Quality of Service Support in Wireless Multi–Code CDMA Systems

vorgelegt von
Diplom-Elektrotechniker
Frank H.P. Fitzek
aus Oldenburg

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.–Ing.-

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.–Ing. Klaus Petermann
Gutachter: Prof. Dr.–Ing. Adam Wolisz
Gutachter: Prof. Dr.–Ing. Peter Noll
Gutachter: Prof. Michele Zorzi

Tag der wissenschaftlichen Aussprache: 06. Juni 2002

Berlin 2002
D 83

For my family

Contents

List of Figures

List of Tables

1 Introduction

Wireless communications have undergone a tremendous development in the last decade. Due to people's vast growing need for mobile conversation, wireless networks emerged and expanded very fast. The new degree of freedom that wireless communications provide, has made wireless devices very popular. Wireless communications is growing so rapidly that the number of wireless customers is estimated to be over one billion by 2005 [2]. As shown in Figure 1.1 there are around 400 million wireless customers world–wide at the beginning of the year 2000 and it is estimated that there will be 1800 million customers in the year 2010. These estimates are based on the the **U**niversal **M**obile **T**elecommunications **S**ystem (*UMTS*) Forum [8]. At the same time, the use of the Internet is also growing drastically. According to a recent study more people will access the Internet from wireless (such as web-enabled cell phones and PDAs) than from wire–line devices [1, 2] within this decade. Figure 1.2 depicts the expected number of Internet customers for the next years. The figure also shows the expectation that wireless Internet access will dominate over the wired Internet access in the future.

The need for speech services was the origin of first and second generation mobile and wireless communication systems such as **G**lobal **S**ystem for **M**obility (*GSM*), **D**igital **A**dvanced **M**obile **P**hone **S**ystem (*DAMPS*), **P**ersonal **D**igital **C**ellular (*PDC*), and **I**nterim **S**tandard 95 (*IS95*) [9]. In contrast to the first and the second generation communication systems which are limited to voice services, future wireless communication customers will ask for a wider range of services. In addition to voice services this range includes data services and real–time services, which are referred to as multimedia services. Multimedia services include the provisioning of video, voice, data, gaming, and many other services. Due to the popularity of accessing multimedia content on the **W**orld **W**ide **W**eb (*WWW*) through wireless devices, third generation (*3G*) and beyond wireless communications will be dominated by multimedia applications. According to the UMTS Forum [1] the utilization of multimedia services will increase significantly after the year 2000. Furthermore they predict that over 60% of the traffic in Europe will comprise of multimedia applications by 2010 [8].

Multimedia applications have heterogenous **Q**uality **o**f **S**ervice (*QoS*) requirements in terms of bandwidth, delay, delay variation and losses [6]. Nevertheless, the typical customer expects that all the communication services known from wire–line communications based on the Internet will also be provided in the wireless format. But the migration of the Internet to wireless devices poses formidable challenges for communication protocols. Until recently, transport protocols that support multimedia applications were designed for wired networks. But the rapid increase of wireless devices requires solutions that are suitable for wireless networks. Especially in wireless communications this induces compli-

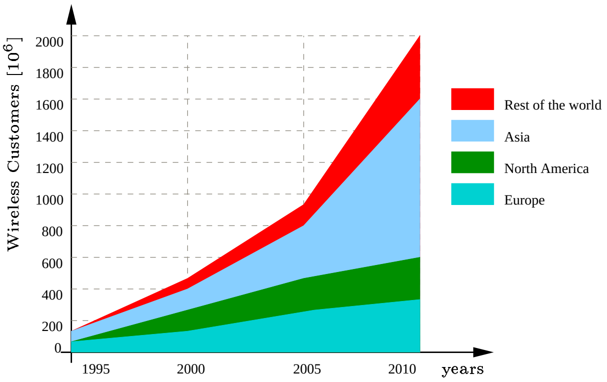


Figure 1.1: World–wide evolution of the number of mobile and wireless customers by region [1].

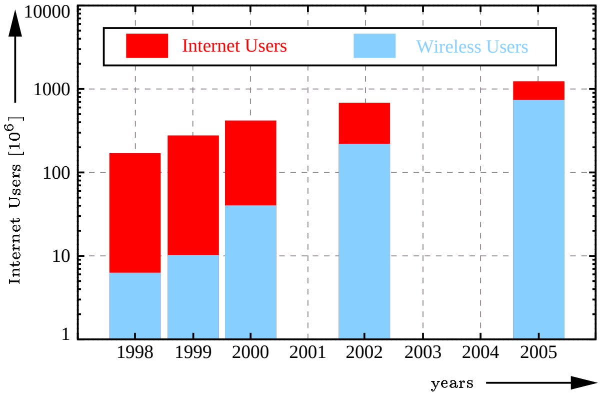


Figure 1.2: World–wide evolution of the number of Internet hosts [2].

cations due to the severe and permanently changing propagation conditions in the mobile and wireless radio channel. In contrast to wired links, wireless links are unreliable; they exhibit a varying bit error probability and temporary outage periods, during which the bit errors are strongly correlated. From the application point of view, error–prone channel conditions are recognized by a reduced throughput, higher loss rate, higher jitter and/or higher delay. Instead of attempting to tolerate the unstable service from the physical layer such as adaptive applications [10, 11, 12, 13], we introduce a new mechanism that helps to stabilize the QoS in terms of throughput, loss rate, jitter and delay at the data link layer. We resorted to this mechanism within the data link layer, because in that way higher layer protocols do not have to be changed at all. Thus, the same protocols as for wired networks can be used.

Nevertheless, data link packets are affected by errors on the wireless link and might be corrupted. In order to reduce the loss rate seen within the data link layer, error correction schemes such as **F**orward **E**rror **C**orrection (*FEC*) or **A**utomatic **R**epeat Re**Q**uest (*ARQ*) are used. In the case of *static* FEC, the FEC overhead is chosen such that transmission is reliable even in the worst channel conditions. However, this overhead would be redundant in case of *good* channel conditions and hence results in wastage of resources. As the spectrum is a limited resource this is not an effective mechanism. On the other hand the throughput is stable, however reduced. Improvements for the throughput at the expense of an increased delay variation and stability of the throughput can be achieved if the FEC code is chosen *dynamically*, dependent on the channel state. This, however, leads to inefficiencies in some important applications, such as MPEG–4 encoded video. In the case of such an application encoding is done based on the channel rate, which changes over a small time scale in case of *dynamic* FEC. However, the MPEG–4 coding can only be done over a time period of a **G**roup **of** **P**icture (*GoP*), which is much larger than the channel variation time scale.

In the case of ARQ the cost is an increased delay variation, because stored packets have to wait until the retransmission process of the corrupted packet is successful. Figure 1.3 shows a simplified ARQ transmission scheme for one transmitter. The input to the queue is an arrival process at a rate λ and the queue is served at rate μ_S. After the server a decision is made as to whether this service was successful or not. In case of an unsuccessful transmission the service has to be repeated and this happens with probability $\alpha(t)$, which varies over time. In other words one could think of a queue that is filled with packets and these packets are transmitted over a wireless link with a given capacity corresponding to μ_S. According to the ARQ strategy corrupted packets have to be retransmitted. To avoid buffer overflows the services rate μ_S has to be larger than the arrival rate λ and the retransmission rate.

The reduced throughput of the *static* FEC and the increased delay–jitter of *dynamic* FEC and ARQ both lead to performance degradation of multimedia applications. Therefore we came up with an idea of multiple servers with individual service rate μ_M given in Figure 1.4. In this case, even with $\mu_M = \lambda$, no buffer overflow will occur if we could only activate another server every time the service failed before. In contrast to the example given in Figure 1.3 the delay and jitter of the stored packets with the approach of multiple servers is bounded. To avoid buffer overflows the serving rate (multiple of μ_M) has to be larger than the arrival rate λ and the retransmission rate. Bounded delays and jitter values are needed to support multimedia streams over the wireless link.

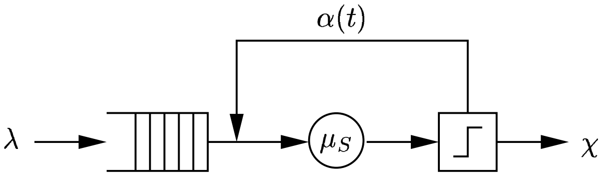


Figure 1.3: Single–server with service rate μ_S and arrival rate λ.

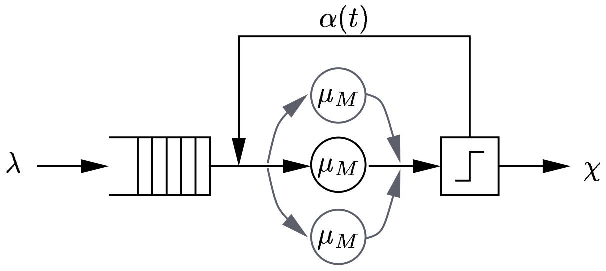


Figure 1.4: Multi–server with service rate μ_M and arrival rate λ.

Unfortunately, in real systems, the number of servers corresponding to channels is limited using orthogonal resources such as in **T**ime **D**ivision **M**ultiple **A**ccess (TDMA), **F**requency **D**ivision **M**ultiple **A**ccess (FDMA), or **C**ode **D**ivision **M**ultiple **A**ccess (CDMA) [14, 15, 16, 17, 18] with orthogonal spreading sequences. Limited resources have to be assigned by a centralized entity among transmitters to avoid *collisions*. For cellular systems the assignment can be easily done within the downlink[1]. But for the uplink [2] the assignment could lead to an overwhelming amount of signaling. On the other hand, there is the possibility to use **P**seudo–**N**oise (*PN*) sequences in combination with CDMA for the uplink. PN sequences are non–orthogonal *resources*. In this case there would be a high number of resources which could be assigned in a distributed fashion among transmitters. In contrast to orthogonal resources, the servers have an impact on each other and will lead to a performance degradation of each transmitter when more servers are active. Regarding Figure 1.4 the bundling of three servers with rate μ_S each will lead to an overall serving rate smaller than $3 \cdot \mu_S$. Each active server has an impact on adjacent servers. Therefore, orthogonal sequences are used within the downlink, while PN sequences are used for the uplink to avoid signaling.

[1] Transmissions from the base station towards the wireless terminal.

[2] Transmissions from the wireless terminals towards the base station.

CDMA is the most prominent access technology over the air interface for the upcoming 3G wireless communication systems. In January 1998, the European standardization body for mobile radio systems, the **E**uropean **T**elecommunications **S**tandard **I**nstitute – **S**pecial **M**obile **G**roup (*ETSI–SMG*), has agreed on a radio access scheme for third generation mobile radio systems, called UMTS using the **C**ode **D**ivision **M**ultiple **A**ccess (CDMA) technology. In parallel to the European activities, the 3G partnership project (3GPP) was formed to work towards a global 3G mobile radio standard. Thus, CDMA technology will be used for 3G mobile communication systems even in Japan and North–America.

Thus, the main idea behind the approach presented in this work exploits the capability of wireless devices to transmit on multiple CDMA channels in the uplink. The capability to send and receive on multiple channels is made possible by **M**ulti–**C**ode CDMA (*MC–CDMA*). In contrast to [19, 20], where a centralized approach was introduced, we allow the wireless terminals to use multiple channels for the uplink autonomously. Thus, our approach operates in a distributed fashion minimizing the amount of signaling. Therefore, our approach can be used in cellular as well as Ad–hoc networks. Using multiple channels in parallel, however, can degrade the overall system capacity if PN spreading sequences are chosen. Each additional active channel will lead to a degradation of the **S**ignal to **N**oise **I**nterference **R**atio (*SNIR*), and hence will lead to higher bit error probabilities. There are many techniques of how to use multiple channels. We compare different multi–code based link–layer transmission strategies and demonstrate which strategies lead to performance improvements (in terms of capacity) for multimedia applications within a 3G wireless CDMA environment. We discuss the trade–off between certain spreading sequence families and their need for signaling between senders and receivers. Our focus is on the video transmission in the uplink direction in a cellular wireless system, i.e., from wireless terminals to a central base station. Our extensive simulations indicate that video transmission gives good performance in terms of video quality and capacity[3] when our approach is employed.

The work is structured as follows. In Chapter 2 we discuss the different requirements of the transport and application layers in supporting various multimedia applications. We discuss one of the multiple access techniques in a cellular network, CDMA, and the different schemes used to support variable bit rate data. We also discuss briefly the characteristics of a wireless channel and its impact on the performance of the CDMA scheme. In Chapter 3 we briefly summarize the related works in link–level QoS support. Here we discuss physical layer techniques such as power control in supporting different QoS requirements. The other data link–level techniques such as ARQ and multiple channel CDMA techniques are discussed. We have proposed a new technique **S**imultaneous **MAC** **P**acket **T**ransmission (*SMPT*) [21, 22, 23] that provides a solution for the problem faced by other data link layer protocols. This technique and its other variations are discussed in detail in Chapter 4. Chapter 5 discusses the generation of traffic traces using different encoders, such as MPEG–4 and H.263. Statistics are gathered from these traffic traces which are very important in modeling the same for the purpose of simulations, to be as close as possible to the real–life scenario. Chapter 6 gives the model definitions. We discuss the scenario, the source models, and the wireless link model. We then give a short

[3] Number of supported wireless terminals within the cell.

description of the simulation tools used within this work. In Chapter 7, we have done the theoretical analysis of the SMPT system. We have calculated the maximum number of users that can be allowed in the system before the outage probabilities of the system exceeds a maximum acceptable value. Chapter 8 discusses the results such as throughput and system capacity for different SMPT schemes supporting different transport layer protocols. Finally, we conclude this work and give a short outlook for SMPT in Chapter 9.

2 Basics

This chapter is divided in three main parts. First we describe the wireless channel and how the nature of the channel affects the system performance. Then we give an overview of requirements of the transport and the application layer supporting multimedia services. In summary we show that real–time video applications have the tightest requirements for the under–laying protocol layers. Moreover, we also conclude that the provision of several types of services with different QoS over the wireless link becomes difficult. At last we describe the CDMA technique and how one of the techniques used by CDMA to support multi rate traffic could be used in our work.

2.1 Wireless Communication Channels

The wireless channel places fundamental limitations on the performance of wireless communication systems [24]. Unlike wired channels, which are predicatable, wireless channels are extremely random. Basic propagation mechanisms are reflection, attenuation, diffraction and scattering. Reflections result in signals being transmitted over multiple paths between sender and receiver (see Section 2.1.2). Attenuation describes the situation that the power of the transmitted signal is received with less power (see Section 2.1.1).

2.1.1 The Free Space Propagation Model

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. Then, the strength 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. In [24, 18] it is shown that

$$P_r(d) = \frac{P_s G_t G_r \lambda^2}{(4\pi d)^2 L}. \tag{2.1}$$

The free space equation shows that the received power declines with the square of the distance. In Section 2.3.2.2 we show that this behavior degrades the system performance and we present mechanisms that overcome the described problem.

2.1.2 Multi–path Propagation

Considering a mobile[1] 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 probably, but not necessarily, 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 possible, because of mountains, buildings, and moving obstacles within the wireless environment, where the radiated signals are reflected and diffracted. If multiple copies of the unique sender–side signals arrive at the receiver this is called *multi–path*. Due to the reflections and diffractions the received multi–path signals differ in amplitude, phase, and delay. Therefore it is possible that the copies interfere with each other. In case multi–path signals have different arrival times at the receiver, reduction of the signal strength at the receiver can occur. This kind of interference is called **I**nter **S**ymbol **I**nterference (*ISI*), because symbols of the same signal interfere with each other. The amount 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 certain energy. A typical example for multi–path communication is given in Figure 2.1 for one LoS path and two NLoS paths. The sender transmits a signal at any time instance, after τ_1 the LoS signal arrives at the receiver. The NLoS signals arrive τ_2 and τ_3 at the receiver after the signal was transmitted. The delay spread of 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. Multi–path degrades the system performance. A limitation of the ISI can be achieved by equalizers, RAKE sender/receiver (see Section 2.3.2.1), and directional antennas.

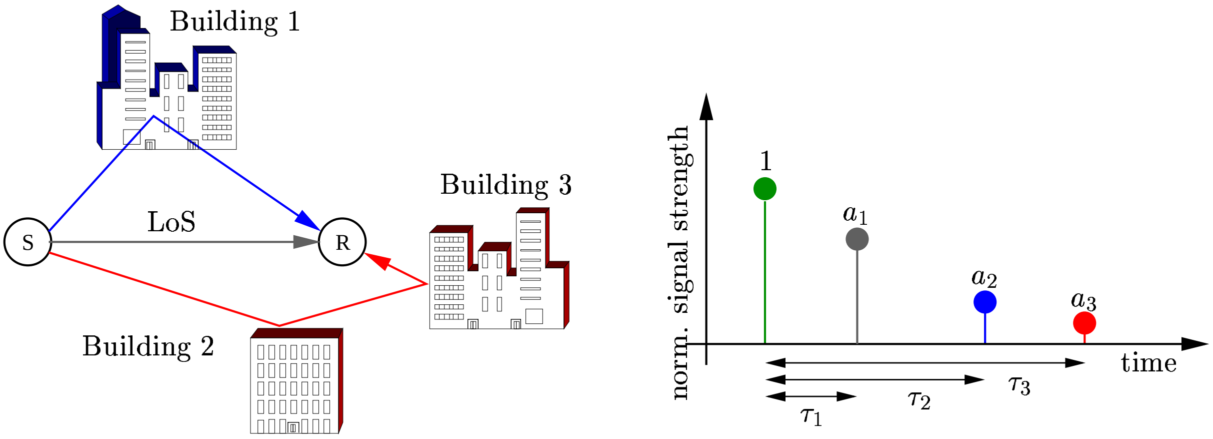


Figure 2.1: Example of multi–path propagation for three paths.

Not only the multi–path interference, but also the *Doppler–Effect* may have a negative impact on the characteristics of the wireless channel. Due to mobility the signal's quality

[1] Mobile in the sense that the sender has a relative velocity in comparison to the receiver.

decreases. Because of the multi–path interference and the mobility of sender and/or receiver the receiving power changes about $30 - 40$dB compared to the mean power value [25]. A typical example for the behavior of the wireless channel is given in Figure 2.2 and 2.3. Obviously there is a significant fluctuation in the change of the power level over the time and the frequency domain (given in Figure 2.2) as well as over the location (given in Figure 2.3 defined by x and y coordinates). A degradation of the signal power over a long time period results in *bursty* or correlated errors on the wireless link [26, 27, 28]. The fading duration causes bursty errors, while the burst–length is given by the fading–length[29].

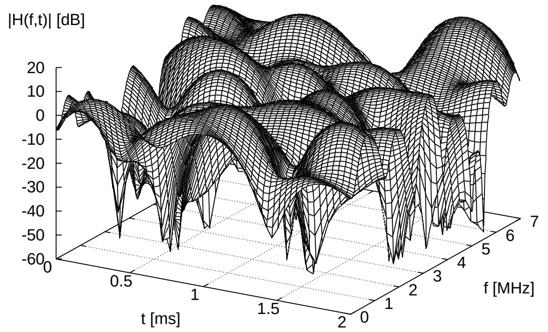


Figure 2.2: Example of the channel quality for a time varying and frequency selective physical channel [3].

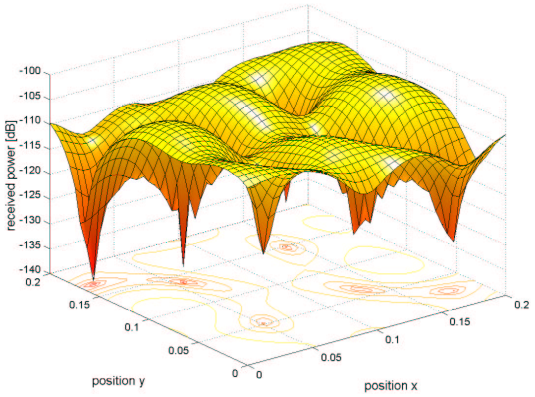


Figure 2.3: Example of the channel quality for a location dependent physical channel [4].

