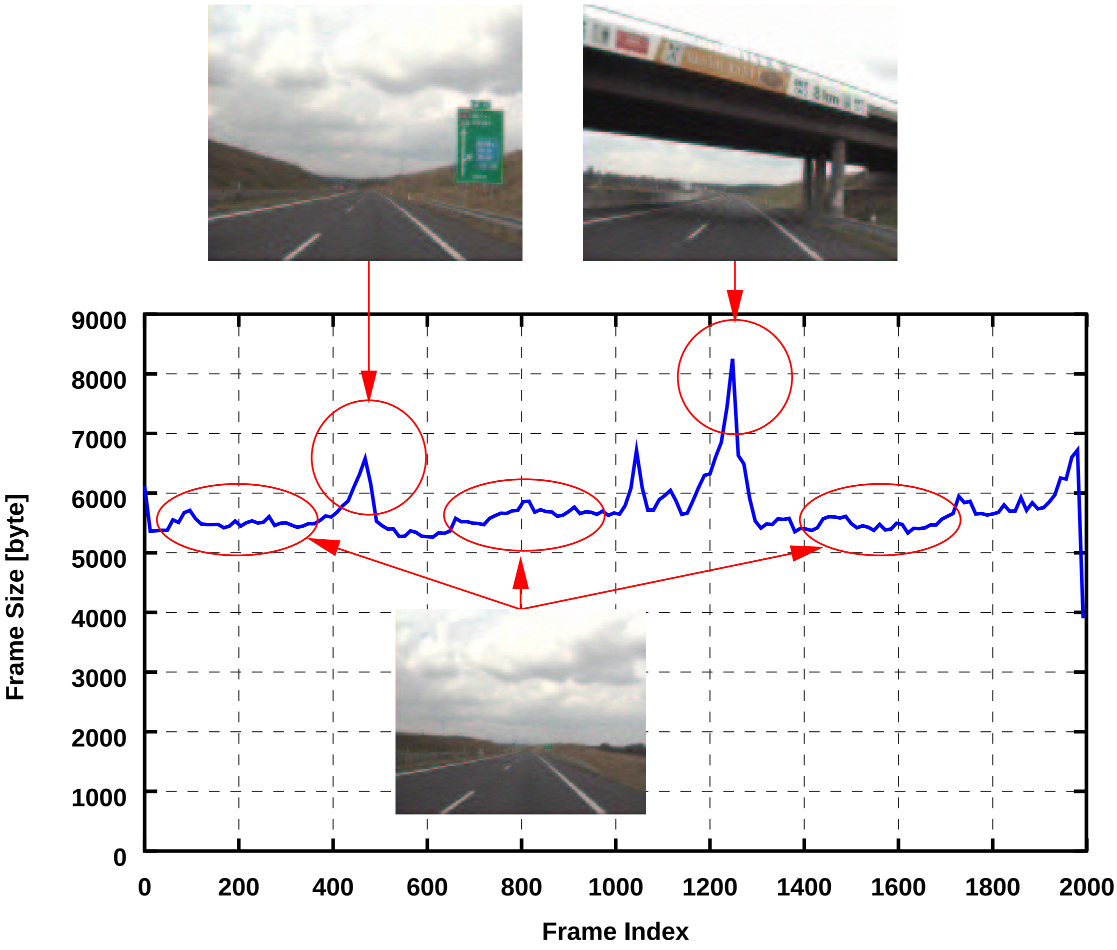


Figure 19: Frame sizes (averaged over GoPs) and content for the *Highway* sequence ($Q = 16$).

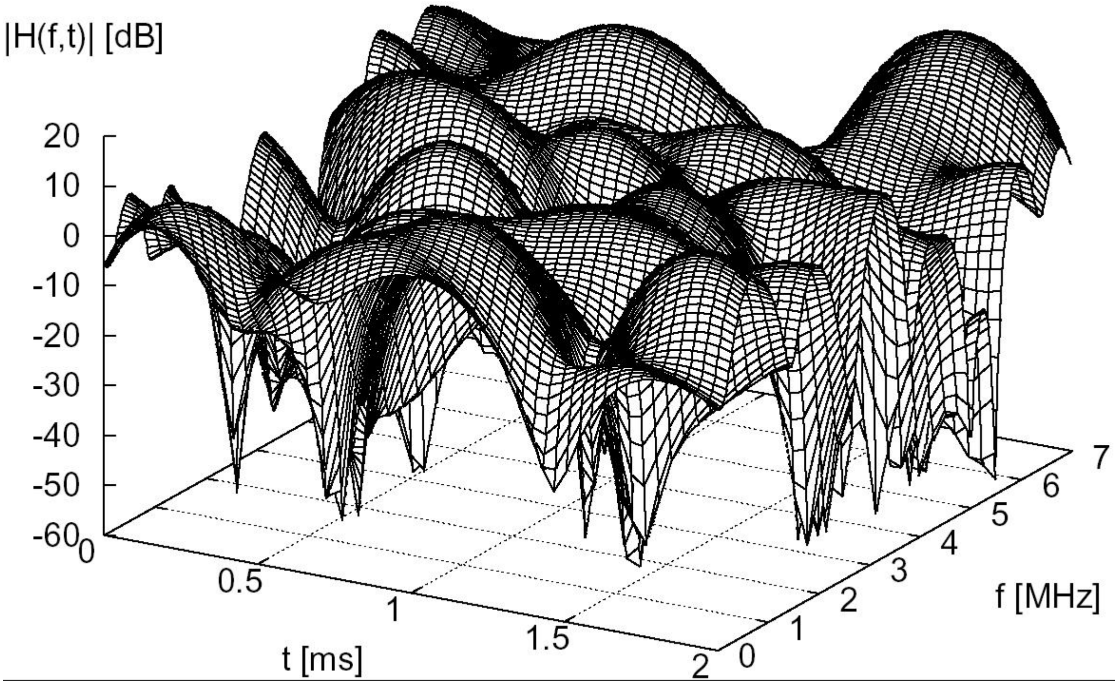


Figure 20: Typical signal quality fluctuations on wireless channel as a function of time and frequency. [1].