In contrast to the given example in Figure 2.1, in real systems the delay spread and the number of paths are strongly correlated to the environment of sender and receiver. In [5] the delay spread of different environments were given. Figure 2.4 shows these delay spread values for *Rural Area*, *Hilly Terrain*, and *Bad Urban*.

2.2 Requirements of the Transport and Application Layer

The typical customer expects that all the communication services known from wire–line communication based on the Internet are also provided in the wireless format. Therefore throughout the whole document we consider an all **I**nternet **P**rotocol (*IP*) based wireless cellular communication system where multiple **W**ireless **T**erminals (*WTs*) operate within a single cell. All **W**ireless **T**erminals (*WTs*) communicate with one central **B**ase **S**tation (*BS*), whose coverage defines the cell boundaries as shown in Figure 2.7. The BS is connected via the backbone to other end–systems, such as servers or other WTs which provide specific IP based services to the customers. For the purpose of this work we assume that all information of the backbone is delivered to the BS. The protocol stack for the WT is depicted in Figure 2.8.

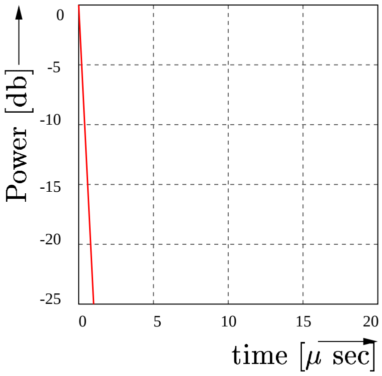


Figure 2.4: Delay spread for *Rural Area* (COST 207) [5].

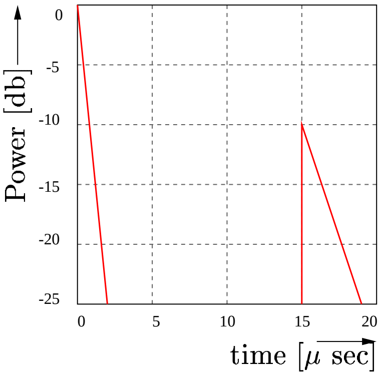


Figure 2.5: Delay spread for *Hilly Terrain* (COST 207) [5].

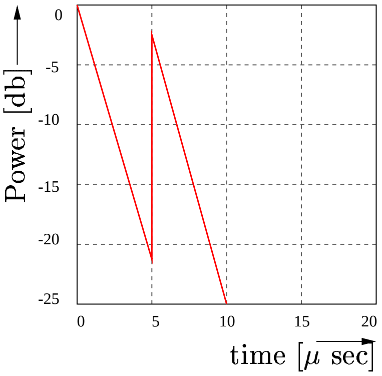


Figure 2.6: Delay spread for *Bad Urban* (COST 207) [5].

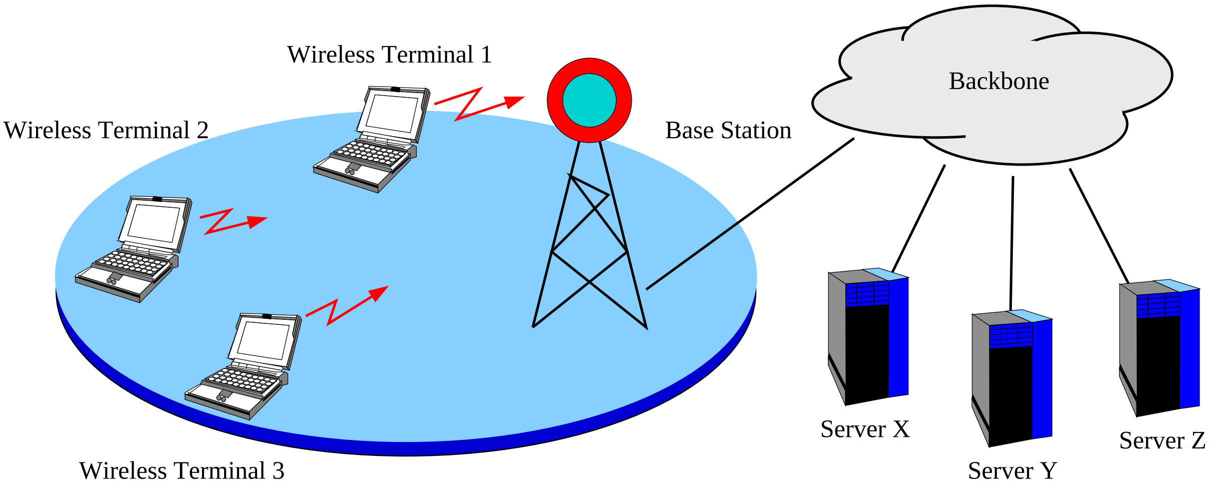


Figure 2.7: System architecture with several wireless terminals and one base station.

Different IP based applications (e.g. for video, voice, and data) can be supported by the WTs. As mentioned earlier we assume the IP (version 4 or 6) at the network layer. Within the transport layer we assume **T**ransport **C**ontrol **P**rotocol (*TCP*) and **U**ser **D**atagram **P**rotocol (*UDP*). Many applications are based on these protocols (see Figure 2.8). TCP provides a connection-oriented and reliable flow between two terminals, while UDP provides a connection-less and unreliable datagram service. Each of the protocols is used for different application types. In the following we introduce these transport protocols and give some corresponding applications.

TCP provides a reliable connection between two terminals on top of a less reliable communication network. It is standardized in RFC 793. TCP is designed to detect damaged, lost and duplicated segments within a given data flow. Therefore each segment is identified by a sequence number. If a set of segments is transmitted successfully the sender waits for an acknowledgment from the receiver. The acknowledgment identifies the correct received segment by its segment number. After sending a segment the sender will wait only a pre-specified time by calculating timeouts, also called **R**etransmission **T**ime **O**ut (*RTO*), for each segment [30]. If the acknowledgment is not received within this time, the whole segment will be retransmitted over and over again until the acknowledgment is received. The **R**ound **T**rip **T**ime (*RTT*) for each segment is measured by the sender side TCP entity and will be used to calculate the RTO. Calculation of the timeouts with use of RTT is described in [31, 32]. Obviously timeouts lead to throughput degradation and increased delay due to the retransmissions that are performed, and also the reduction of the **C**ongestion **W**indow (*CW*). One of the key features of TCP is its flow control. For each connection TCP provides a CW. The size of the congestion window indicates a range of acceptable sequence numbers beyond the last segment successfully received. Furthermore the size of the congestion window corresponds to the maximal allowed throughput. In case acknowledgments are missing the CW will be shrunk (depending on the TCP mechanism, either to one or to the half). By this means fairness shall be guaranteed among all TCP connection and the maximum overall throughput can be achieved. The reliability and congestion control mechanisms described above require that TCP maintains status information for each data flow.

Roughly speaking, data applications do not tolerate any loss; they are however, delay tolerant. There exist a few data applications on top of TCP. The most common applications are the FTP and the HTTP. FTP was first proposed in 1971 and therefore it was one of the *early* applications for the Internet. It allows exchange of data between two terminals. HTTP is one key feature of the **W**orld **W**ide **W**eb (*WWW*). As FTP it allows to exchange data between terminals, but furthermore it contains meta-information about the data transferred. HTTP is used between WWW clients and servers for hypertext document service. The traffic characteristics of the well–known application, web file transfer, have been studied extensively in [33, 34].

On the other side **U**ser **D**atagram **P**rotocol (*UDP*) provides a connection-less and unreliable datagram service. Application programs send messages to other programs with a minimum of protocol mechanism. The protocol is transaction oriented, and features such as delivery guarantee and duplicate protection, like those given by TCP, are not provided. For real–time communication on the Internet UDP is a suitable transport protocol. In contrast to TCP, which requires a reliable transmission, UDP transmission allow

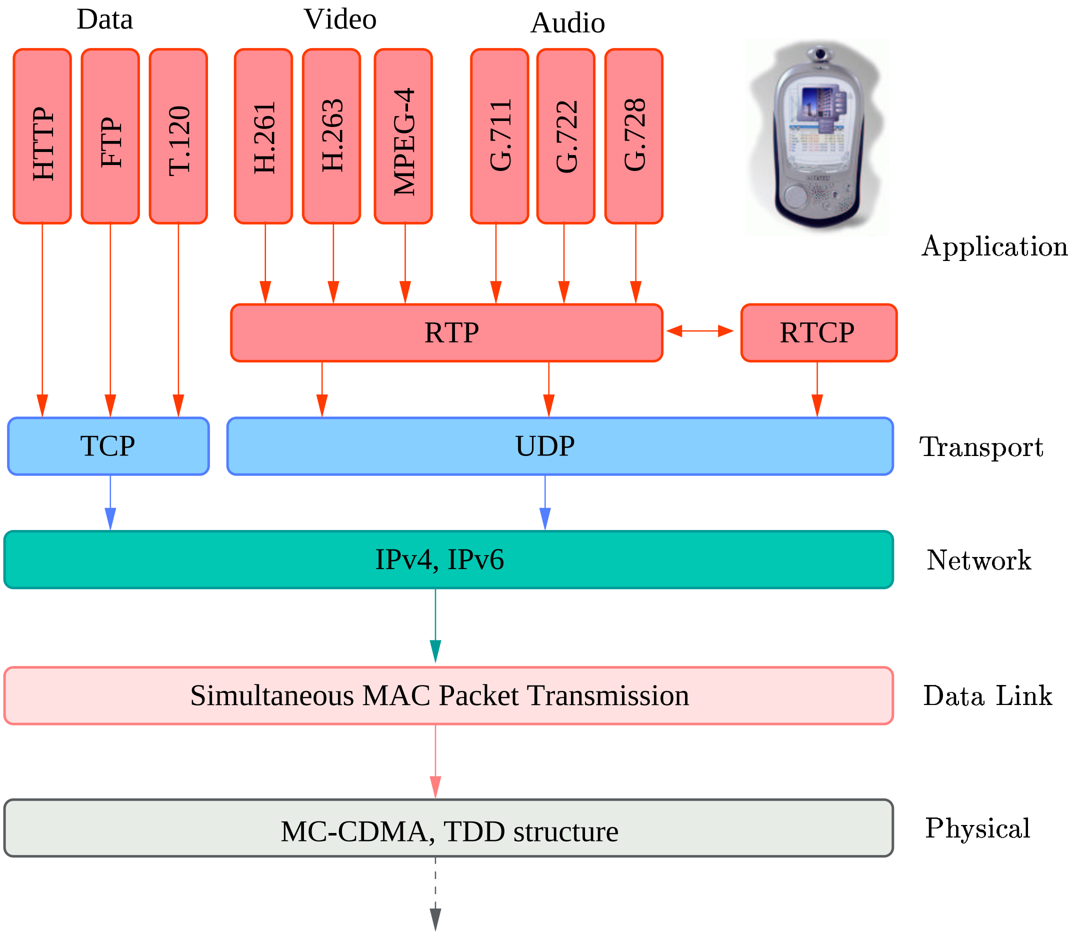


Figure 2.8: Internet protocol stack for multimedia device supporting voice, video, and data

lost segments. In case segments are lost real–time applications would result in lower but probably still acceptable quality. Retransmission of lost segments could possibly increase the segment delay of the transmitted and all stored segments. This would eventually starve the receiving application. Real time services tolerate losses, but they are highly delay sensitive. **R**eal **T**ime **P**rotocol (*RTP*) is an IP–based protocol providing support for the transport of real-time data such as video and audio streams. The services provided by RTP include time reconstruction, loss detection, security and content identification. UDP was chosen as the target transport protocol for RTP because reliability is not as important as delay constraint in packet delivery. RTP is designed to work in conjunction with the **R**eal **T**ime **C**ontrol **P**rotocol (*RTCP*) protocol to get feedback on quality of data transmission. RTCP is standardized in RFC 1889 and 1890. In an RTP session, participants periodically send RTCP packets to convey feedback on quality of data delivery and information of membership.

Both protocols, TCP and UDP were designed for wired networks. But error characteristics of the wireless channel differ significantly from that of the wired channel. The wired channel is characterized by small packet losses and errors occurring randomly. In contrast, the wireless channel is characterized by time–variable packet loss probability with bursty errors [35, 36, 37, 38, 39, 40]. Moreover the wireless channel is time–variant between the BS and WTs. The variability of the wireless channel quality is based on the mobility of each WT, effects due to fading, interference from other WTs, and shadowing. This degrades the channel performance significantly and has also a great impact on the performance of higher protocol layers. As mentioned before UDP will tolerate losses. But bursty errors, e.g. the loss of two consecutive voice packets has a great impact on the application layer [41]. Roughly speaking UDP based applications can be applied for wireless systems, but the performance degrades, even more if bursty errors occur. Since TCP has been designed for wired communication systems, where the errors are very small, TCP suffers if it is applied directly to a wireless communication systems. It is already well known that TCP is not able to support sufficient wireless Internet access [38, 36, 39]. In certain wireless channel conditions the TCP throughput decreases dramatically, even if the IP throughput is still high. This indicates that the TCP assumption of congestion is wrong. Due to wireless channel errors the retransmission timer expires. Interested readers are refered to [42, 36, 39, 38] for a detailed explanation. In [35] a performance evaluation of TCP over wireless CDMA links is presented and the performance of TCP was improved by the introduction of a simple radio link protocol that performs ARQ retransmissions.

Having explained the transport layer protocols, their requirements and their problems over the wireless link, we will now focus on the applications using the underlying transport layer. We have already claimed that customers want seamless communication services without any distinction between wire–line and wireless based systems. We have also mentioned that this, however, poses formidable challenges for communication service providers. Primarily, in wireless communication, all the services asked for by the customer have to be provided over unreliable wireless links that suffer from frequent outages. Secondly, the wide variety of communication services, that is, the applications (in network layer terminology), require vastly different **Q**uality **o**f **S**ervice (*QoS*) parameters from the underlying networking protocol layers.

Hence we want to describe the different requirements of the transport and application layer. A subset of these communication services such as *data*, *real–time*, *streaming*, and *gaming* services are depicted in Figure 2.9. Note, that even though the depicted communication services are not complete, they represent the most interesting services for wireless communication systems. Few examples are given above each service. Below them some QoS requirements are depicted for each service. The thickness of the arrows between the services and the QoS requirements represents how strong the requirements for a specific QoS are for that service.

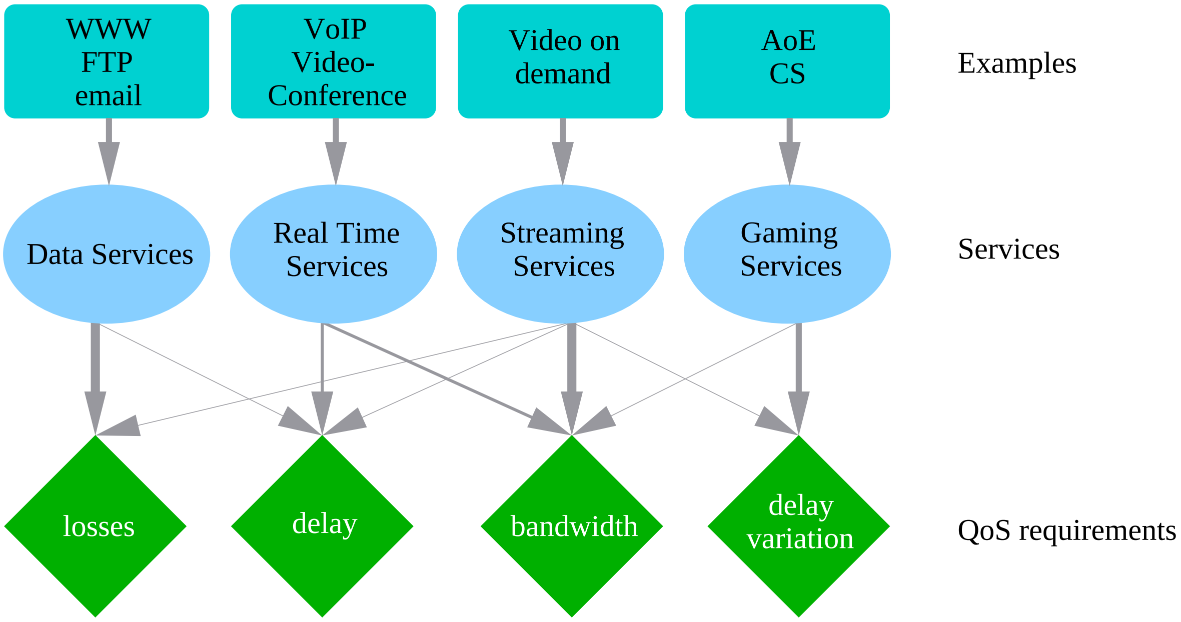


Figure 2.9: Emerging applications for wireless systems and their QoS requirements.

Data services can not tolerate any loss, are delay tolerable, and are bandwidth adaptive. Examples of *Data* applications are **F**ile **T**ransfer **P**rotocol (*FTP*) and **H**yper **T**ext **T**ransfer **P**rotocol (*HTTP*). A quickly emerging application for wireless systems is *gaming* [43]. The characteristics of gaming traffic depend on many factors, such as game design, game style (e.g., fast paced "shoot 'em ups" such as *Half–life* (Hlf) or *Counter Strike* (CS), or slow paced strategic decision games), player experience, and playing style. The few studies that have investigated gaming traffic [44, 45] indicate that games have typically moderate bandwidth requirements on the order of 10 kbit/s. The delay requirements are very tight for fast paced games (typically on the order of 100 ms). The traffic characteristics of the strategic game *Age of Empire* (AoE) are illustrated in Figure 2.11. Roughly speaking, for gaming services we need low bandwidth, but the delay constraints are very tight.

Audio and video *streaming* applications comprise the transmission of pre–recorded audio and video content upon user request. Streaming means that a stored multimedia content is delivered to a client without significant delay in starting. The streaming is played out before it has downloaded completely to the client. In streaming applications the audio or video is being played out at the user while the transmission is in progress.

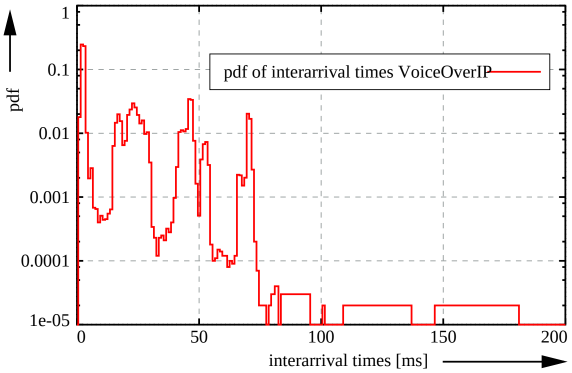


Figure 2.10: Probability density function of packet inter-arrival times for G.723.1 audio–codec measured over wireless LAN with no background traffic (see [6]).

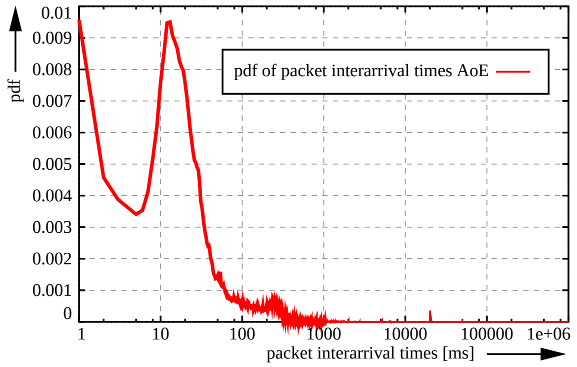


Figure 2.11: Probability density function of packet inter-arrival time for the strategic game *Age of Empire* over a typical wired LAN environment (see [6]).