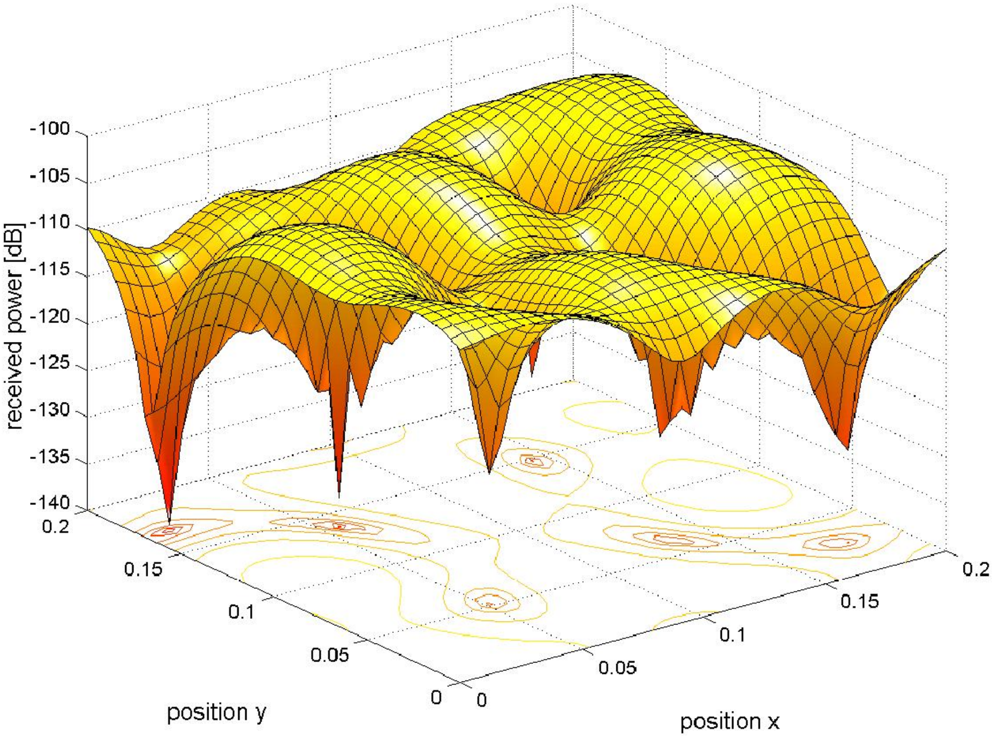


Figure 21: Typical signal quality fluctuations on wireless channel as a function of location [2].

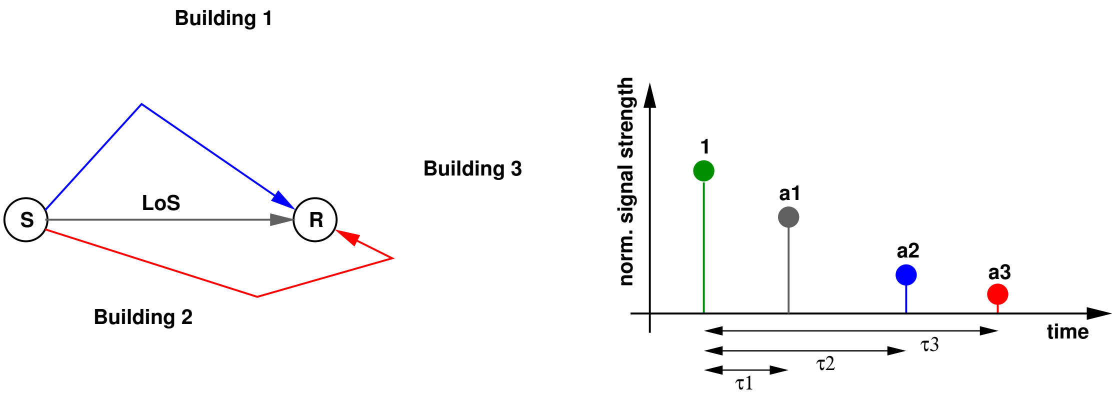


Figure 22: Example of multi–path propagation for three paths.

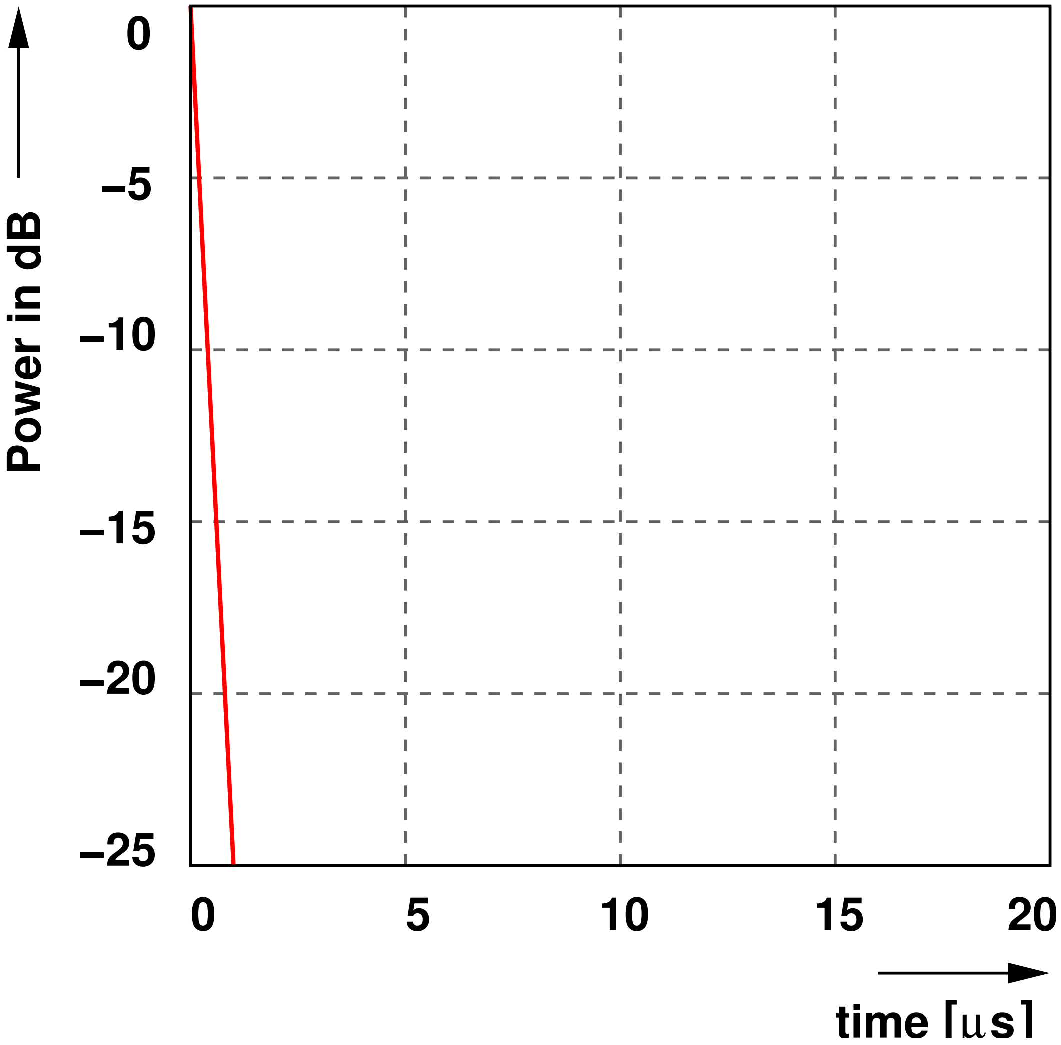


Figure 23: Delay spread for *Rural Area* (COST 207) [3].