(In contrast, with FTP where the entire audio or video file is downloaded before playback commences.) Moreover, it can be seen that the requirement on bandwidth is highly time variable, see [46] for more details. The MPEG–4 and H.263 video encoding standards have been specifically devised for wireless communication. These standards are expected to dominate in 3G and 4G wireless systems. Streaming services are characterized by high demand for bandwidth, while the requirements on delay and delay variation can be satisfied by a larger play–out buffer. *Real–time* audio and video applications are needed for the transmission of live events such as conferences, concerts, and sporting events. Conventional voice communication is supported by **V**oice **o**ver **IP** (*VoIP*). The traffic characteristics in terms of the inter–arrival times for VoIP traffic is illustrated in Figure 2.10. Interactive or real–time audio applications have tight delay bounds. In [47] it is shown that end–to–end delays need to be less than 250ms. In Table 2.1 we give a quantitative mapping of the communication services and their QoS requirements. In [6] this Table is given more precisely. Nevertheless, we show that there exists a wide variety of services with totally different QoS requirements in terms of losses, delay, delay–variation, and bandwidth. Moreover the QoS requirements has been identified as heterogenous and time variable. From Table 2.1 we can see that *real–time* video application have the tightest QoS requirements with high bandwidth, low losses, and small delay constraints. For *streaming* video applications very high bandwidth and nearly no losses are needed. Therefore we will focus mainly on *real–time* and *streaming* video transmission over wireless links in this work. A detailed overview for digital video encoding is given in Chapter 5 for MPEG–4 and H.263. At this point we want to describe video transmission and show the need of video based application for a stabilized wireless link characteristic.

The task of video compression is to decrease the amount of data that has to be conveyed. This is even more important for wireless links where the data rates are mostly compared to wired links. Compression is done at the sender–side by eliminating redundancy. At the receiver side the signal is reconstructed and transmission errors are corrected. Early video compression schemes worked on a picture by picture base, this means each picture

Table 2.1: Application requirements in terms of bandwidth, delay, and losses for the different categories such as data, real time, non–real time, games, and network services [6].

Type of application and example			[kbit/s]	losses[%]	delay[ms]
Data		FTP	limitless	0	TCP timer
Real Time	Audio	Voice	$\leq$64	10^{-4}	$\leq$300
		Voice over IP	10–64	$5 \cdot 10^{-2}$	$\leq$300
	Video	MPEG-4	$\leq$2000	10^{-2}	$\leq$40
		H.320	$\leq$64	10^{-4}	$\leq$40
Non Real Time	Audio	CD	150	10^{-4}	buffer length
	Video	MPEG-4	limitless	0	buffer length
Games	MP	Strategic (AoE)	20	0-10^{-2}	500ms
		Shoot+Run (CS)	20	0-10^{-4}	100ms
Network Service		NFS	limitless	0	–

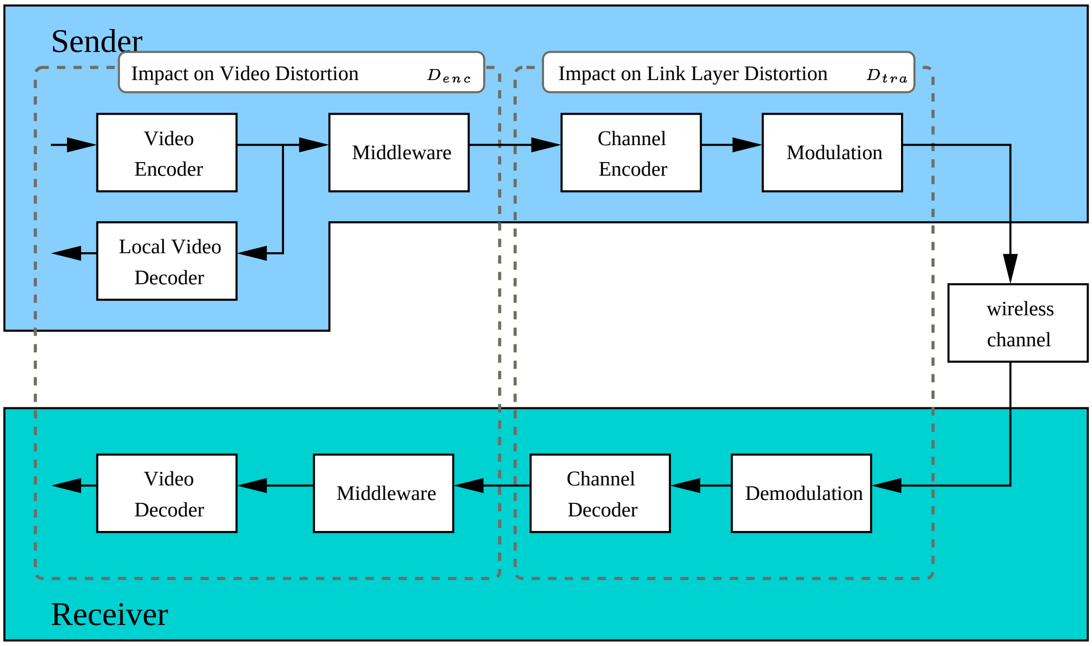


Figure 2.12: Transmission chain for hybrid coded video with overall distortion caused by D_{enc} and D_{tra}.

of the video sequence was encoded independently, one after another. For encoding, each picture was scanned into blocks. **D**iscrete **C**osine **T**ransformation (*DCT*) of each block was applied, and the values of the DCT Matrix were quantizised. The values of the matrix were read using the *zig–zag* method [48, 49] and later, encoded using a variable length encoding scheme. This method is called intra–coded video and is implemented in H.261. The compression gain depends dramatically on the number of spectral coefficients in the DCT Matrix and the granularity of the quantization.

These compression schemes lead to first compression gains, but are not sufficient for video transport over wireless or low bandwidth links. Therefore new standards like MPEG–4 and H.263/H.26L were developed and they produce smaller data rates compared to H.261 using the temporal prediction between pictures. In other words they are concentrating on the changes between two pictures hoping that the amount of information is lower. Therefore different types of frames are introduced: intra and inter coded frames. Intra coded frames contain the complete picture information, while inter coded frames contain only partial pictures information. Even though MPEG–4 and H.263/H.26L provide highly efficient video encoder/decoder with best compression efficiency, researchers of the video group of the Heinrich–Hertz–Institut [48] do not believe that this will allow video transmission with sufficient quality over wireless links for 3G. The reason for this is that the wireless environment is characterized by varying bandwidth and delay. In [48] it is claimed that the overall distortion D_{all} (resulting in a reduced video quality) of a system, transporting video over the wireless link, is divided into two parts. In Figure 2.12 we can see the transmission chain for hybrid coded video with sender and receiver divided into the encoder and the transmission entity. The distortion parts are produced independently by the encoding and the transport entity. Hereafter the distortion of the encoder entity is denoted by D_{enc}, while the distortion of the transport entity is denoted by D_{tra}. In [48] it was claimed that for good video quality at the receiver–side the overall distortion $D_{all} = D_{enc} + D_{tra}$ has to be as small as possible. Furthermore it was claimed in [48] that mechanisms were found to reduce D_{enc} significantly.

Nevertheless, the main reason that D_{tra} is so large, is the variability of the wireless link in terms of bandwidth over time. There are two solutions for this problem. Either fast adaptation on the wireless link by the sender is done or a stabilization of the wireless link must be achieved.

Adaptation on the wireless link One possibility is to adapt the payload of the video encoder output to the current channel properties. This can be achieved by simulcasted, trans–coded, or scaled bit streams. But this approach is too slow for the changes on the wireless channel, because the change of the channel conditions is in the range of 10^{-6} seconds, while the encoder needs stability over hundreds of 10^{-3} seconds. E.g. MPEG4 produces frames each 40 ms and multiple frames belong to one **G**roup **o**f **P**icture (*GoP*). For good quality the bandwidth should be stable over one GoP–time.

Stabilization of the wireless link From the point of view of the encoder the transport layer should offer stable bandwidth to ensure bounded delays for each video frame. Therefore the throughput on the wireless link has to be stabilized. As an example **F**orward **E**rror **C**orrection (*FEC*) could be used to stabilize the throughput on the wireless link. Both *heavy* and *low* channel coding results in low video quality. Heavy channel coding use up a major part of the overall bandwidth (only small bandwidth remain for the encoder), while low channel coding results in lost video frames.

The discrepancy between the video requirements and the characteristic of the wireless channel is the motivation for this work. In this work we introduce a new scheme that satisfies the requirements of real–time video transmission over wireless links by stabilizing the throughput on the wireless link.

2.3 CDMA Concept

Code **D**ivision **M**ultiple **A**ccess (CDMA) technology was chosen for the air interface of the upcoming third generation wireless systems (3G). Therefore we have chosen CDMA and a detailed survey of general CDMA benefits are provided in this section. Different CDMA systems are introduced and classified. Because **M**ulti–**C**ode CDMA (*MC–CDMA*) systems play a major role in this work, we discuss this system in more detail. For the understanding of CDMA we begin with the basic principles of spread spectrum systems.

2.3.1 Basic Principles of Spread Spectrum

Spread–spectrum techniques gained their popularity by the needs of military communications. In contrast to narrow-band communication, spread–spectrum techniques were proved to be more resistant against hostile jammers. If a communication system is considered a spread–spectrum system it has to satisfy the following criteria: (1) The bandwidth of the spread signal has to be greater than the information bandwidth. Since this criteria is satisfied also by frequency modulation, pulse code modulation, and delta modulation, there is a second condition: (2) The spread signal is composed of the information signal and the spreading sequence. The spreading sequence has to be independent from the information. In Figure 2.13 a spread spectrum transmitter and receiver are depicted. At the sender side the information signal i(t), with data rate R_i and bandwidth B_i, is spread by a spreading sequence c(t). The spreading sequence has the code symbol rate R_c, also called the chip rate. The ratio of spreading bandwidth B_s and information bandwidth B_i is denoted as the processing gain $G_{Spreading} = \frac{B_s}{B_i}$ of a spread–spectrum system.

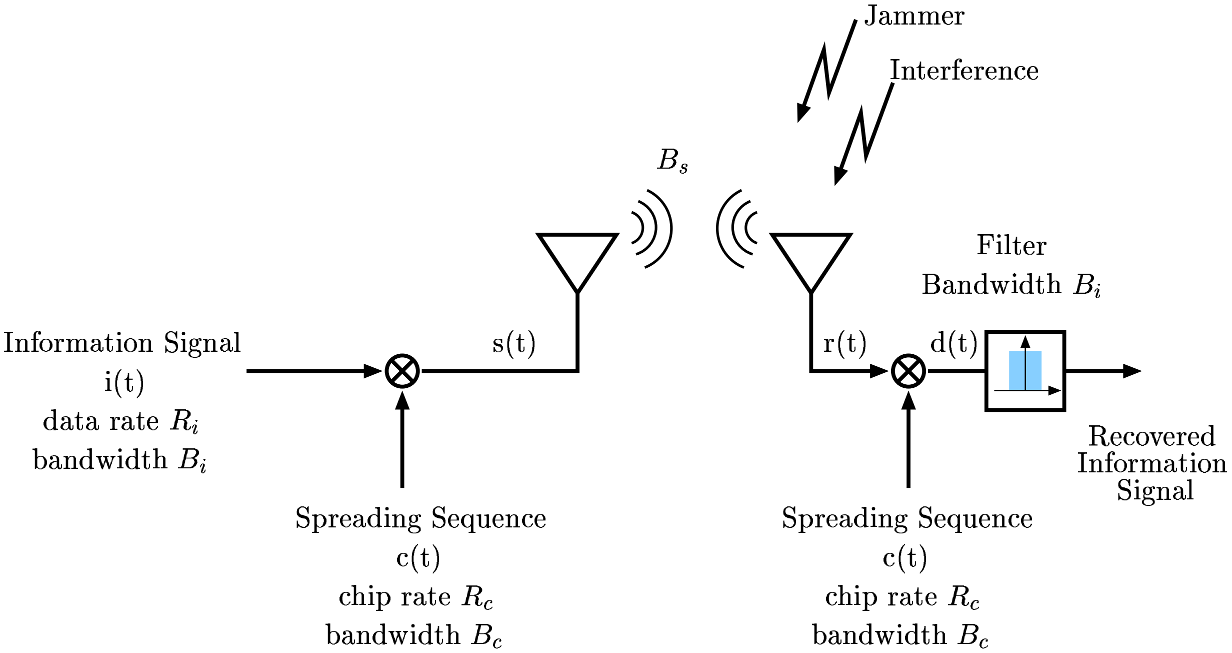


Figure 2.13: System model for spread spectrum transmission.

The processing gain does not combat white noise as it is with frequency modulation and pulse code modulation, because the spread signal is independent of the information signal. But spread–spectrum signals offer the following applications for wireless communication system: (1) Spread spectrum modulation is capable to deal with multi–path interference. Furthermore, in Section 2.3.2.1 we will see that enhanced techniques have been developed to suppress multi-path interference even further. Dependent on the spread spectrum scheme and modulation methods the multi–path interference rejection gain differs. (2) The receiver of a spread spectrum system is able to distinguish between different transmitted signals using the spreading sequence. For multiple access capability the spreading sequence design is very important. The spreading sequence is the identification for a transmitter–receiver pair. (3) Other advantages such as low probability of interception, privacy, and anti–jam capability are more relevant for military needs.

Different spreading techniques are possible. Figure 2.14 gives a classification of spread spectrum techniques. The diagram shows techniques such as **D**irect **S**equence *(DS)*, **F**requency **H**opping *(FH)*, and **T**ime **H**opping *(TH)* with their sub–classifications. All techniques can be combined to combat their disadvantages and combine their benefits. These systems are referred to hybrid spread–spectrum systems. These systems consist of combinations of two or more pure spread–spectrum systems. There exist four possible hybrid systems, namely DS/FH, DS/TH, FH/TH, and DS/FH/TH (see Figure 2.14).

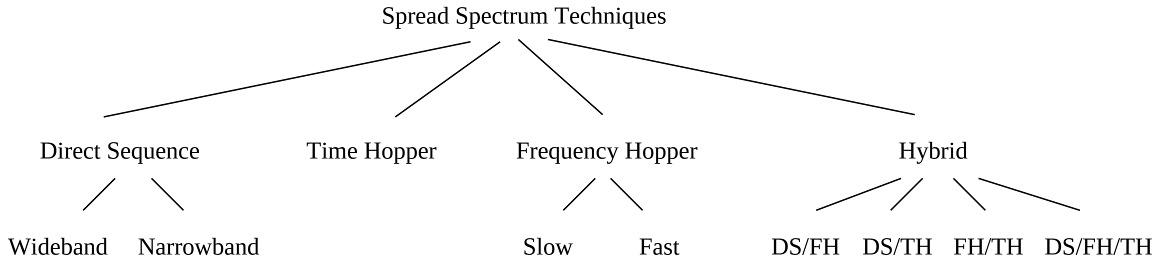


Figure 2.14: Classification of different spread spectrum system.

In the following we give a short introduction to the different spreading techniques and discuss their benefits.

2.3.1.1 Direct Sequence

In a **D**irect **S**equence **S**pread **S**pectrum *(DSSS)* transmitter the information signal is directly modulated by a spreading sequence. The spreading sequence consists of a number of spreading chips with time duration τ_{chip}. The information signal consists of a number of information bits with time duration τ_{bit}. Spreading is achieved if multiple chips represent one bit. If τ_{bit} is a multiple of τ_{chip} the processing gain G_{DS} can be easily calculated by:

$$G_{DS} = \frac{\tau_{\text{bit}}}{\tau_{\text{chip}}} \tag{2.2}$$

Figure 2.15 shows a possible DSSS transmission process. It can be seen how the original information signal $i(t)$ is spread before the transmission over the wireless link and how it will be despread at the receiver side. Furthermore the influence of the presence of jammer is also mentioned. If the data rate R_i is very small in comparison to the chip rate R_c,

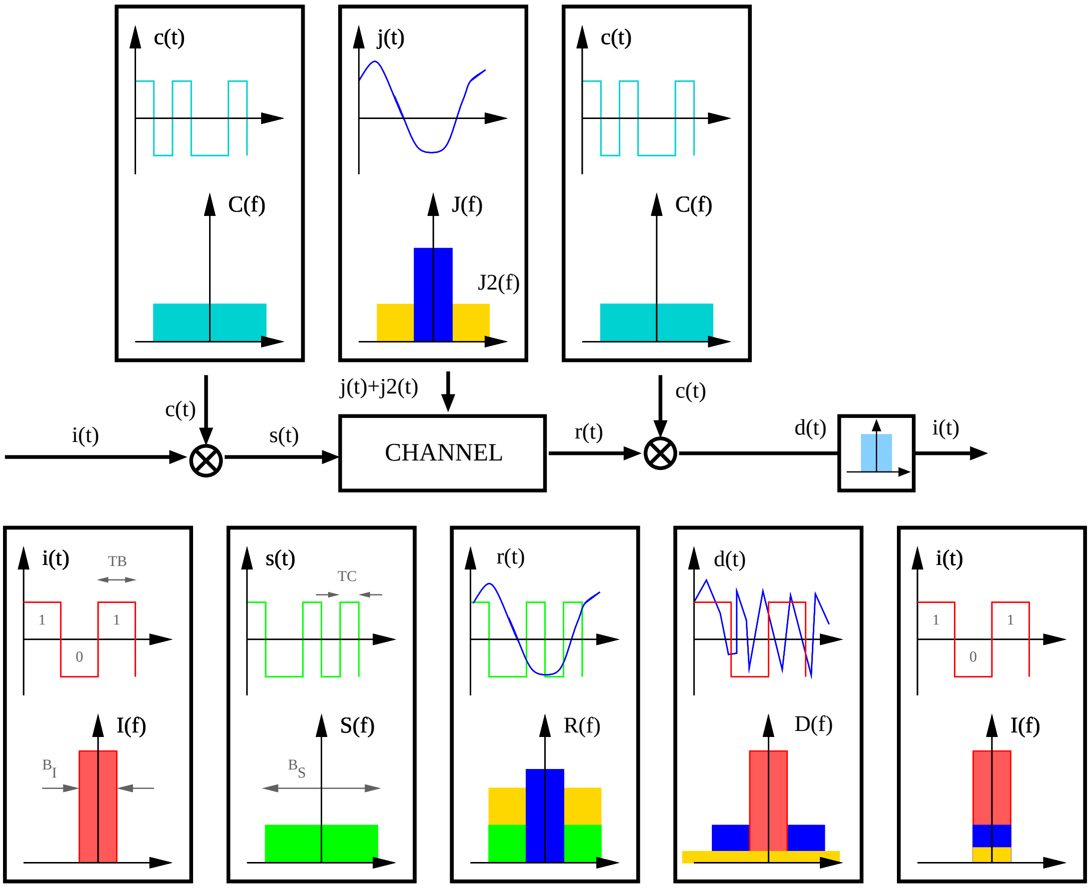


Figure 2.15: Direct sequence spread spectrum transmission process.

than the spread signal $s(t) = i(t) \cdot c(t)$ will have approximately the bandwidth of the spreading signal $(B_c \simeq B_s)$. At the receiver side the received signal $r(t)$ will be multiplied once again with the same spreading sequence $c(t)$, which results in a despreading of the original signal $i(t)$, if the autocorrelation $\varphi_{cc}^{E}(\tau)$ is nearly zero for all $\tau \neq 0$. After the de-spreading the signal will be filtered with bandwidth W_i to remove high frequencies.

The generation of DSSS signals can be achieved by a simple multiplication of information and spreading sequence. Figure 2.16 shows the generation of a DSSS signal for G_{DS}=10. A key feature of DSSS is that multiple access capability can be achieved without synchronization between different transmitters. Multi-path interference is combated if delayed signals differ by only one chip duration. In this case all delay signals are treated as interference. On the other side the transmitter receiver pair has to be fully chip–synchronized. As explained in Section 2.3.2.2 the near–far effect has to be taken under consideration, because DSSS systems are using the full bandwidth and therefore a transmitter closer to the receiver will constantly interfere and destroy signals from transmitters that are far away.