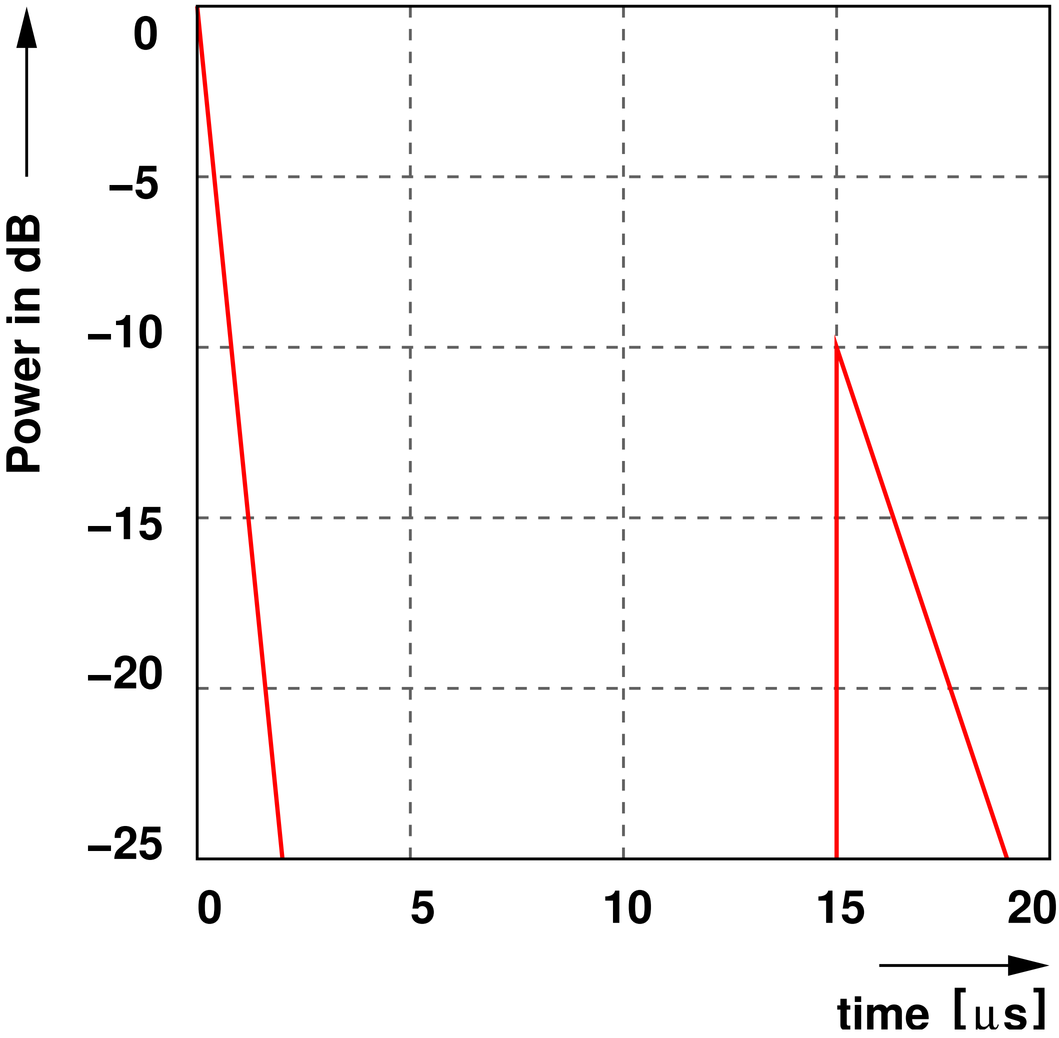


Figure 24: Delay spread for *Hilly Terrain* (COST 207) [3].