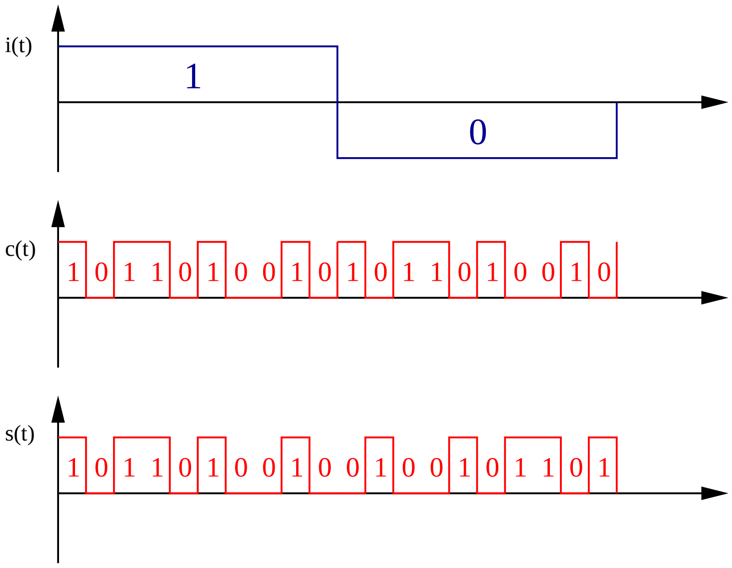


Figure 2.16: Direct sequence spread spectrum signal generation G_{DS}=10.

2.3.1.2 Frequency Hopping

Frequency **H**opping **S**pread **S**pectrum *(FHSS)* systems change the carrier frequency of the modulated information signal periodically. During a time interval τ_{hop} the frequency is constant. Afterwards both the sender and receiver *hop* to another frequency. The whole bandwidth B_s is divided into frequency slices of B_{s^*}. The set of available frequencies is called a hop–set with N frequency slices. FHSS systems differ totally from DSSS systems in terms of frequency occupancy. While DSSS systems occupy the whole bandwidth, FHSS systems use only one frequency slice at any point in time. FHSS systems are divided into *slow* and *fast* frequency hoppers as depicted in Figure 2.14. If one information bit is transmitted over several frequency slices ($\tau_{\mathrm{hop}} \leq \tau_{\mathrm{bit}}$) the frequency hopper is referred to be *fast*. Otherwise if several bits are transmitted over one frequency slice it is referred to be a *slow* hopper. The relation of hop duration τ_{hop} to information bit duration τ_{bit} depends upon the number of hops per information bits. If τ_{bit} is an integer multiple of τ_{hop}, then we denote $\tau_{\mathrm{bit}} = k \cdot \tau_{\mathrm{hop}}$. This leads to the processing gain G_{FH} (see [50]).

$$G_{FH} = \frac{B_s}{B_i} = N \cdot k \cdot B_{s^*} \cdot \tau_{hop} \tag{2.3}$$

The advantage of FHSS systems in contrast to DSSS systems is the less strict synchronization requirement. FHSS allows synchronization errors of the size of τ_{hop}, DSSS only in size of τ_{chip} [51].

In Figure 2.17 a **M**ulti **F**requency **S**hift **K**eying (*MFSK*) FHSS transmitter and receiver is depicted. The incoming data passes the MFSK modulator and then the hopping frequency (given by a **P**seudo–**N**oise (*PN*) generator) is modulated. The receiving process is given in Figure 2.18. The dashed red line represents the hopping frequency, while the relative frequency (solid blue line) gives the information. The mapping of relative

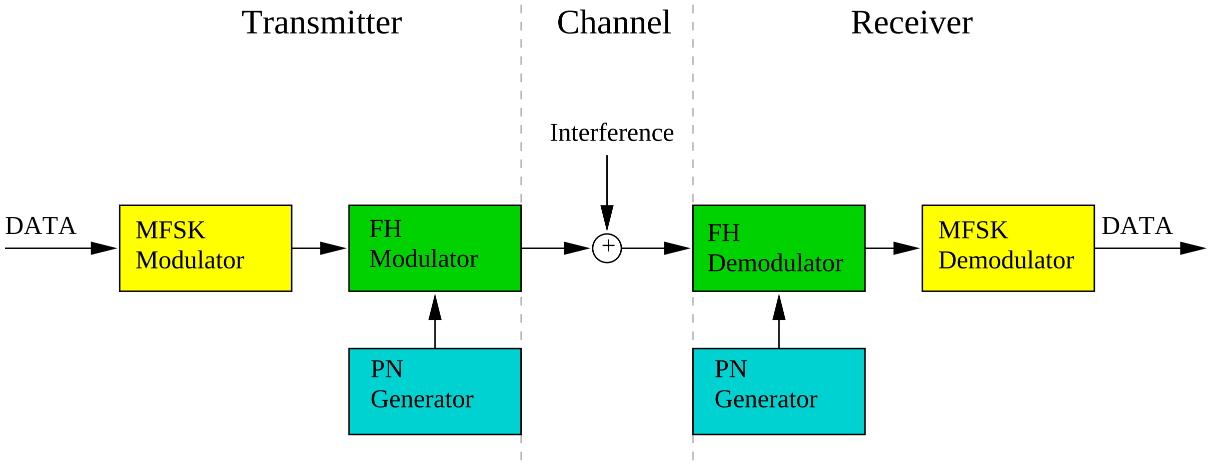


Figure 2.17: MFSK FHSS transmitter and receiver.

frequencies to bits is given on the right side of Figure 2.18.

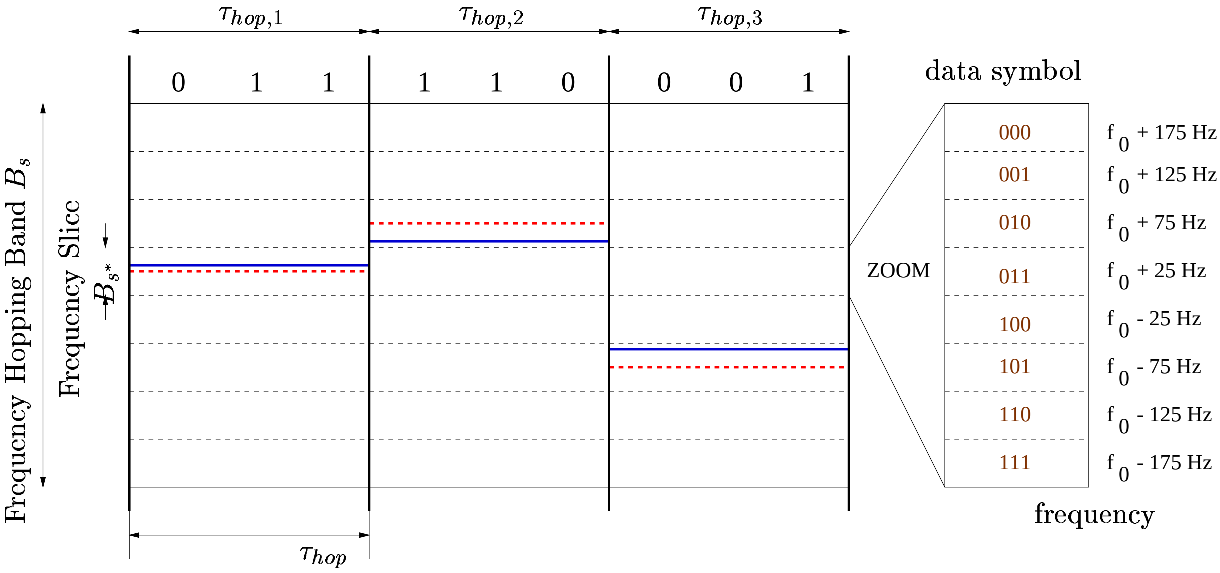


Figure 2.18: Example of MFSK FHSS receiving process.

2.3.1.3 Time Hopper

Within a **T**ime **H**opping **S**pread **S**pectrum *(THSS)* system the time axis is divided into frames of the duration τ_{frame}. As depicted in Figure 2.19 each frame is divided again into N slots with time duration τ_{slot}. A single WT will only use one slot out of k possible slots within one frame. Within this slot the WT sends with a k times higher data rate in contrast to the situation where the WT would transmit within the whole frame. Interference among simultaneous wireless terminals can be minimized if coordination between

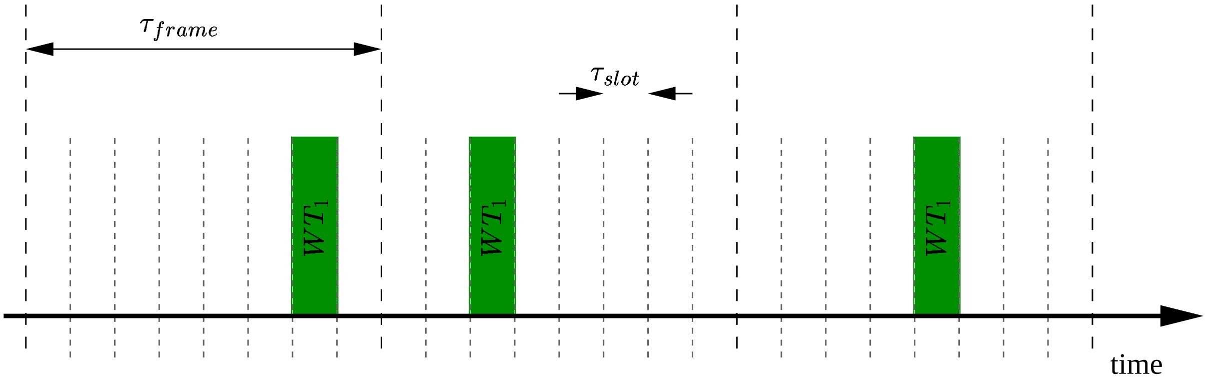


Figure 2.19: Time axis for a THSS system.

terminals can be achieved. This also avoids the near–far effect (will be introduced in Section 2.3.2.4.1). In the absence of coordination situation occur in which more than one terminal will use a time slot. The receiver will not be able to detect either of the signals correctly. Both terminals *collide* on the wireless link. For such cases error correction schemes are required. Time hopper have nearly the same acquisition time to that of discrete–sequence systems, but their implementation is much simpler than that of a frequency-hopper.

2.3.1.4 Hybrid Systems

Hybrid Spread Spectrum systems consist of combinations of two or more pure SS systems. Therefore there exist four possible hybrid systems by combining such systems, namely DS/FH, DS/TH, FH/TH, and DS/FH/TH. The combination of different SS schemes leads to increased complexity of the transmitter and receiver, but offers also a combination of their advantages. *E.g.* if DSSS and FHSS schemes are combined, the hybrid SS system offers the multi-path interference rejection of the DSSS system and the immunity of the FHSS system. In Table 2.2 the advantages and disadvantages of different SS schemes in terms of multiple access capability, multi-path interference rejection, synchronization, and hardware complexity are shortly listed.

2.3.2 Basic Principles of CDMA

For wireless communication systems the provision of a multiple access capability is indispensable and can be applied in different ways. The classical method is **F**requency **D**ivision **M**ultiple **A**ccess (FDMA). A more recent technique is **T**ime **D**ivision **M**ultiple **A**ccess (TDMA). Both techniques assign particular frequency or time slices to different wireless terminals. When all slices are occupied in the system no additional wireless terminal can be accommodated. Multiple access capability is also provided by **C**ode **D**ivision **M**ultiple **A**ccess (CDMA). The most common techniques of CDMA are frequency–hopped CDMA or direct–sequence CDMA. CDMA allows multiple users to simultaneously use a common channel for transmission of information [15]. A CDMA transmitter will *code* its infor-

Table 2.2: Comparison of different spread–spectrum techniques with their advantages and disadvantages.

SS scheme	advantage	disadvantage
direct–sequence	• best behavior for multi-path–rejection [50] • best anti-jam rejection [50, 53] • best interference rejection [50, 53] • no synchronization among terminals [51, 52] • simple implementation [51, 52] • most difficult to detect [50, 53]	• near–far problem [50, 51, 52, 14] • require coherent bandwidth [50] • long acquisition time [50, 53] • synchronization of code signal • within fraction of chip time [51, 52]
frequency–hopper	• great amount of spreading [50, 53] • no need for coherent bandwidth [50, 51, 53] • short acquisition time [50, 51, 52, 53] • inherent security [14] • less affected by the near–far effect [50, 51, 52, 53]	• complex hardware [50, 51, 52, 53] • error correction is needed [50, 53]
time–hopper	• high bandwidth efficiency [50, 53] • less complex hardware [50, 51, 52, 53] • less affected by the near–far effect [50, 51, 52, 53]	• error correction is needed [50, 51, 52, 53] • long acquisition time [50, 51, 52, 53]

mation signal with a code sequence or spreading sequence. Afterwards, the transmitter sends the coded signal to the receiver. Using the same code sequence as the transmitter, the receiver is able to decode the received signal. Also, in case the receiver receives more than one signal, it will be able to decode the information from the desired transmitter, if the code sequences satisfy cross-correlation and auto-correlation requirements (see Section 2.3.2.3). The bandwidth of the coded signal is much larger than the information bandwidth. One may say the information signal was *spread.* The coding process is therefore also called a spread spectrum modulation, while the coded signal is called a spread-spectrum signal. The spreading of the information signal gives the CDMA its multiple access capability. In Figure 2.20, 2.21, and 2.22 it is shown that subscribers can be separated in the frequency, time, or code domain.

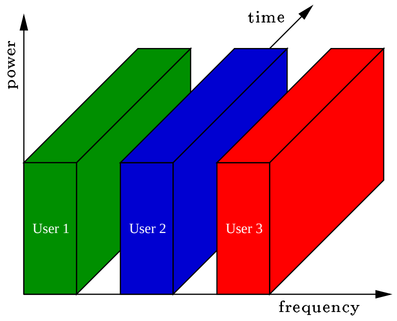


Figure 2.20: FDMA

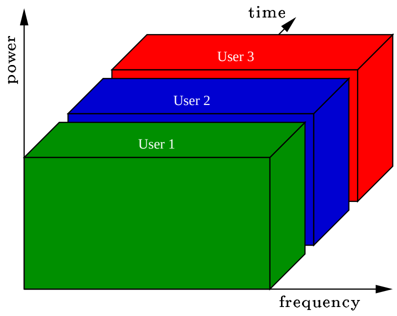


Figure 2.21: TDMA

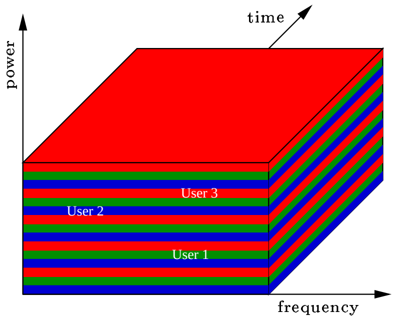


Figure 2.22: CDMA

Different multiple access techniques.

A key–feature of CDMA systems using **P**seudo–**N**oise (*PN*) sequences is that they can tolerate *overload* (these are situations where the actual **B**it **E**rror **P**robability (*BEP*) is very high for a short time), if all wireless terminals in one cell can tolerate a certain degradation in their performance. Thus CDMA systems using PN sequences do not have any sharply defined system capacity, like TDMA or FDMA systems. But for CDMA systems using PN sequences the **B**it **E**rror **P**robability (*BEP*) increases with the number of active terminals. To illuminate this property we take a frequency hopping system as an example. In Figure 2.23 the transmission process of two wireless terminals is given. Each terminal following the principal of frequency hopping *hops* to another frequency slice at the beginning of a new slot. Under certain conditions depending on the hopping strategy both terminals might use the same frequency slice. It is obvious that in the same way as the number of wireless terminals within one cell increase, the number of collisions increase as well. Note, even DS and TH spread spectrum will tolerate *overload* situations.

The following two sections provide a short introduction in further CDMA key features. Section 2.3.2.1 introduces the RAKE receiver. Section 2.3.2.2 gives a more detailed view on the power control entity. As already mentioned above the design of spreading sequences plays a major role for a CDMA system. Therefore we introduce two different families of spreading sequences in Section 2.3.2.3. Later, in Section 2.3.2.4 we classify the different CDMA systems.

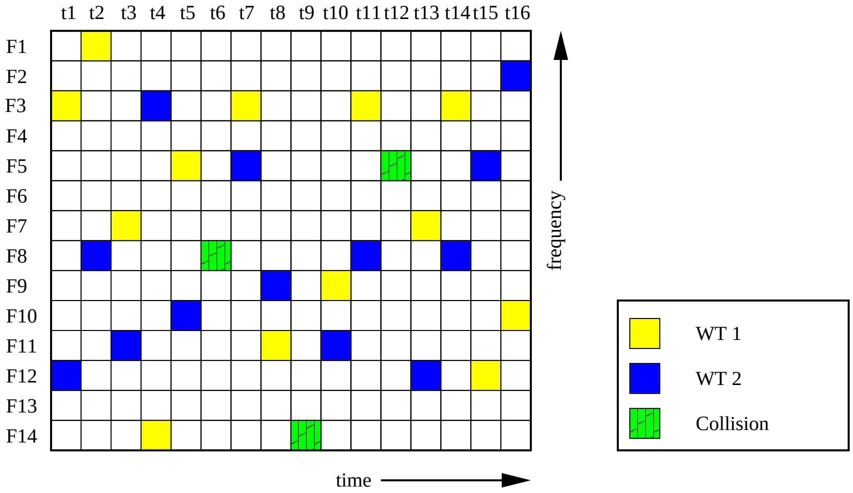


Figure 2.23: FHSS multiple access interference.

2.3.2.1 RAKE Receiver

In Section 2.1.2 we have shown that multi–path degrades the system performance. CDMA receivers resolve multi–path interference if signals arrive more than one chip apart from each other. The direct LoS–signal is the strongest one. For this strongest signal the other multi–path signals are regarded as interference and therefore they can be suppressed with the processing gain. A better receiver performance is achieved by RAKE receivers. The example depicted in Figure 2.24 illustrates the basic principle of RAKE receiver. After spreading and modulation of the information signal on the sender side the signal suffers from the multi–path channel. Different delay entities and attenuation factors in Figure 2.24 model obstacles at different distances and with different reflection properties. The multi–path signal (coming from different propagation paths) is demodulated and passed to the RAKE receiver. The ideal RAKE receiver contains a receiver *finger* for each multi–path component. In reality the number of fingers is limited by the receiver‘s hardware complexity or costs. In this case only the strongest signals with a delay of less than $\tau_{\max}$ are taken into account. In each finger the signals are despread and time aligned with one of the multi–path channels. After the despreading process the signals are weighted and combined. The weight of a signal corresponds in the best case to its attenuation factor. Roughly speaking, the RAKE receiver is an inverse of the multi–path channel. If the conditions on the multi–path channel change, the parameters of the RAKE receiver have to be adapted. In [54] the performance gain achieved with RAKE receivers in DS–CDMA systems is shown. Recently, RAKE transmitters (also called Pre–RAKE, while the RAKE receiver is also called Post–RAKE) have been proposed. If the the multi–path channel is known at the transmitter, a RAKE transmitter produces weighted signals such that there will be only one signal at the receiver. Barreto etal. [54] have shown that a substantial performance improvement can be achieved by applying Pre– and Post–RAKEs. But the trade off between channel improvements and hardware complexity has to be taken into account.

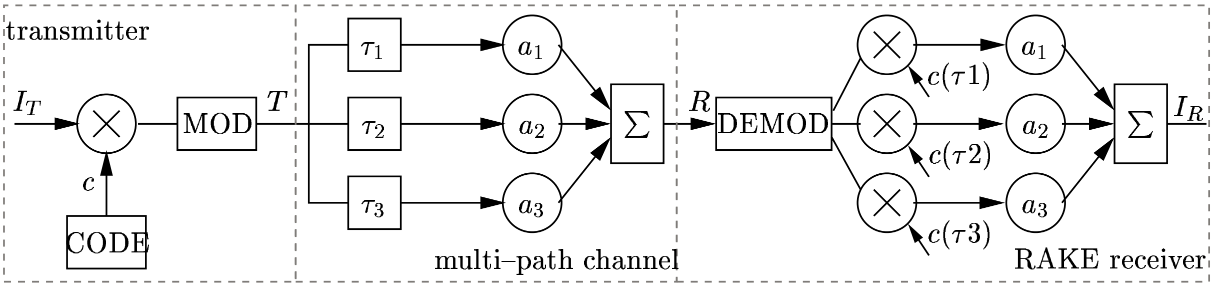


Figure 2.24: RAKE receiver structure with three fingers and multi–path channel.

2.3.2.2 Power Control Mechanism

In a DS–CDMA system all transmitters use the same bandwidth at the same time to send their information to the receiver. If we consider multiple transmitters sending to one receiver, the signals are received with different power levels, because wireless terminals have different distances to the base station (see Section 2.1.1). The signal strength can differ in the range of 100dB [55]. Due to the attenuation effect transmitters closer to the receiver have higher power levels than transmitters that are far away. This effect is called the *near–far effect*. The near–far effect plays an essential important role if multi–user interference is considered. For free space propagation, the received power falls off as the square of the distance from the wireless terminal to the base station. In addition to the near–far effect the signal strength differs dramatically due to the changing propagation conditions (moving obstacles, traffic characteristics, hand over). To overcome the changing signal strength power control entities are implemented in the transmitters. These entities are called **T**ransmitter **P**ower **C**ontrol (TPC). The TPC adjusts the transmission power P_{trx} at sender–side to ensure that all signals arrive at the receiver with the same power level P_{rcv}.

For cellular CDMA mobile communications systems the TPC plays a major role for the up–link. Within the down–link all signals sustain the same channel propagation and no near–far effect has to be taken under consideration. TPC for the down–link are only implemented to adjust a required SNIR. Furthermore, it has to be mentioned that the CDMA capacity is interference limited. Thus, the lower the power level of one terminal, the higher is the number of supported terminals. The TPC for the down–link is only implemented to adjust a required **S**ignal to **N**oise **I**nterference **R**atio (*SNIR*). In contrast to the up–link the conditions on the down–link change more slowly. Most TPCs consist of two types of power control, namely open loop and closed loop. Some wireless CDMA communication systems also have an outer loop power control, such as IS–95. The open loop power control measures the SNIR of the incoming signals and adjusts the transmission power to meet the desired SNIR. Unfortunately, the conditions for up–link and down–link can differ dramatically and therefore the open loop power control provides only a first estimation for the TPC. The closed loop power control measures the signal power at the receiver and controls the transmitter‘s power. IS–95 receivers send only one bit of information to change the transmitting power in the range of 1db/1.25ms at the transmitter side. The outer loop power control gets information about the **P**acket

Error **P**robability (*PEP*) and compares it with the required PEP. In case of deviations the transmitter power is adjusted. For cellular CDMA wireless communications systems the TPC plays a major role for the up–link.

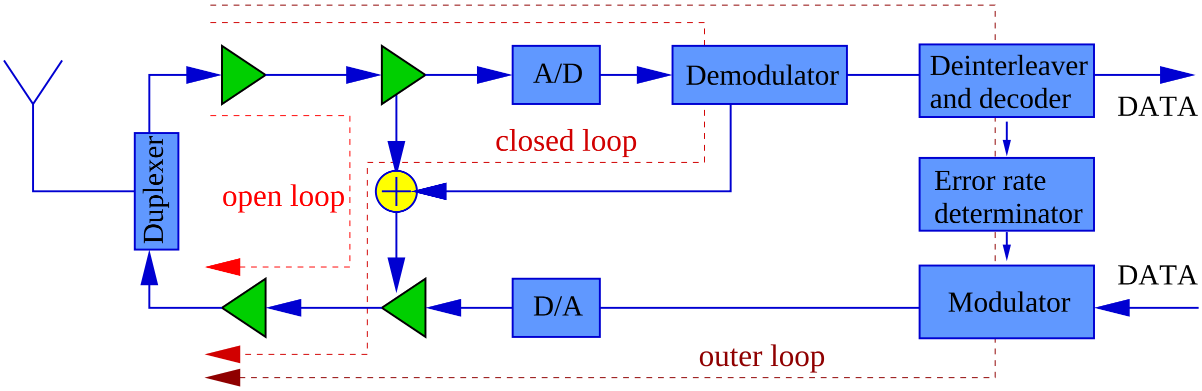


Figure 2.25: Power control loops for IS–95.

In [16] it was claimed that the **p**robability **d**ensity **f**unction (*pdf*) of the received power depends on the TPC algorithm, speed of the **A**daptive **P**ower **C**ontrol (*APC*) system, dynamical range of the transmitter, spatial distribution of the wireless terminals, and propagation statistics. In [56] it was shown that the pdf–function can be assumed to be log normal. The pdf was given as

$$f(P_{rcv}) = \frac{1}{\sqrt{2\pi}\sigma P_{rcv}} \cdot exp\left[-\frac{ln^2(P_{rcv})}{2\sigma^2}\right] \tag{2.4}$$

where σ is the imperfection in the TPC entity. In [16] it was shown that the performance degradation of a DS CDMA system with $\sigma = 1$ db was about 60%. **Therefore the system performance of a CDMA systems depends particularly on the system parameter** σ. The smaller σ the better the performance. But not only the variance of the used energy within one cell has to be taken under consideration. Also the inter–cell interference plays a major role in CDMA systems. In Figure 2.26 the interference produced/received by the center cell for/from neighboring cells is given. In [18] it is claimed that each cell in the first tier (see Figure 2.26) contributes/receives about 6% of the interference. Greater tiers do not have a great impact (second tier about 0.2% of the interference). Independent from the exact amount of interference this small example shows that the overall energy used within the cell has to be limited. Power control for voice oriented continuous traffic has dominated the work of most of the researchers for decades. But the next generation wireless networks are designed to support packet oriented data traffic in addition to the *old–fashioned* voice oriented traffic. Packet oriented traffic brings up new problems for the TPC which are quite different to voice traffic.

In [57] it was mentioned that packetized data without delay constraints can be sent in dependency of the channel state. If the wireless terminal has to invest too much power to transmit the packet successfully, it could also wait and send this packet later when channel conditions are quite good. This situation is referred to as *back off*. In the back

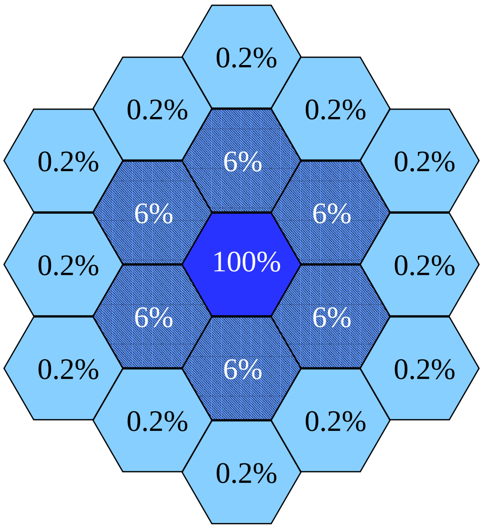


Figure 2.26: Inter–cell interference for adjacent cells.

off case the delay of the packet will increase – a situation which is not allowed for voice oriented services (only small buffering can be tolerated). In a nutshell data communication introduce a new degree of freedom; *the trade–off between power consumption and delay constraints.* In consideration of this new degree of freedom Bambos etal. [57] presented a new protocol approach for adaptive power control, which also takes delay constraints of packets into account. We will refer to this protocol in Section 2.3.2.4.1.

2.3.2.3 Spreading Sequences

The proper choice of spreading sequences enables multiple access capability for spread–spectrum based wireless communication systems. A sequences is a non-ambiguous identification for a transmitter receiver pair. Spreading sequences can be divided into *orthogonal* and pseudo–noise spreading sequences. This section gives a comprehensive overview of spreading sequences of CDMA systems. Differences between pseudo–noise and orthogonal spreading sequences in terms of auto– and cross–correlation functions are given. The properties of spreading sequences to accomplish multiple access capability is investigated. Furthermore the sets of spreading sequences are discussed for multi–code CDMA.

In a spread spectrum communication system the original information signal with bandwidth W_I is *spread* to a much larger transmission bandwidth W_{SS} using spreading sequences. At the receiver side the same replica of this spreading sequence is used to recover the original information. **In cellular radio communication multiple access capability is achieved by taking advantage of the autocorrelation and cross–correlation properties of the spreading sequences.** Such systems are refered to as code division multiple access (Therefore the spreading sequences are also called code sequences in the literature.). A spreading sequence consists of C units called *chips*. The chips are two–valued. The autocorrelation function of a spreading sequence reflects the similarity of this sequences with a replica of itself delayed by a time gap τ_{gap}. For a given time gap $\tau_{\text{gap}} = 0$ the autocorrelation value is one. In any other case $\tau_{\text{gap}} \neq 0$ the autocorrelation value should be small to minimize the interference among copies of the original signal that are generated and delayed by multi-path propagation. The cross–correlation value of two different spreading sequences represents the interference level for two signals from different wireless terminals with delay τ_{gap}. This value should be as small as possible for all τ_{gap} such that a maximum number of subscribers are allowed in the cell. As a matter of fact, the optimization of the autocorrelation and the cross–correlation can not be done separately. They depend on each other and each optimization of one side leads to a degradation of the other side.

2.3.2.3.1 Orthogonal Sequences Orthogonal sequences offer zero cross–correlation for $\tau_{\text{gap}} = 0$. If all transmitters are synchronized and no multi–path is considered, the multiple access interference can be neglected. For any other value of τ_{gap} they have large cross–correlation values. Therefore orthogonal sequences are only applied if perfect synchronism can be guaranteed within the system. The autocorrelation properties of orthogonal sequences are also poor if $\tau_{\text{gap}} \neq 0$. This happens if we consider multi–path interference. In such a situation equalization is applied to recover the original signal.

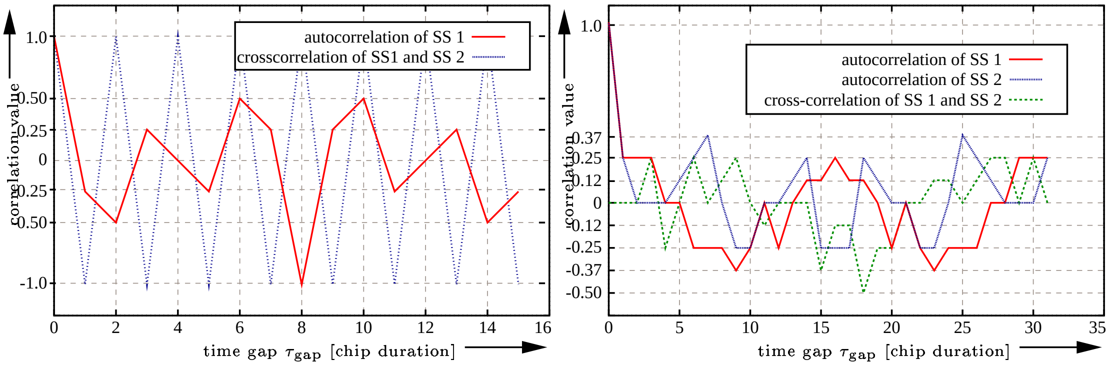
Several methods to generate orthogonal sequences are presented in [15, 14]. As representatives of the orthogonal code family *Walsh* and *orthogonal Gold* sequences are presented in this section. Walsh sequences can be easily obtained by applying the Hadamard transformation ($\otimes$ defines the Kronecker product), which is given by

$$H_0 = [1], H_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}, H_{n+1} = H_n \otimes H_n \tag{2.5}$$

Each row of the Hadamard matrix represents one valid sequence with length N. This leads to an overall number of sequences $N_S = N$. Note, the overall number for PN sequences is much larger. In general the number of achievable codes N_S is limited for orthogonal spreading sequences. Another representative of the family of orthogonal spreading sequences are the orthogonal Gold sequences. Figure 2.27 (right side) shows that orthogonal Gold sequences have reasonable cross–correlation and autocorrelation values. These values are much better than for sequences of the Walsh family.

2.3.2.3.2 Pseudo Noise Sequences PN sequences are binary sequences, which exhibit random properties similar to noise. Within the class of PN sequences, the most popular representatives are *Maximal Length* sequences, *Gold* sequences, and *Kasami* [15] sequences. All sequences can be generated using a **L**inear **F**eedback **S**hift **R**egister (*LSFR*), which is build by f feedback–taps [58]. Sequences generated with a LSFR having the maximum possible period length for an f–stage shift register are called *Maximal Length* or simply *m–sequences.* The length of an m–sequence can be proven to be $2^f - 1$. The number of possible codes depends on the number of possible sets (also called *primitive* irreducible generators) of feedback–taps. Golomb [58] showed that the overall number of sequences generated by a LSFR of degree f equals

$$N_S(f) = \frac{2^f - 1}{f} \prod_{i=1}^{k} \frac{P_i - 1}{P_i}, \tag{2.6}$$



Walsh sequences with length 16

Orthogonal Gold sequences with length 32

Figure 2.27: Auto– and cross–correlation function for Walsh and orthogonal Gold sequences.

where P_i equals the prime decomposition of $2^f - 1$. Table 2.3 provides the sequence length of a given degree f and the resulting number of achievable code sequences (see [59] for higher values of f). m–sequences have three important properties: (1) the balance property (the number of ones and the number of zeros differ by at most one), (2) the run length property (half the runs of ones and zeros have length 1, and $1/2^k$ length k for $k < n$) and (3) the shift–and–add property (combining two shifted replicas of an m–sequence yields another valid m–sequence). The periodic autocorrelation function of an m–sequence is only two–valued:

Table 2.3: Spreading sequence length with related number of maximal LSFR (degree f) spreading sequences.

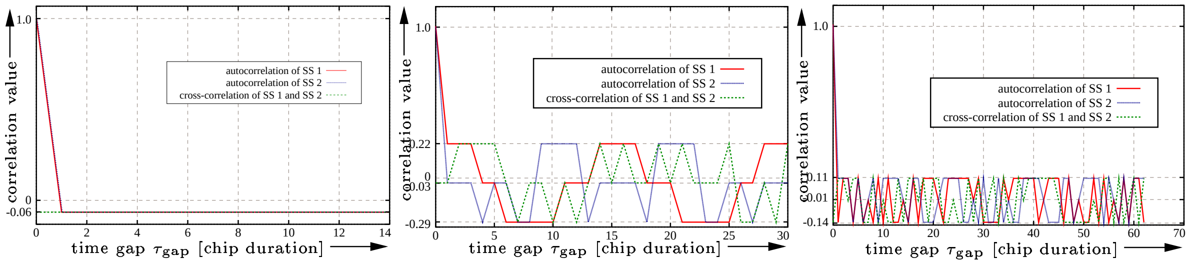
f	2	4	6	8	10	12	14	16	18	20
P_i	3	15	63	255	1023	4095	16383	65535	262143	1048575
N_S	1	2	6	16	60	144	756	2048	8064	24000

$$\Phi_{CC,mseq} := \begin{cases} 1 & k = l \cdot N \\ -\frac{1}{N} & k \neq l \cdot N \end{cases} \tag{2.7}$$

m-sequences have good autocorrelation properties, while the cross–correlation properties are very poor compared to Gold sequences. Gold sequences can be achieved by combining two *preferred* m–sequences. Different Gold sequences are achieved by combining one m–sequence with a delayed replica of another m-sequence. This gives an overall number of Gold sequences of $2^f + 1$ (considering two m–sequences and the possibilities of combination $2^f - 1$). The cross–correlation function $\Phi_{CC,gold}$ is three–valued $\{-t(f); -1; t(f) - 2\}$ with

$$t(f) := \begin{cases} (1 + 2^{\frac{f+1}{2}}) \cdot 2^{-f} & \text{if f odd} \\ (1 + 2^{\frac{f+2}{2}}) \cdot 2^{-f} & \text{if f even} \end{cases} \tag{2.8}$$

Gold [60] claimed that the cross–correlation between these sequences is uniform and bounded. Combinations of Gold sequences give *Kasami* sequences. Kasami sequences have optimal cross–correlation values achieving the Welsh lower bound [61]. Therefore Kasami sequences are a promising candidate for spreading in W–CDMA. The scaled cross–correlation function $\Phi_{CC,kasami}$ is also three–valued $\{-1; -2^{f/2} - 1; 2^{f/2} - 1\}$. The auto– and the cross–correlation function for m-sequences, Gold sequences and Kasami sequences are depicted in Figure 2.28. For these specific examples it is shown that cross-correlation for m–sequences equals (-1/16; 1), while Gold and Kasami sequences, which are three–valued, give (-9/35; -1/35; 7/35) and (-9/63; -1/63; 7/63) respectively. Moreover, the Figure gives a first quantative estimate for auto– and cross–correlation.



m–sequences with length 15 Gold sequences with length 31 Kasami sequences with length 63

Figure 2.28: Auto– and cross–correlation function for m-sequences, Gold sequences, and Kasami sequences.

Whenever synchronism can *not* be achieved PN sequences should be applied. These sequences are easy to generate and offer reasonable cross–correlation and autocorrelation values. For synchronous systems orthogonal codes should be used. In cellular wireless communication orthogonal codes should be used for the down–link and PN sequences should be used for the up–link. A special case are multi–code CDMA systems, which are explained in the following section. In this work, we concentrate on the up–link transmissions and therefore use the PN sequences. In Section 3.4.5 we give a short introduction of our work for down–link transmissions with orthogonal sequences. Interested readers may refer to [62, 7] for further information.

2.3.2.4 Classification of Enhanced CDMA System

Adaptive **P**ower **C**ontrol (*APC*) in CDMA systems is one way to obtain heterogenous QoS support. In contrast to recent power control mechanisms, which were designed to assign equal power to the wireless terminals, adaptive power control assigns power related to the QoS requirements. Another subgroup of the enhanced CDMA systems are **M**ulti–**R**ate CDMA systems. These are the only systems that are allowed to change their rates instantaneously and offer variable bit rates. The specific schemes of this subgroup are **F**ixed **S**preading **G**ain (*FSG*), **V**ariable **S**preading **G**ain (*VSG*), and **M**ulti–**C**ode CDMA (*MC–CDMA*). In contrast to Single–Code CDMA systems Multi–Code systems support multiple times the basic bit rate. A more detailed description is given in Section 2.3.2.4.4.

2.3.2.4.1 Adaptive Power Control As outlined above the power control is designed to overcome the near–far effect and tries to control the received power from the WTs. Another goal of the power control is to increase the system capacity. Under the assumption that all wireless terminals require the same QoS, the power for each of the terminals is controlled such that sender's signals are received at the base station with the same power. This is the power control algorithm for voice oriented services.

To support time–variable QoS the power level of each terminal was identified in [63, 64, 65, 66] as a controllable parameter. The proposed approach dynamically adapts the QoS requirements with power control schemes. As an example we assume k wireless terminals. Suppose that terminal $i = 1$ has the highest QoS requirements QoS_1 with power level P_1. All other terminals adjust their power level P_i according to the TPC‘s commands. The power level for WT i is given by

$$P_i = P_1 \cdot \frac{QoS_i}{QoS_1}. \tag{2.9}$$

During the transmission process, the power is adjusted depending on the channel state to achieve a required SNIR, e.g. if the channel becomes worse (higher interference) the power level is increased to stabilize the SNIR. The advantage of such a system is that it can be easily integrated into existing CDMA systems. Adaptive power control is a strong candidate for time–variable QoS support. In contrast to the VSG and MC approaches adaptive power control support does not include variable bandwidth.

2.3.2.4.2 Fixed Spreading Gain In the **F**ixed **S**preading **G**ain (*FSG*) approach [67], the spreading gain of each bit stream is maintained constant. The chip duration is also maintained constant. The users vary the transmission time to adapt the variable bit rate requirements. The low rate users transmit for a shorter time as opposed to the higher rate users. It can be seen that the lower rate users suffer from more multi–access interference from the higher rate users, as the higher rate users transmit for longer time. However, the higher rate users suffer from multi–access interference for a shorter time. Hence, the performance for higher rate users is better than for lower rate users. To compensate for this degradation in the performance, the lower rate users transmit with more power to increase their signal to interference ratio.