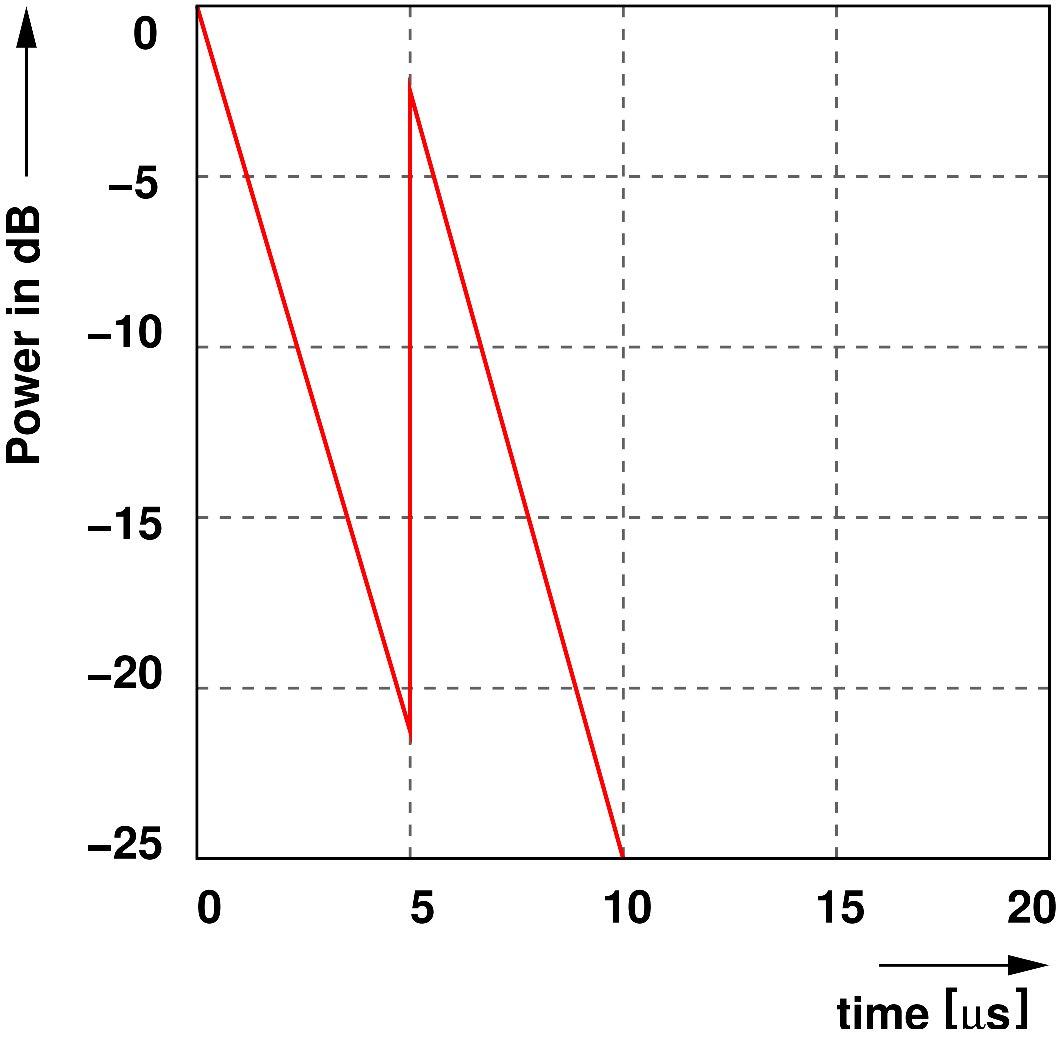


Figure 25: Delay spread for *Bad Urban* (COST 207) [3].

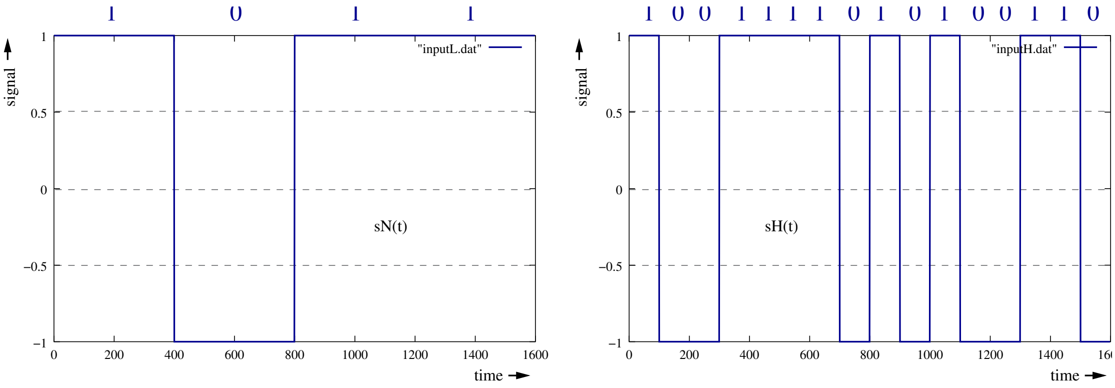


Figure 26: Low and high data rate signal.

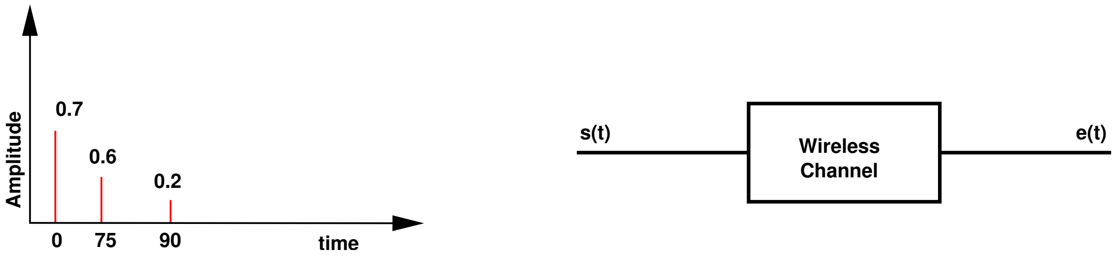


Figure 27: Channel Characteristics.

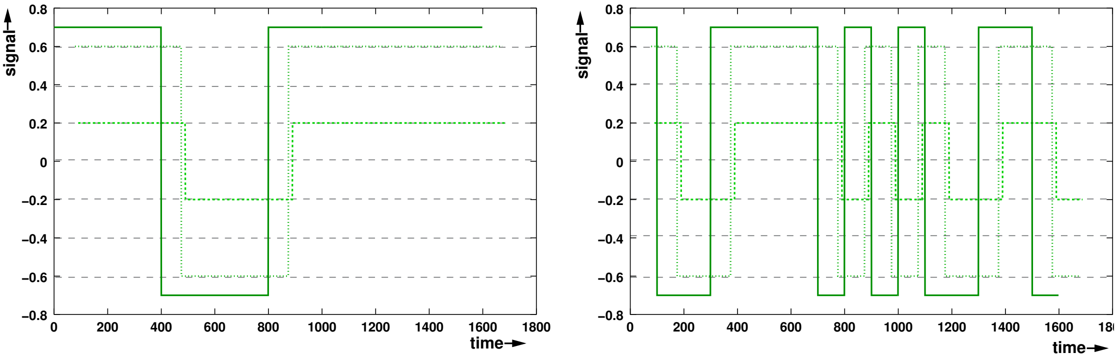


Figure 28: LoS and NLoS copies of the original signal.

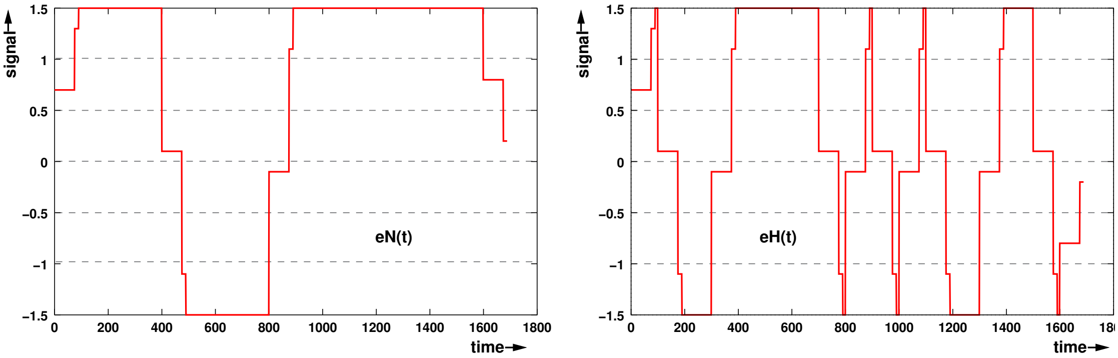


Figure 29: Received signal.