2.3.2.4.3 Variable spreading gain The **V**ariable **S**preading **G**ain (*VSG*) approach [68] offers the possibility to achieve flexible data rates. Under the assumption that the bandwidth for transmission is fixed, the spreading gain is reduced as the information bit rate increases, and vice versa. Changing the spreading gain has an impact on the energy per bit ratio. If the transmission power is fixed an increased data rate leads to a smaller energy per bit ratio (leading to a higher bit error probability). Therefore, the VSG approach is coupled with a variable power control entity to satisfy the same SNIR for wireless terminals with different bit rates. In case the transmission power is adapted to the transmission rate, high data rate terminals will influence low data rate terminals. Note, that different terminals with the same SNIR requirement can be accommodated by allocating power proportional to the terminals‘ data rates.

$$P_i = P_1 \cdot \frac{R_i}{R_1} \tag{2.10}$$

Whenever the transmitter changes the spreading gain, the receiver has to be informed about this change. This results in an increased signaling overhead. Since the power level varies according to the chosen data rate the TPC will also add some more signaling overhead. In the case of short spreading sequences, VSG–CDMA systems suffer from multi–path interference, because of **I**nter **S**ymbol **I**nterference (*ISI*). If we consider a constant delay spread for the wireless link, the same number of chips are influenced, but the number of influenced symbols depends only on the spreading gain. In [52] it is claimed that with VSG the high power terminals degrade the performance of low power terminals. In addition, high power terminals are using smaller spreading codes and therefore the spreading sequences have a significant impact on the system performance. Thus it might be useful to implement higher bit rates with Multi–Code CDMA systems [52].

2.3.2.4.4 Multi–Code Code Division Multiple Access In this section the capability of WTs to transmit and receive on multiple CDMA channels by the means of MC–CDMA is discussed. We discuss different families of spreading sequences with their need for signaling for MC–CDMA. We then compare the MC–CDMA approach with the enhanced power control and the variable spreading gain approach. In Figure 2.29 the statistical multiplex effect of the air interface is depicted. For illustration purposes we choose three WTs which transmit packets over the wireless link to the BS. For this example we assume that WTs are able to transmit also on multiple channels. This example shows how the three different *streams* of each WT with highly variable load are multiplexed on the air interface.

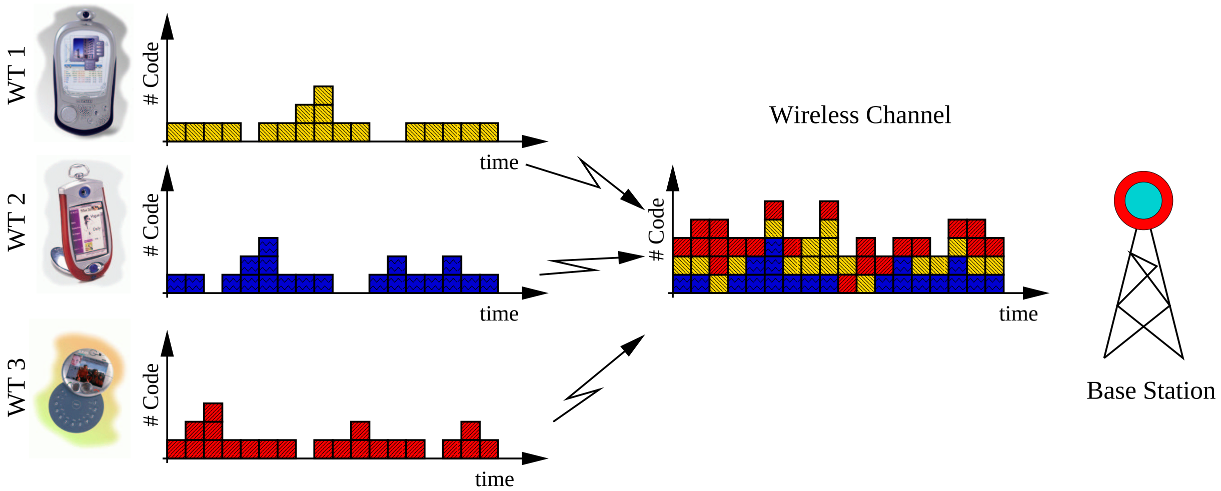


Figure 2.29: Statistical multiplex effect on the air interface of CDMA.

MC–CDMA [69] can support a basic bit rate as well as integer multiples of the basic bit rate. In MC–CDMA a high data rate is split into smaller data rates. Each small data rate is then spread by a different code sequence over the entire coherent bandwidth. All spread signals are modulated and transmitted over the wireless link. Figure 2.30 depicts a Multi-Code CDMA transmitter. It shows that a high data rate is split into smaller data rates. Afterwards each small data rate is spread by a code sequence over the whole coherent bandwidth. All spread signals are modulated and transmitted over the wireless link.

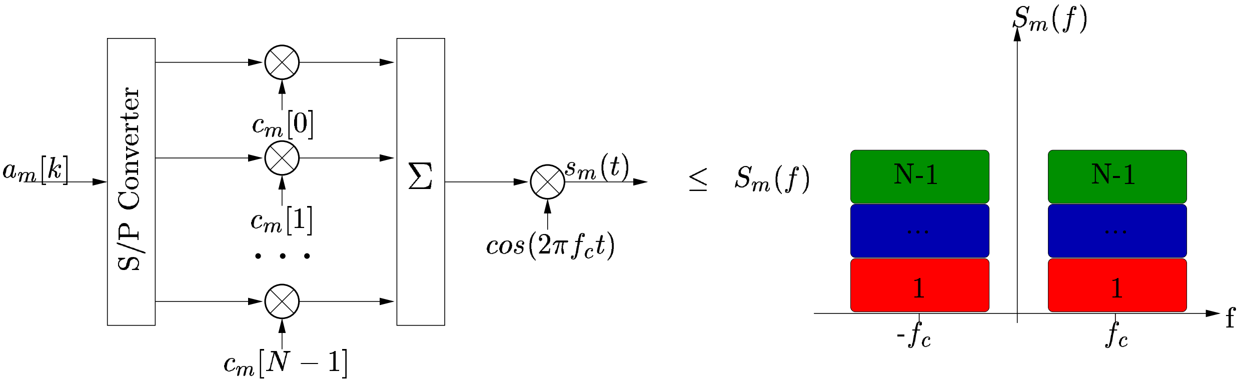


Figure 2.30: Multi–Code CDMA sender.

The choice of spreading codes is very important to decrease self–interference. Since orthogonal sequences are used the number of overall sequences within the cell is limited. In case one terminal wants to transmit on more channels it has to request these channels from a centralized code depository. Thus delay–sensitive services are hard to support. Orthogonal code sequences do benefit from the fact that self interference does not exist as long as the system is synchronous. Synchronism exists only within the down–link and between channels of one wireless terminal in the up–link. In asynchronous systems orthogonal codes lead to poor system performance.

The other choice of codes are PN code sequences. The number of these codes is large, however, the interference among the channels is high. Lin and Gitlin [69] proposed to use pseudo random sequences to distinguish between the asynchronous up–link transmissions of the different wireless terminals. Additionally, they proposed to use orthogonal sequences to distinguish between synchronous up–link transmissions of a given wireless terminal. This approach gives a large total number of codes in conjunction with low self–interference between the parallel channels of a given wireless terminal. Further improvements are achieved when a central entity assigns the terminal specific pseudo random sequences [70]. Note that this does not require additional overhead for bit rate changes because the total number of codes is large and the codes are assigned to the terminals during link establishment. The hardware complexity, however, is higher as RAKE receivers are required for each channel to suppress multi–path interference. It is noted in [52] that the MC–CDMA approach is slightly more promising for multi–rate CDMA than the Variable Spreading Gain approach. It is argued that the former has a smaller signaling overhead and lower multi–path interference.

2.3.2.5 Comparison of APC, VSG, and MC–CDMA

A brief overview of benefits and drawbacks of the three approaches for supporting time–variable QoS is given in Table 2.4. Note that combinations are possible to combine their benefits. In [71] the performance in terms of SNIR and BEP of VSG and the MC–CDMA systems is compared. The authors in [71] summarize that whether a wireless terminal uses one of theses approaches, it has an identical effect on both the SNIR and the BEP of the other wireless terminals in the system. Furthermore the authors remark that multi–code systems have the disadvantage of instantaneous amplitude variations (for VSG they exhibit a constant instantaneous amplitude) and additional hardware complexity. In [72] methods are introduced which reduce the hardware complexity of MC–CDMA systems significantly. Nevertheless in [52] it is claimed, that multi–rate can be implemented with MC–CDMA in a slightly better way than with VSG, because of higher signaling overhead and lower multi–path interference rejection of the later. Adaptive power control approaches have no multi–rate capability. For all approaches a high performance power control entity has to be implemented. Henceforth we use the MC–CDMA approach.

Table 2.4: Enabling CDMA technologies with their benefits and drawbacks for time–variable QoS support.

	Enhanced Power Control	Variable Spreading Gain	Multi–Code CDMA
multi–path interference	no further impact	high rate→ high ISI	no further impact
multiple access interference	→ power level	→ bit rate	→ number of channels
hardware complexity	TPC already exist	further oscillator	RAKE receiver for each channel
	high performance power control		
signaling overhead	TPC message	TPC message change in SG	not necessarily only for WBE sequences (Section 2.3.3.1)
enable higher data rates	no	yes	yes
granularity	—	high degree depends on spreading codes	multiple of CDMA channels
applied in	—	UMTS FDD [73]	UMTS TDD [73]

2.3.3 Wireless CDMA Channels

In the following we investigate the characteristics of the CDMA uplink wireless channel. We concentrate on the usage of **P**seudo–**N**oise (*PN*) sequences. In particular the CDMA based bit error probability for non orthogonal spreading sequences depends on the overall number of active channels in one cell. Therefore we give a detailed overview of the major bit error probability models and the impact on throughput and capacity.

2.3.3.1 Bit Error Probability in CDMA

Direct **S**equence *(DS)* CDMA communications systems achieve their multiple access capability by assigning each WT a unique spreading sequence (see Section 2.3.2.3). Because of different distances between WT and BS the transmitted signals arrive at the BS with random delays τ, carrier phases ϕ and power levels P. The *exact* error probability depends on the particular spreading sequences used by the WT, random amplitude, delay and carrier phase. The calculation of the exact **B**it **E**rror **P**robability (*BEP*) for a CDMA system with **P**seudo–**N**oise (*PN*) sequences is difficult to evaluate. Therefore a variety of different error models based on Gaussian approximations can be found in [14]. The design of the spreading sequences is not taken under consideration with the Gaussian approximations.

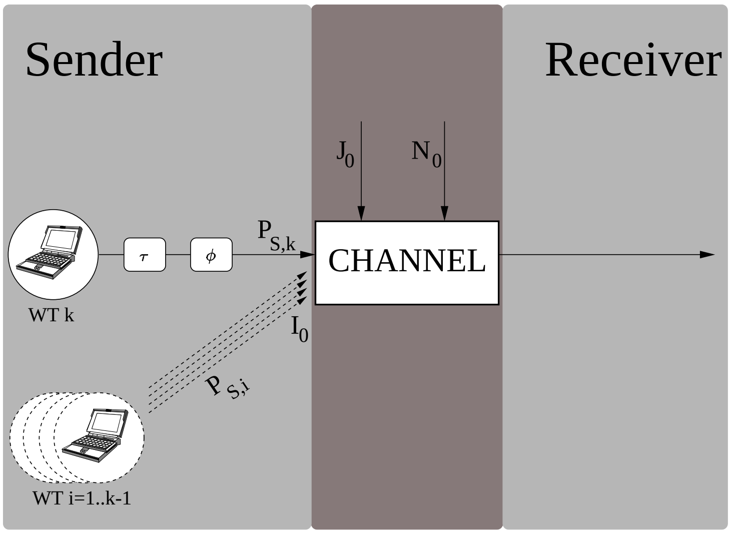


Figure 2.31: General CDMA Channel.

As given in Figure 2.31 we suppose that J WTs simultaneously access the wireless channel. We further assume that each bit is spread by $G_{Spreading}$ chips. The transmitted signal $s(k)$ is assumed to be a sequence of two independent, identically distributed random variables (chips), where the probability of each variable is given by 1/2. Therefore we use the classical expression for uncoded BPSK modulation considering additive white Gaussian noise with power spectral density I_0 and energy per bit E_b (given in Equation 2.11) to calculate the bit error probability of a DS CDMA system. Assuming that all WTs arrive chip synchronously but phase asynchronously at the receiver side the bit error probability $p_{biterror}^{BPSK}$ results in

$$p_{biterror}^{BPSK} = Q\left(\sqrt{2 \cdot SNIR}\right) = Q\left(\sqrt{\frac{2 \cdot E_b}{I_0}}\right). \tag{2.11}$$

where $Q(x)$ is related to the complementary error function and is defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-t^2/2} dt. \tag{2.12}$$

The energy per bit E_b can be calculated by dividing the received signal power P_S by the data rate R_i.

$$E_b = \frac{P_S}{R_i}. \tag{2.13}$$

To calculate the power spectral density I_0 we assume perfect power control for all WTs. Perfect power control implies that all signals arrive at the receiver with the same power level. Moreover we are not considering multi–path interference in our model. Thus the noise power spectral density I_0 is influenced by three different terms as depicted in Figure 2.31. The additive white Gaussian noise power spectral density N_0 is characterized by the communication system, while the jammer density Y_0 depends only on the jammer‘s hardware and his jamming technique (broadband or band limited). For simplicity we assume only broadband jamming. The last factor are other active WTs which are using the channel simultaneously. The sum over the energy of all other WTs will be spread over the whole spreading bandwidth W_{SS}. The inter-chip interference was not taken into account within our assumptions. Because of the perfect power control assumption each signal will arrive with the same power level, which results in a very simply equation for the noise power spectral density I_0 (see Equation 2.14)

$$I_0 = \frac{I}{W_{SS}} = \frac{P_S \cdot (k-1)}{W_{SS}} + N_0 + Y_0, \tag{2.14}$$

where I is the overall noise power by all other WTs and k denotes the number of active channels within the cell. According to a **S**ingle–**C**ode CDMA (*SC–CDMA*) system we assume one channel per WT and if α_i represents the activity of WT i, k is generally given by $k = \sum_{j=1}^{J} \alpha_j$. Using Equation 2.11, 2.13 and 2.14 we obtain:

$$p_{biterror} = Q\left(\sqrt{\frac{2 \cdot \frac{P_S}{R}}{\frac{P_S \cdot (k-1)}{W_{SS}} + N_0 + Y_0}}\right). \tag{2.15}$$

$$G_{Spreading} = \frac{T_B}{T_C} = \frac{\frac{1}{T_C}}{\frac{1}{T_B}} = \frac{W_{SS}}{R_i}. \tag{2.16}$$

The spreading gain has been defined as the number of chips per bit for a DS CDMA system. Thus it can be calculated by the ratio of T_B and T_C. This is equivalent to the ratio of W_{SS} and R_i.

$$p_{biterror} = Q\left(\sqrt{\frac{2 \cdot G_{Spreading}}{k - 1 + \frac{N_0 \cdot W_{SS}}{P_S} + \frac{Y_0 \cdot W_{SS}}{P_S}}}\right). \tag{2.17}$$

The complementary error function Q function can be transformed in the $erfc$ function

$$erfc(\alpha) = 2 \cdot Q\left(\sqrt{2} \cdot \alpha\right). \tag{2.18}$$

We model both the background noise and the jamming (in a friendly environment the jammers can also be neglected) as b wireless terminals that transmit all the time, resulting in a decreased SNIR. This simplification leads to the simplified Equation 2.20, which can be found in [74]

$$p_{biterror}(k) = Q\left(\sqrt{\frac{2 \cdot G_{Spreading}}{k - 1 + b}}\right). \tag{2.19}$$

Hereafter, we use the following notation for $p^{+}_{biterror}$ to distinguish different CDMA systems: the superscript + indicates if the system assumptions are based on synchronous or asynchronous phase and chips as given in Figure 2.20 (e.g. PACS signifies asynchronous phases (PA) and synchronous bits (CS).).

$$p^{PACS}_{biterror}(k) = Q\left(\sqrt{\frac{2 \cdot G_{Spreading}}{k - 1 + b}}\right); k - 1 + b > 0 \tag{2.20}$$

As mentioned before the bit error probability depends also on the delay and carrier phases. In [14, 75] we can find further bit error probability approximations. Equation 2.21 gives the bit error probability for chip and phase asynchronous CDMA systems, while Equation 2.22 reflects phase and chip synchronous CDMA systems. Equation 2.23 reflects phase synchronous and chip asynchronous CDMA systems.

$$p^{PACA}_{biterror}(k) = Q\left(\sqrt{\frac{3 \cdot G_{Spreading}}{k - 1 + b}}\right); k - 1 + b > 0. \tag{2.21}$$

$$p^{PSCS}_{biterror}(k) = Q\left(\sqrt{\frac{G_{Spreading}}{k - 1 + b}}\right); k - 1 + b > 0 \tag{2.22}$$

$$p^{PSCA}_{biterror}(k) = Q\left(\sqrt{\frac{1.5 \cdot G_{Spreading}}{k - 1 + b}}\right); k - 1 + b > 0 \tag{2.23}$$

An improved Gaussian approximation has been derived by Holtzman [76] and applied in [77]. The bit error probability for the improved Gaussian approximation is given in Equation 2.24 and 2.25. The calculations for the bit error probability are still simple enough but lead to quite accurate results. In [14] it is claimed that the improved Gaussian approximation should be used, if the number of WT is small or the spreading gain $G_{Spreading}$ is large.

$$p^{IMP}_{biterror}(k) = \quad \frac{2}{3}Q\left(\sqrt{\frac{3 \cdot G_{Spreading}}{k-1+b}}\right)$$

$$+\tfrac{1}{6}Q\left(\frac{G_{Spreading}}{\sqrt{\frac{(k-1+b)N}{3}+\sqrt{3}\sigma}}\right)$$

$$+\tfrac{1}{6}Q\left(\frac{G_{Spreading}}{\sqrt{\frac{(k-1+b)N}{3}-\sqrt{3}\sigma}}\right) \tag{2.24}$$

$$\sigma^2 = (k-1+b)\left(\frac{23}{360}\cdot G^2_{Spreading} + \left(\frac{1}{20}+\frac{k-2+b}{36}\right)(G_{Spreading}-1)\right). \tag{2.25}$$

Until now we have focused on SC–CDMA systems. But the bit error probabilities given in the equation above can be used even for the MC–CDMA approach. As mentioned before the MC–CDMA performance depends significantly on the influence of parallel channels. Therefore the motivation for the present section is to show the impact of three different spreading sequence allocations schemes for multi–code in terms of the **S**ignal to **N**oise **I**nterference **R**atio (*SNIR*). We consider J WT with an asynchronous MC–CDMA channel with the **B**inary **P**hase **S**hift **K**eying (*BPSK*) modulation scheme, in which each WT, say WT j, is assigned a certain number $1 \leq R_j \leq R_{\max}$ of spreading sequences. Spreading sequences used by user j span an N-dimensional space $\mathbf{U}_j$ that is a subspace of a M-dimensional space $\mathbf{M}$ spanned by all $k = \sum_{i=1}^{J} R_i \leq J \cdot R_{\max} = S_{\max}$ spreading sequences. Figure 2.32, 2.33, and 2.34 show three different allocation schemes with their M-dimensional spreading sequence space. In the following section we provide a description of the different spreading design rules for MC–CDMA.