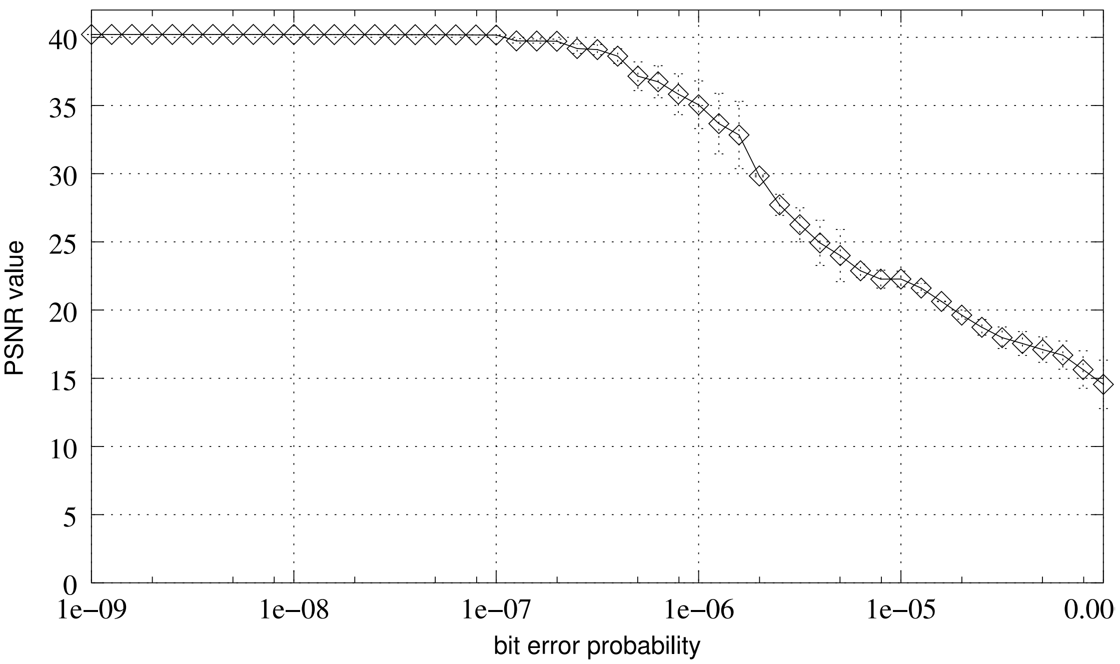


Figure 30: Average video frame PSNR as a function of bit error rate, IIII GoP

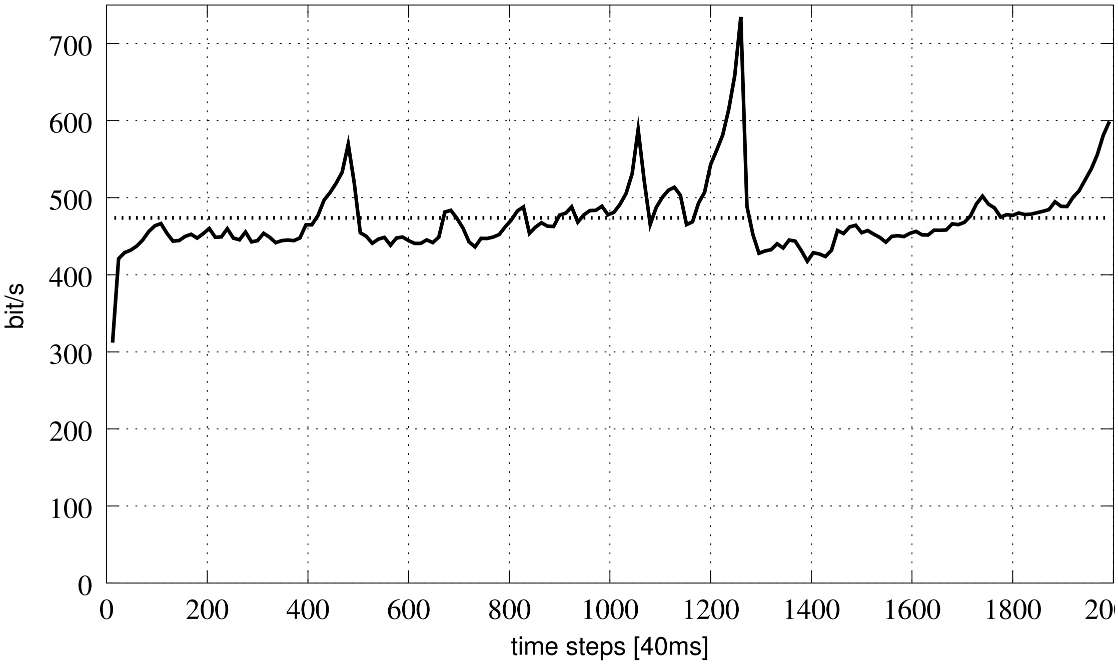


Figure 31: Video traffic (bit rate) as a function of time IIII GoP, avg. bit rate = 471.8 kbit/s

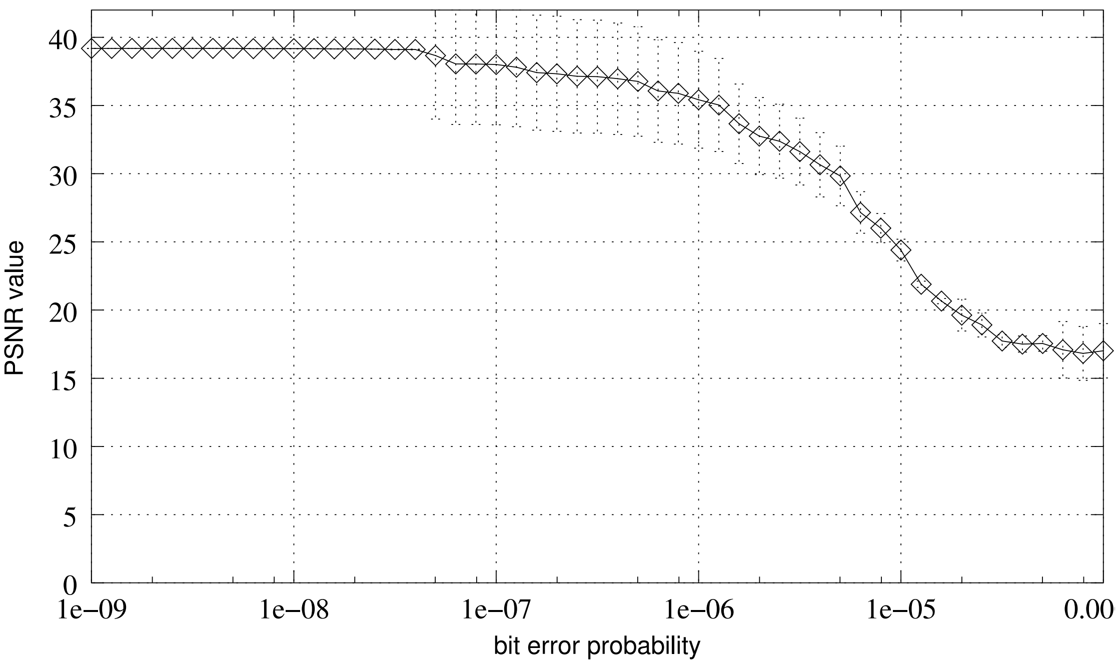


Figure 32: Avg. video frame PSNR as a function of bit error rate, IBBPB GoP

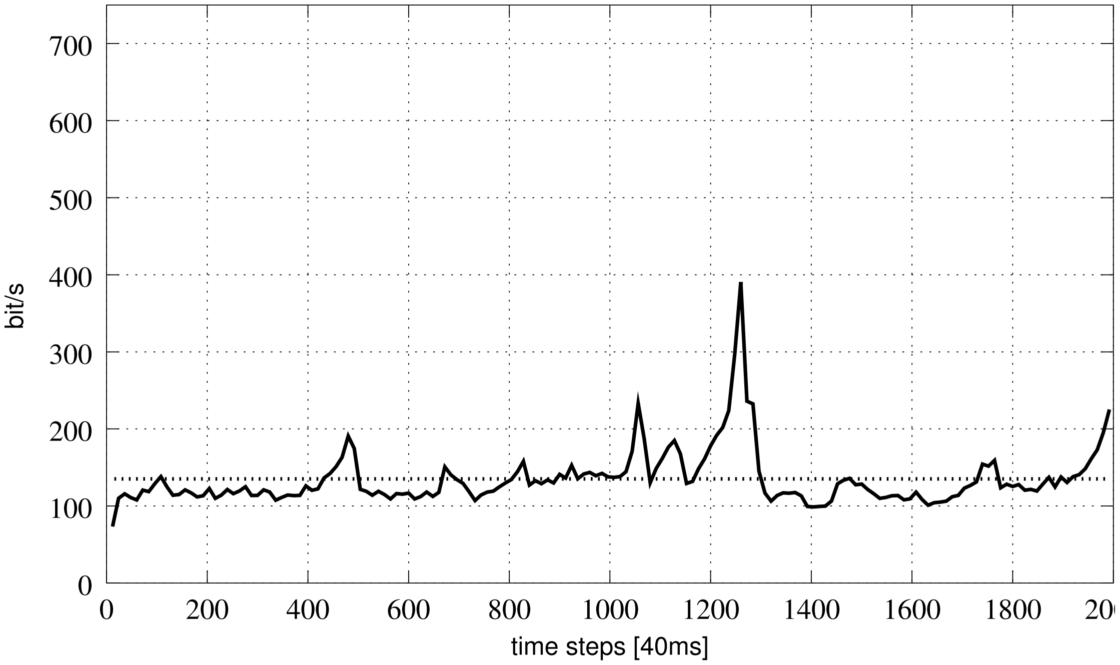


Figure 33: Bit rate as a function of time, IBBPB GoP, avg. bit rate 135.2 kbit/s

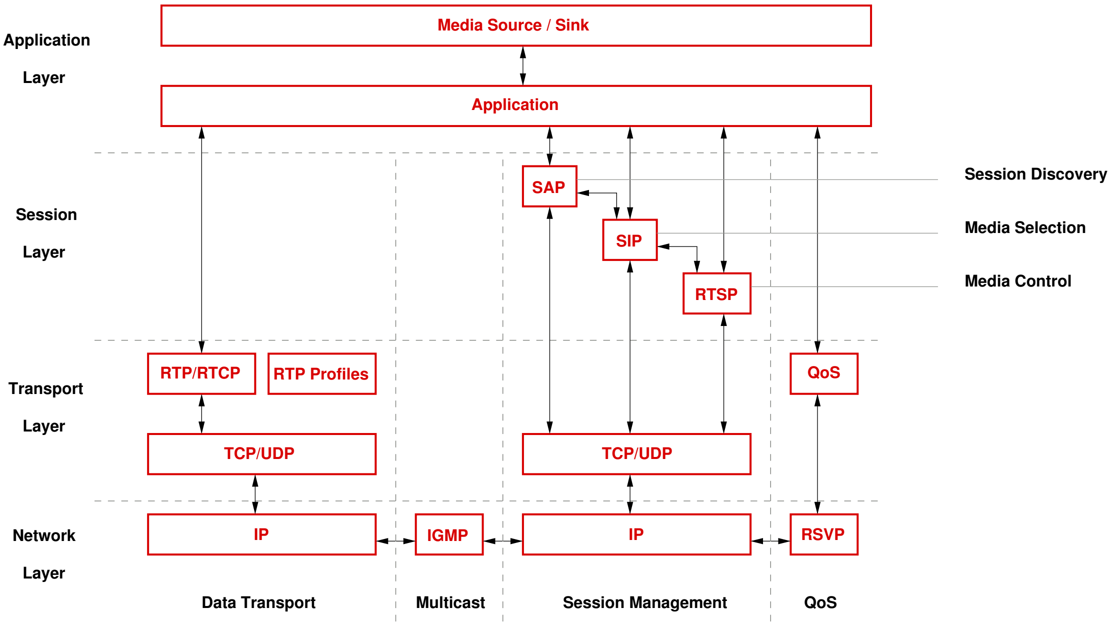


Figure 34: IP Protocol Suite.