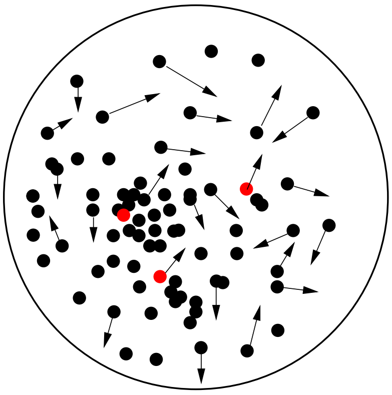


Figure 2.32: Random sequences

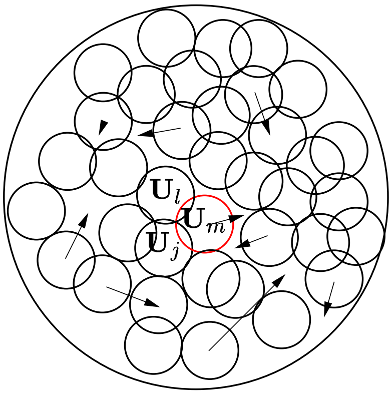


Figure 2.33: Random subspaces

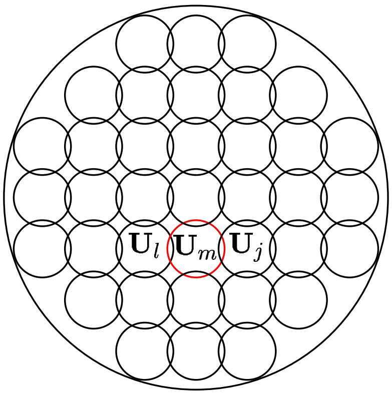


Figure 2.34: Controlled subspaces

Different approaches for allocation of spreading sequences.

2.3.3.1.1 Random Sequences The first approach called *random sequences* or PN sequences chooses the spreading sequences randomly i.e. instead of carefully designing a set of spreading sequences, the sequences are chosen randomly from the set of all possible sequences. Sequences are changed from bit to bit[2] so the vectors associated with

[2] Such as in IS–95 between sender and receiver sequences are generated out of one long and well known PN sequence.

each users *jump* in the space **M** which is illustrated in the Figure 2.32. We assume the the spreading sequences are chosen randomly by a centralized entity (e.g. base station) and signaled at the beginning of each communication session to the WTs. This procedure takes care that sender and receiver use the same spreading sequence and that no spreading sequence is used twice (we refer to Section 4.5 for details). Figure 2.35 gives one possible implementation of a MC–CDMA sender. Multiple channels d_1 up to d_R are spreading their information with different PN sequences. The PN sequences are created by the `Generator` entity. There exists only one `Oscillator` entity for all channels. After the spreading process all signals are multiplexed to signal s and send over the wireless link.

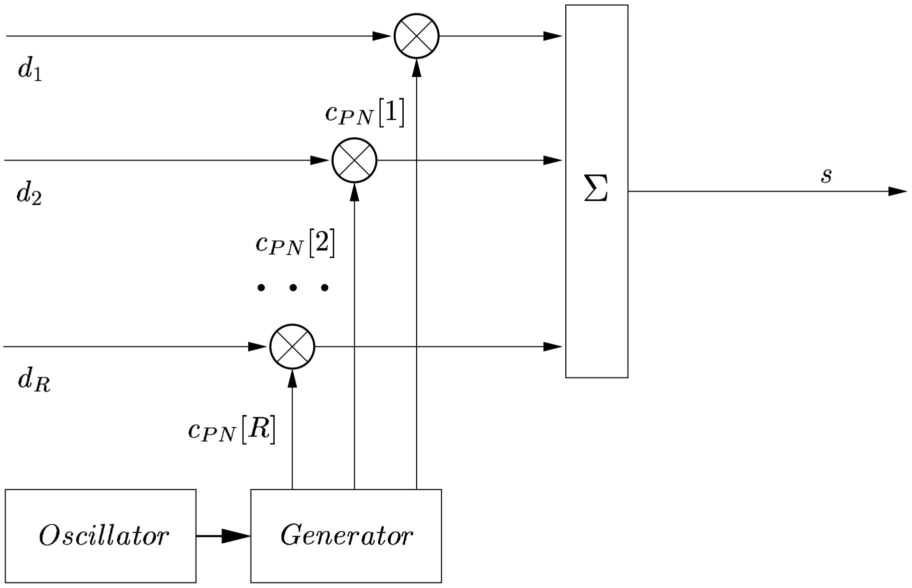


Figure 2.35: Multi–code CDMA sender for randomly chosen spreading codes.

The advantage of random sequences is that there is no need for planning, design and/or adaptive allocation of spreading sequences. This feature avoids overwhelming signaling. In case the *random sequences* approach is used all active channels have an influence on each other. As an example if all WT are active and the WTs are using all their channels, k (as the parameter to calculate the BEP in Equation 2.20, 2.22, 2.23, 2.21, and 2.24) is given by $J \cdot R_{max}$. In case R_g channels are active at WT g the number of active channels seen by WT g within the single cell for each channel is given by

$$k = \sum_{j=1}^{J} R_j. \tag{2.26}$$

This k value can be used to calculate the BEP.

2.3.3.1.2 Random Subspaces

The next approach is called *random subspaces.* The method of allocating spreading sequences presented here reduces the interference in some manner by exploiting the information about the synchronous transmission channels. The idea is to make each user span a subspace that is spanned by orthogonal sequences used on the synchronous channels. Thus, the interference term V^s_{j,R_j} is equal to zero. In case *random subspaces* are used the interference of parallel used channels of the same WT can be neglected and therefore k is given by

$$k = \sum_{j=1;i\neq g}^{J} R_i + 1. \tag{2.27}$$

Note that spreading sequences still have to change from symbol to symbol randomly due to the existence of other asynchronous users causing the subspaces to *jump* in the space **M** which is illustrated in the Figure 2.33. For illustration purpose we assume that each WT has the same number of maximal parallel channels R_{max} (all subspace have the same size). While the dimension of the subspace is determined by the orthogonal spreading–sequences, the *hop* process is determined by the PN–sequence. Such random subspaces can be easily generated by means of the Hadamard matrix (see Section 2.3.2.3.1). Hence, all we need to do is to generate a random spreading sequence and multiply columns of the Hadamard matrix by the elements of the spreading sequence so that the first row becomes the generated random sequence. This approach shows how to use this additional information in a simple way in order to improve the system performance. Since the subspaces are generated without considering the other spanned subspaces by other users, they can overlap in the sense that they interfere. By overlapping, we mean the interference measured in terms of SNIR such that stronger overlapping leads to a smaller SNIR values.

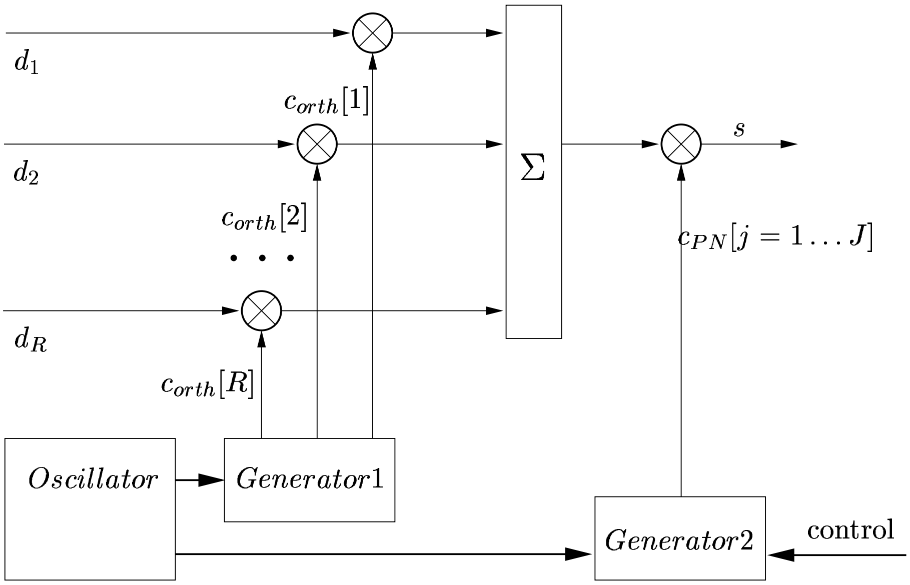


Figure 2.36: Multi–code CDMA sender with overlapping sub–spaces using orthogonal and random spreading–sequences.

In Figure 2.36 one possible sender structure is given. In contrast to the sender structure in Figure 2.35 firstly the data signals d_1 up to d_R are spread with a small orthogonal spreading sequence. The spreading sequences are different and are used to identify different channels of one WT. After the multiplexing process the multiplexed signal is spread again with one unique PN sequence. This PN sequence has to be assigned by one central entity and signaled once over the `control` channel. While the PN sequence is unique and controlled by one central entity, the list of orthogonal sequences can be the same for all WTs. Note, even for this sender structure only one `Oscillator` entity is necessary for all channels. The advantage of this allocation schemes is given by smaller interferences than for the *random subspaces* and the signaling takes place only in the beginning of one communication session. The disadvantage is the need for a centralized entity. This makes this approach impractical for certain scenarios such as given in 9.2.2.

2.3.3.1.3 Controlled Overlapping Strong overlapping of subspaces discussed in the previous section can result in a significant performance degradation. We introduce a new approach which is called *controlled overlapping*. The interference cannot be completely avoided (Figure 2.34) since it would require sequences with perfect aperiodic correlation functions and such sequences do not exist. In case the *controlled overlapping* approach is used and full synchronism in terms of chip and phase can be achieved **W**elch **B**ound **E**quality (*WBE*) [78] sequences are used. The bit error probability is given in Equation 2.28.

$$p_{biterror}^{WBE}(k) = \begin{cases} 0 & : \quad k + b \leq G_{Spreading} \\ Q\left(\sqrt{\frac{G_{Spreading}}{k - G_{Spreading} + b}}\right) & : \quad k + b > G_{Spreading} \end{cases} \tag{2.28}$$

where k is calculated as in Equation 2.26.

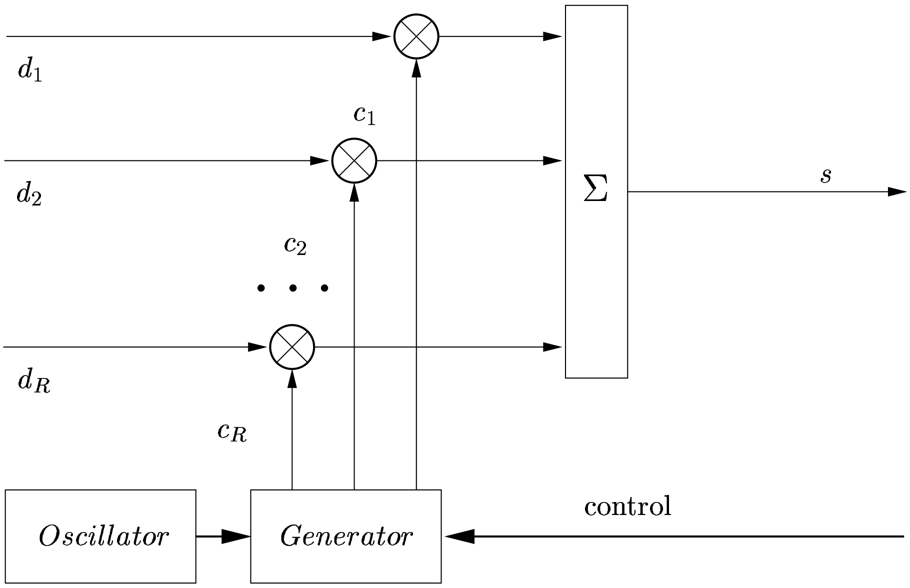


Figure 2.37: Multi–code CDMA sender for controlled overlapping.

As given later in Figure 2.38 the WBE sequences produce smaller interferences than any other spreading allocation scheme. The problem is that every spreading sequence in the system has to be changed if only one channel is added or released to achieve the best system performance. Every time a spreading sequence has to be changed signaling over the wireless link is necessary. In case the sequences are not changed the system becomes sub–optimal. There might be a trade–off between the signaling period and the signaling overhead. But nevertheless on wireless links, which are highly error–prone, the signaling can be lost or be delayed and this results in a high performance degradation. In Figure 2.37 one possible sender structure for the controlled overlapping scheme is given. The only difference to Figure 2.35 is the control channel to the `Generator` entity.

2.3.3.1.4 Comparison of different spreading allocation schemes In Table 2.5 the differences of the spreading allocation schemes in terms of interference, signaling, and need for a centralized entity is given. The symbols `+`, `-`, `++`, and `--` refer to the amount of interference, signaling or need for a centralized entity.

Table 2.5: Comparison of different allocation schemes.

	Interference	Signaling	Need for centralized entity
Random Sequences	++	−−	NO
Random Subspaces	+	−	YES
Controlled Overlapping	−−	++	YES

2.3.3.1.5 Illustration of the BEP for different approximations Figure 2.38 depicts the error probability of the four standard Gaussian approximations, the improved Gaussian approximation, and the WBE sequences. The four standard Gaussian approximations show nearly the same characteristic. Note, the asynchronous system offers the best error behavior. A dramatic performance gain can be achieved if WBE sequences are applied. Note, that for such systems a large signaling overhead and hard system requirements in terms of full synchronism has to be taken under consideration.

2.3.3.2 Packet Error Probability in CDMA

Considering the bit error probabilities in Equation 2.20, 2.22, 2.23, 2.21, 2.24, and 2.28 we give the **P**acket **E**rror **P**robability (*PEP*) for a packet data unit of the length L_{PDU} [bit] in Equation 2.29 under the assumption that the bit errors occur without correlations.

$$p_{pkterror}(L_{PDU}, k) = 1 - (1 - p^*_{biterror}(k))^{L_{PDU}} \tag{2.29}$$

If we assume a coding scheme that allows us to correct e bit errors the packet error probability becomes smaller and is given in Equation 2.30[79, 53, 80].

$$p_{pkterror}(e, L_{PDU}, k) = 1 - \sum_{i=0}^{e} \binom{L_{PDU}}{i} (1 - p^*_{biterror}(k))^{L_{PDU}-i} (p^*_{biterror}(k))^i \tag{2.30}$$

In Figure 2.39 we depict the packet error probability $p_{pkterror}$ for different Gaussian approximations.

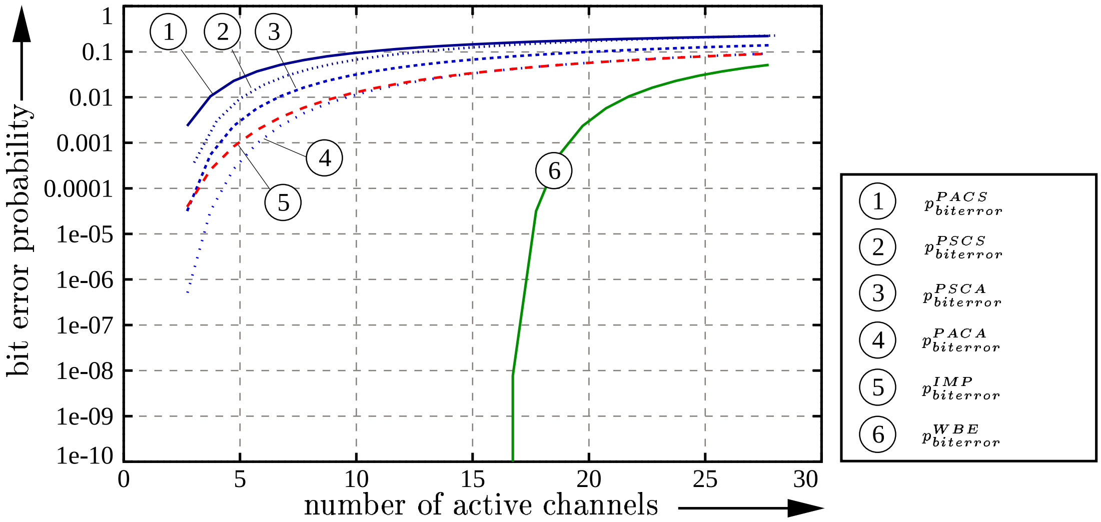


Figure 2.38: Bit error probability $p^{*}_{biterror}$ for different Gaussian approximations and WBE sequences with b=0 and $G_{Spreading}$=16.

2.3.3.3 Maximum Throughput for CDMA Systems for Packet Services

It was already mentioned that CDMA systems are able to tolerate overload, if all wireless terminals in one cell will tolerate a certain degradation in their performance. The main goal is now to investigate the *capacity* defined as the number of active terminals within a single cell CDMA system achieving the maximal throughput. Therefore we look at Equation 2.31 (see [79]), which gives the throughput T for a single CDMA cell in dependency of the packet length L, the size of redundancy R, the number of correctable bits e, the number of active wireless terminals k, and the type of Gaussian approximation.

$$T(L, e, k) = (L - R) \cdot k \cdot (1 - p^{*}_{pkterror}(k)) \tag{2.31}$$

For different Gaussian approximations the throughput of one CDMA cell is depicted in Figure 2.40. The behavior of all throughput curves is nearly the same. For a small number of active channels the throughput increases linearly. At a specific point (different for all Gaussian approximations) the throughput decreases rapidly. Thus for each Gaussian approximations an optimal number $k_{opt} = max(T(k))$ of active channels can be found, where the throughput of the cell is maximal. For a given Gaussian approximation Figure 2.40 shows the optimal number of supportable active channels. The Gaussian approximation $PACA$ has a number of 13 active channels resulting in an overall throughput of 9kbit per slot.

We will show in course of this work that it might lead to a better performance in terms of delay constraints and loss probability, if other than the optimal number k_{opt} of CDMA channels is distributed among the WTs. From this it follows that not the optimal throughput for a single cell CDMA system will be achieved, but WTs can be supported with their heterogenous and time–variable QoS requirements. Note, in case we distribute the optimal number k_{opt} of CDMA channels, the PEP is approximately 10%. This large PEP might be insufficient for some applications.

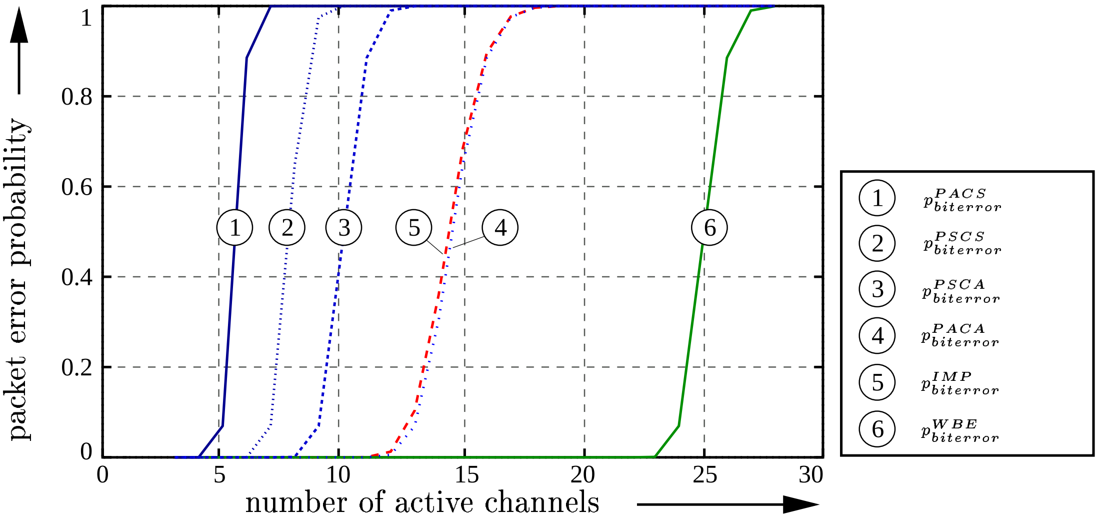


Figure 2.39: Packet error probability $p_{pkterror}$ for different Gaussian approximations and WBE sequences with b=0, e=30, L_{PDU}=1023 bit, and $G_{Spreading}$=16.