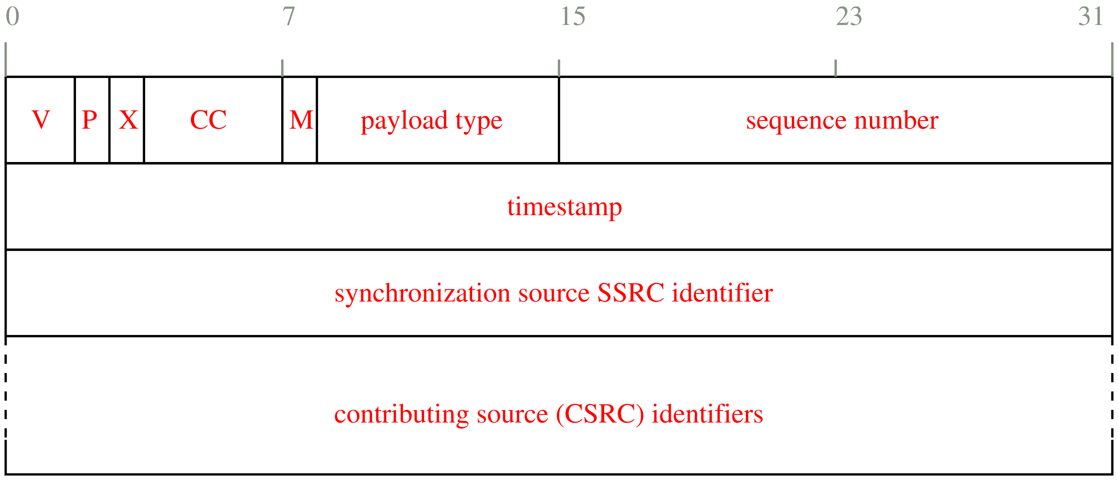

0
7
15
23
31
V
P
X
CC
M
payload type
sequence number
timestamp
synchronization source SSRC identifier
contributing source (CSRC) identifiers

Figure 35: RTP header format.

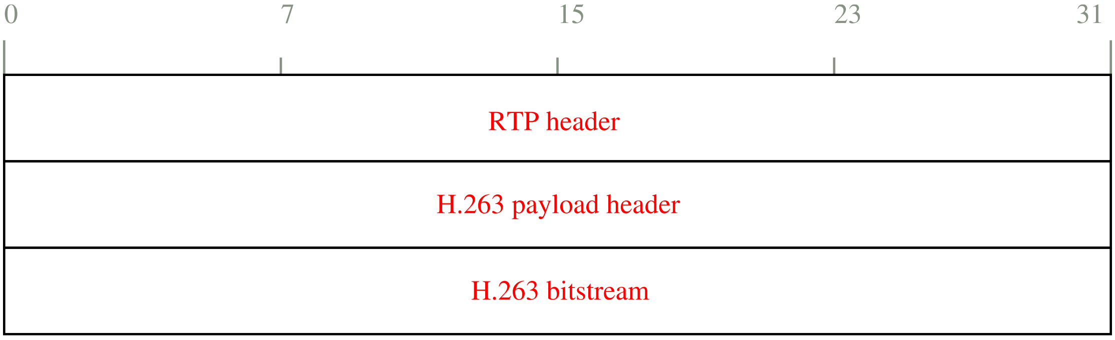


Figure 36: RTP payload header format and location.

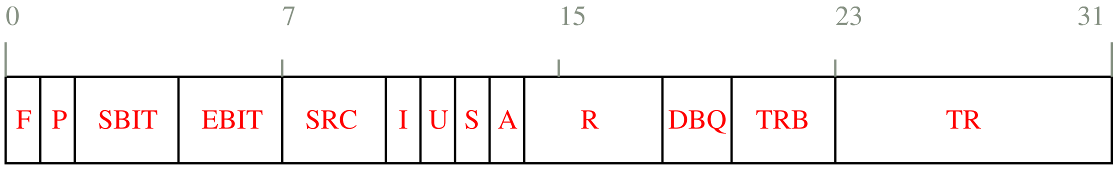
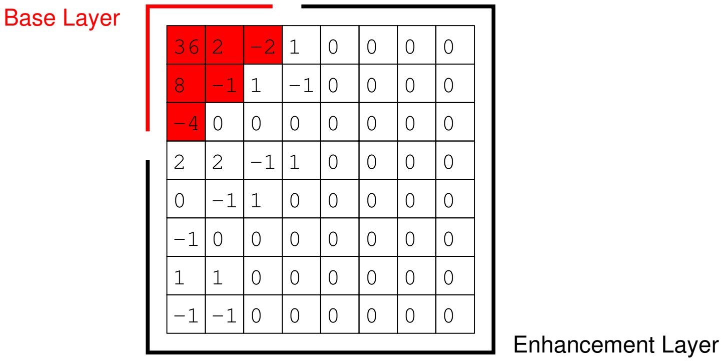


Figure 37: H.263 payload encapsulation header.



36	2	−2	1	0	0	0	0
8	−1	1	−1	0	0	0	0
−4	0	0	0	0	0	0	0
2	2	−1	1	0	0	0	0
0	−1	1	0	0	0	0	0
−1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
−1	−1	0	0	0	0	0	0

Figure 38: Data partitioning by priority break point setting.

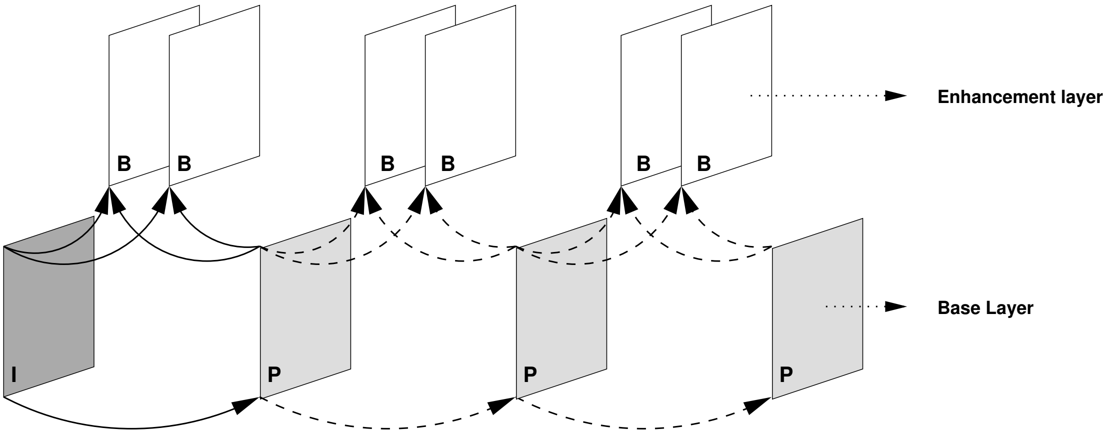


Figure 39: Temporal scalability with all B–frames in enhancement layer.

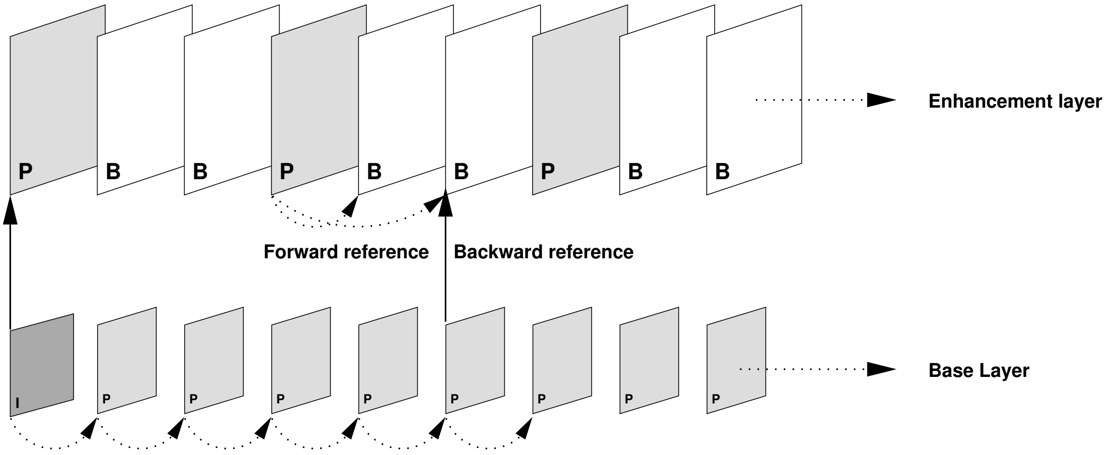


Figure 40: Example for spatial scalability and cross–layer references.

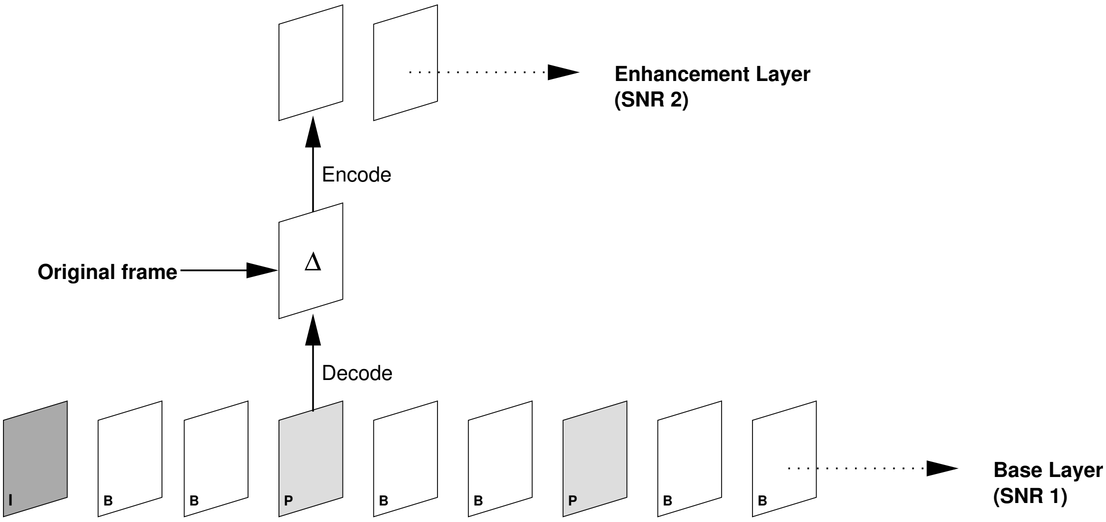


Figure 41: Example for SNR scalability.

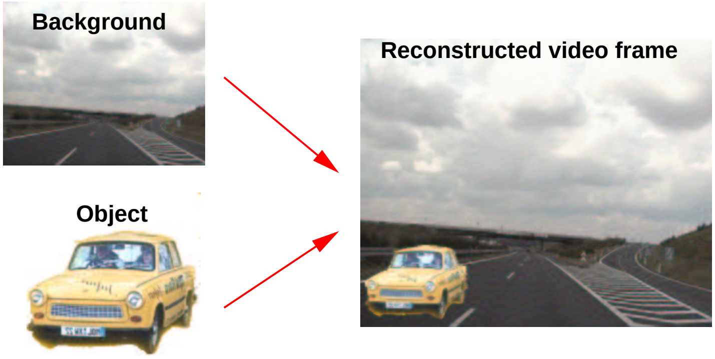


Figure 42: Example for object–based scalability.

Table 1: Characteristics for different video formats.

Standard	**QCIF** ITU–T H.261		**CIF** ITU–T H.261		**TV** ITU-R/CCIR-601		**HDTV** ITU-R 709-3	
Format	PAL [25 Hz]	NTSC [30 Hz]	PAL [25 Hz]	NTSC [30 Hz]	PAL [25 Hz]	NTSC [30 Hz]	PAL [25 Hz]	NTSC [30 Hz]
Sub–sampling	4:2:0		4:2:0		4:2:2		4:2:2	
Columns (Y)	176		352		720		1920	
Rows (Y)	144		288		576	480	1080	
Columns (U,V)	88		176		360	360	960	
Rows (U,V)	72		144		576	480	1080	
Frame size [byte]	38016		152064		1244160	1036800	4147200	
Data Rate [Mbit/s]	7.6	9.1	30.4	36.5	248.8	298.6	829.4	995.3

Table 2: Frame statistics for the *Highway* sequence (QCIF, $Q = 16$)

Aggregation Level	**Attribute**	**Value**
Video Sequence	Total size [byte]	9047512
	Total # of frames	1999
	Playing time [s]	79.96
Frame	Minimum size [byte]	3347
	Maximum size [byte]	9178
	Mean size [byte]	4526.02
	Frame size variance	1261651.93
	Frame coefficient of variation	0.25
	Mean frame bit rate [bis/s]	905203.80
	Peak frame bit rate [bit/s]	2936960.00
	Peak to mean ratio	3.24
	Mean I–frame size	7861.53
	Mean P–frame size	4526.02
	Mean B–frame size	4036.88
GoP	Number of GoPs	166
	Minimum GoP size [byte]	7283
	Maximum GoP size [byte]	84719
	Mean GoP size [byte]	54205.58
	Mean GoP bit rate [bit/s]	903426.33
	Peak GoP bit rate [bit/s]	1411983.3
	GoP peak to mean	1.56
	GoP variance	20190776.03
	GoP coefficient of variation	0.08