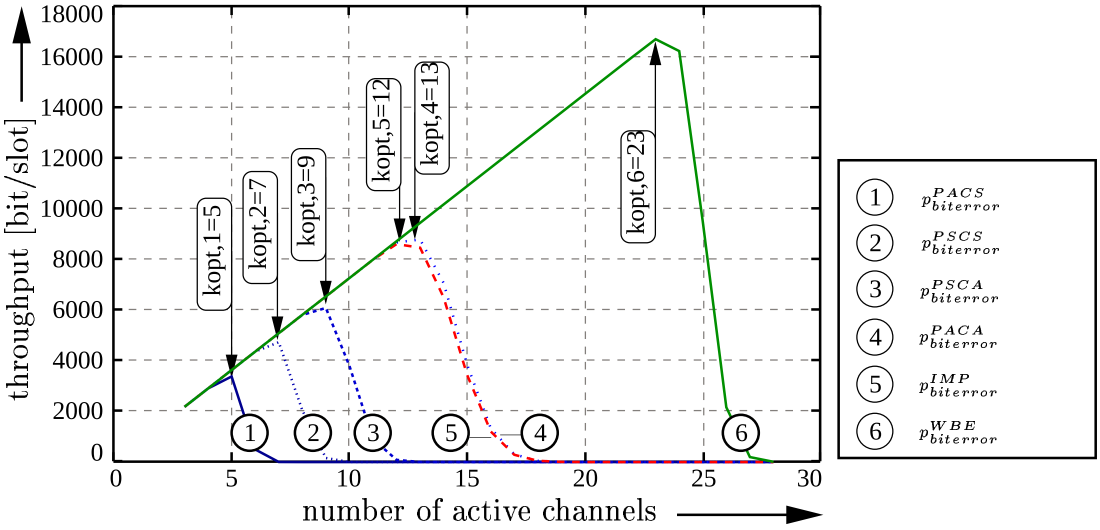


Figure 2.40: Throughput T for different Gaussian approximations and WBE sequences with b=0, e=30, L_{PDU}=1023 bit, and $G_{Spreading}$=16.

3 Related Work in Link Level QoS Support

This chapter gives a survey of link–level QoS support. For CDMA systems the power control unit has been identified as a controllable parameter for QoS support. We describe the general applicable **F**orward **E**rror **C**orrection (*FEC*) scheme. Mechanisms like **A**utomatic **R**epeat Re**Q**uest (*ARQ*) mechanisms are explained. Several concepts are introduced which use **M**ulti–**C**ode CDMA (*MC–CDMA*). Because this work is based on MC-CDMA we report on some work in this field.

3.1 Power Control

Different power control algorithms for time–variable QoS support have been introduced in the literature. The key problem for all algorithms is to know the transmission information, i.e. whether a WT will send or not. This is important because the actual number of active WTs results in different transmission powers. To deal with this lack of information the following approaches have been proposed: 1.) In [63] power control on the assumption that all WTs transmit permanently is presented. This will lead to a conservative utilization of system capacity, because it is based on a peak–rate bandwidth allocation. On the other side no information of the WT's transmission state is needed. 2.) In [64] a control channel is introduced to signal the WT's transmission requests. 3.) Another approach, presented in [66, 65], is a statistical description of the WT's traffic characteristic. This approach is implemented in a distributed fashion and is adequate when the number of WTs is large. For a small number of WTs it is not adequate.

The following approaches based on power control are discussed in more detail, since they concentrate on QoS support for higher protocol layers. In [81] the adaptive power control is used to provide time–variable QoS support for the network layer. The proposed schemes calculate the maximum number of retransmissions for MAC packets using their due–date information. Furthermore different traffic classes are introduced, where each class is represented by a given maximum loss rate and given number of retransmissions. In [57] two families of power control algorithms **P**ower **C**ontrolled **M**ultiple **A**ccess (*PCMA*) –1 and PCMA–2 for the realistic situation of responsive interference are introduced. The authors designed their algorithms under the constraints that these algorithms should share the wireless channel in an efficient way offering the required QoS in terms of power consumption and packet delay constraints to each WT. To maximize the channel capacity no overwhelming signaling effort has to be taken into account. Therefore the algorithms should be *simple*, *distributed*, *autonomous*, and *asynchronous* [57]. We discuss this point in Section 4 and verify the algorithms presented in this work with these attributes. The basic concept of the presented schemes is to transmit power when there is low noise or interference and not transmit power when the interference is high. Both algorithms, PCMA–1 and PCMA–2, have three different operational phases depending on the transmission queue and the interference situation of the wireless link. To illustrate the algorithm, assume the situation where queued packets are transmitted successfully over the wireless link and further packets have to be transmitted. **Aggressive Phase:** In case the interference on the wireless link increases both algorithms try aggressively to match and overcome the interference by increasing power. **Soft–backoff Phase:** In case the interference increases beyond some threshold (different for the algorithms) they start lowering the transmit power until the power becomes zero. **Hard–backoff Phase:** As the interference increases further the WT goes into the hard–backoff mode. In Figure 3.1 the operational phases of the PCMA algorithm in combination with the related signal power and interference is given. The intuitive explanation for the algorithms is, that under changing interference conditions the algorithms try to match a certain signal noise ratio and therefore guarantee an error–free communication. If the interference is too large it is better to backoff and wait for lower interference on the wireless link. Note, there is always a trade–off between invested power and packet delay. Throughout simulations it was shown that both PCMA algorithms offer higher system capacity than a generally accepted benchmark for power control presented in [82]. For a low load situation the PCMA algorithms have slightly larger mean delay values. But while the algorithm presented in [82] lead to instability for a lesser amount of load, the PCMA algorithms are still stable for a larger amount of load. The reason for the higher load is that wireless terminals with bad channel conditions take themselves out of the transmission process and lead to better signal noise situation within the system.

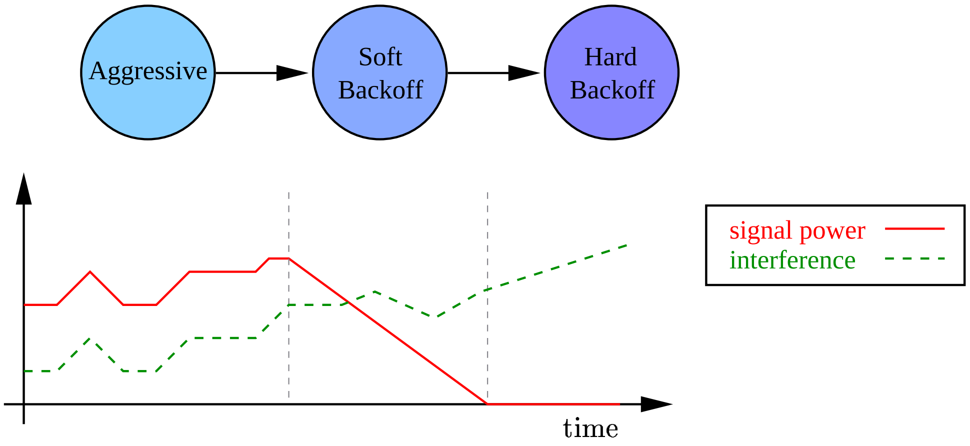


Figure 3.1: Operational phases of the power controlled multiple access scheme with related signal power and interference.

3.2 Forward Error Correction

Shannon's channel coding theorem states that under the assumption that 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 at the transmitter to each information packet. At the receiver this redundancy will be used to detect and/or correct errors within the information packet.

A FEC scheme based on block coding is called a binary (n,k,e) **B**ose–**C**haudhuri–**H**ocquenghem (*BCH*) code [83]. This code allows to correct e bit errors in a $n = 2^m - 1$ bit long word adding k bits redundancy. Values for n,k, and e are given in [83]. A subclass of non–binary BCH codes are represented by **R**eed **S**olomon (*RS*) codes. An (n,k,e) RS code is capable of correcting any combination of e or fewer symbol errors within the block–length n. If the payload consists of k symbols the number of correctable symbol errors equals $\frac{n-k}{2}$. If even one bit within a symbol is prone to an error, the symbol is also error–prone. A RS code with symbols of length j bits can maximally correct $j \cdot e$, if only e symbols are prone to errors. Thus, RS have an excellent burst error suppression capability.

An advantage of FEC is constant throughput with bounded time delays independent of error probabilities on the wireless channel. However, in [84] it is claimed that FEC schemes must be implemented for the worst case channel characteristics to achieve error–free communication, if the wireless link is non–stationary, and therefore the error probability varies over time. Thus, static FEC is always associated with unnecessary overhead reducing the channel throughput when the channel is nearly error–free. Furthermore, to obtain high system reliability, a variety of error patterns have to be corrected. This requires a powerful, long code resulting in complex hardware structures and high transmission overhead. By adding redundancy the latency of all packets will increase constantly, which might be not acceptable for real–time services and a wide range of data–services. In order to overcome the described FEC problems *adaptive* FEC was introduced. *Adaptive* FEC applies redundancy in dependency on the characteristics of the loss process at that time. In [85] *adaptive* FEC is applied to support Internet telephony and the authors claim that other mechanisms like **A**utomatic **R**epeat Re**Q**uest (*ARQ*) (see Section 3.3) are not acceptable for real–time audio applications. Nevertheless, *adaptive* FEC schemes will lead to higher hardware complexity and possibly larger signaling overhead.

We have already mentioned, that for a high system performance FEC has to be adaptive to the time–variable channel conditions. For CDMA systems the channel conditions depend strongly on the number of overall active channels. In dependence to the applications used by the WTs the load might change from slot to slot. Figure 3.2 shows the bit error probability p_{bit} versus the number of active channels. We took the improved Gaussian approximation for calculation of the bit error probability of a CDMA system with a spreading gain $G_{Spreading} = 16$ (see Equation 2.24). Figure 3.3 depicts the resulting packet error probability versus the number of active channels for packet length 511 bit for different redundancy levels. Obviously more redundancy leads to smaller packet error probabilities. We chose a BCH code where we can correct 0,8,16,31, and 63 errors of a 511 bit long packet. The packet length, the redundancy, and the number of correctable bits are taken from [83].

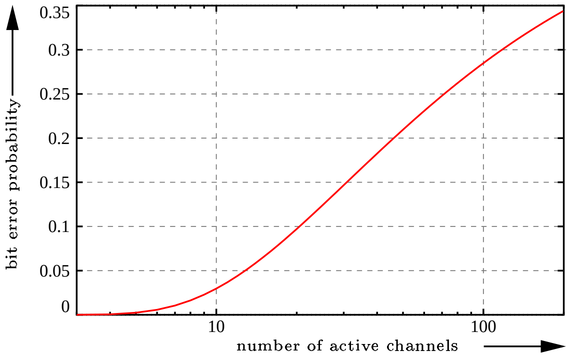


Figure 3.2: Bit error probability p_{bit} versus the number of active channels within single cell CDMA system for packet length 511 bit.

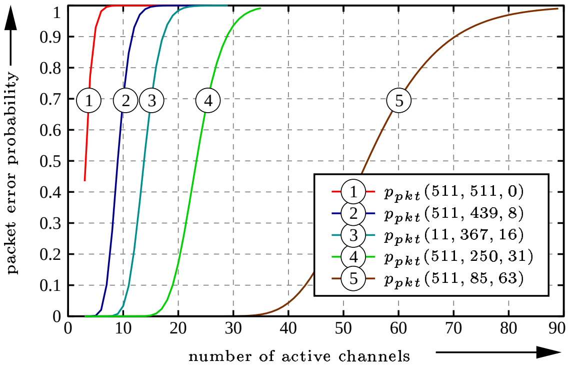


Figure 3.3: Packet error probability versus the number of active channels within single cell CDMA system for packet length 511 bit for different redundancy levels.

The overall throughput for a single CDMA system within one time slot is depicted in Figure 3.4 for different redundancy levels. The coding schemes offers the highest throughput at different number of channels. The scheme with 261 bit redundancy (curve 4) offers the highest throughput for a single cell for a given number of 18 active channels. Figure 3.5 gives the throughput values for a single WT. The highest throughput for a WT can be achieved with low redundancy. With low redundancy only a few WTs can be supported in the cell.

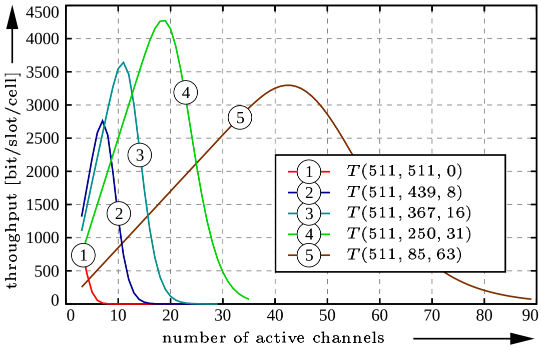


Figure 3.4: Cell throughput versus the number of active channel within single cell CDMA system for different redundancy levels.

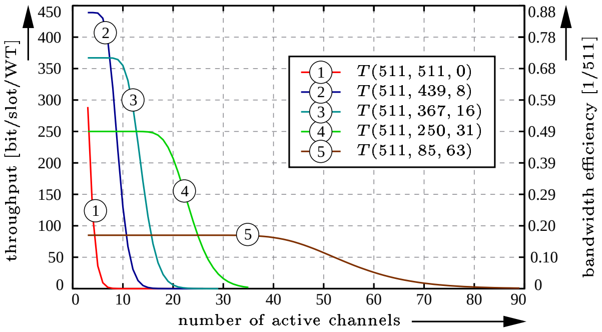


Figure 3.5: Throughput of single WT versus the number of active channel within single cell CDMA system.

3.3 Automatic Repeat Request

The general concept of **A**utomatic **R**epeat Re**Q**uest (*ARQ*) is to detect error–prone **L**ink–layer **P**acket **D**ata **U**nit (*LPDU*) and to cause the **D**ata **L**ink **L**ayer (*DLL*) transmitter to repeat the transmission process for these packets. Error detection by the means of FEC was introduced in Section 3.2. This data link method offers a reliable service for the network layer. Different approaches have been introduced. The approaches differ in complexity and efficency of the retransmission process. The simplest type of a retransmission 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, retransmissions have to be done until the packet has been received successfully. Note, some approaches work with negative acknowledgments. In this work we will concentrate on the *Send and Wait* protocol. 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. However, the problem of the ARQ protocol is the introduction of variable delays, which are not acceptable for real–time services [84, 86].

3.4 Channel Aggregation

One possible solution for channel aggregation using a common code was shown by Bito etal. [87]. The authors considered the case where the number of spreading codes is limited. They propose a scheme that assigns each terminal exactly one unique spreading code. If a terminal wants to use additional CDMA channels it uses its spreading code with random time offsets on the additional channels. Two or more channels of a given terminal that happen to be bit-synchronous will have unacceptably large bit error probabilities. Therefore, each terminal has to make sure that its CDMA channels are bit–asynchronous. For channel aggregation on the IS–95 downlink Jungbauer etal. [88] propose the use of turbo codes for channel aggregation in IS–95. They focus on the downlink of a synchronous CDMA system. An IS–95 CDMA channel provides a bit rate of up to 9.6kbit/sec. To support applications with higher bit rates, multiple channels are aggregated using MC–CDMA. Jungbauer etal. show that in combination with turbo codes the transmission quality of a terminal is independent of the number of codes it uses (i.e., the co–channel interference is suppressed; convolutional codes do not have this property). The idea of using channels in parallel to support QoS in terms of bandwidth was already used in **I**ntegrated **S**ervices **D**igital **N**etwork (*ISDN*). Therefore we provide a short overview of this technique and the appropriate protocols. Afterwards we give an overview of **M**ulti–**C**ode CDMA (*MC–CDMA*) related work.

3.4.1 Rate Adaption in ISDN with PPP

Integrated **S**ervices **D**igital **N**etwork (*ISDN*) is a system for digital phone connections. ISDN is based on two major types of channels. The first type is called a *bearer* or B–channel. A user information of 64kbit/s can be placed on a B–channel. The second type is called *common signaling* or D–channel. The function of the D–channel is to carry information to establish connections, terminate connections, or exchange status reports. The D–channel operates at 16 kbit/s. Typical connection times for call setup are three seconds. The latency on an ISDN line is typically about half that of an analog line. This improves the performance of interactive applications, such as games.

Regarding the ITU standards ISDN exists in two major service varieties: **B**asic **R**ate **I**nterface (*BRI*) and **P**rimary **R**ate **I**nterface (*PRI*). The BRI is the most common ISDN service providing two B–channels and one D–channel. Therefore BRI is also referred to as 2B+D. For higher capacity ISDN offers also the PRI service. Depending upon the country where it is deployed PRI supports 23 B–channels and a single D–channel (North America, Japan, and Korea) or 30 B–channels and one D–channel (rest of the world). These two types are referred to as 23B+D or 30B+D respectively. It is also possible to support multiple PRI lines with one 64 kbit/s D–channel using **N**on-**F**acility **A**ssociated **S**ignaling (*NFAS*).

Rate adaptation in ISDN is performed in different ways: The first solution is pure ISDN based. Multiple B–channels operate simultaneously for one communication session permitting a much higher data transfer rate. Using a channel aggregation protocol such as BONDING supports a user information rate of 128 kbit/s using two B–channels. The idea behind BONDING was already mentioned in RFC 1717 and later obsoleted by RFC 1990 [89, 90] defining the Multi–Link PPP [89]. This is now used in virtually all modern ISDN equipment. The second approach to use rate adaptation in ISDN is the use of the **P**oint–to–**P**oint **P**rotocol (*PPP*) [91]. Multi–Link PPP allows ISDN devices to bond multiple B–channels into a logical channel. More sophisticated versions of Multi–Link PPP allow dynamic usage of the B–channels: additional with channels are brought up if needed and released if the need for extra bandwidth passes.

3.4.1.1 Static Rate Adaptation in ISDN

BONDING is the possibility within ISDN to bundle multiple B–ISDN channels. Each B–ISDN channel is setup separately. Before the full bandwidth can be used delay equalization aligning the B–ISDN channels is performed. Because the delay alignment procedure takes place only once, and is not permanently monitored, any change in delay of one of the B–ISDN channels lead to error–prone data transfer. Delay recovery can be achieved by disconnecting and reestablishing both B–ISDN channels. With BONDING each data segment is splitted bit by bit between two B–channels. On the receiver–side the bits are reassembled to the data segment. We refer to this as *static* rate adaptation because both B–channels are always in use (and therefore have to be paid by the subscribers). A more *dynamic* scheme is described in the following section.

3.4.1.2 Dynamic Rate Adaptation in ISDN based on PPP

Dynamic rate adaptation in ISDN is based on the protocol family of PPP. Hence, we first give a short introduction in the PPP and the Multi–Link PPP. Interested readers refer to RFC 1618 [92], where an explanation for PPP over ISDN is given.

The **P**oint–to–**P**oint **P**rotocol (*PPP*) is introduced in the RFC 1661 [91]. The protocol is designed to allow full–duplex transmission of packets over simple communication links between two terminals. There is no special need for control signals; if available using control signals can lead to greater functionality. In contrast to the particular ISDN interface, PPP has no limitation regarding the transmission rate. Note, PPP is independent of the under–laying presentation of the physical layer.

In case both terminals are capable to run the Multi–Link PPP specified in RFC 1717 [89], the transmission bandwidth can be increased by bundling serval B–channels. Multi–Link is negotiated by sending invitation messages during the initial communication phase. Higher protocol PDUs which are fragmented at the sender–side are reassembled at the receiver side. Each fragment is identified by the multi–link header. If fragments are sent over different channels in the bundle, it can happen that fragments are received out of order, because of different delay properties of the channels. The key–feature of the Multi–Link in contrast to BONDING is the possibility to switch on or off channels in dependency of the bandwidth–requirements given by the higher protocol layers. In [92, 90] it is recommended to use the Multi–Link PPP procedure instead of BONDING. The reasons are the following: (1) BONDING has its own initialization procedure, which might conflict with other techniques, (2) PPP is only software based and therefore no change in hardware has to be done, and (3) the dynamic usage of channels like it is done in multi–link PPP might lead to smaller costs.

Once the capability of dynamic rate adaption is given, we have to specify how to manage bandwidth over such communication links. One approach to manage the dynamic bandwidth is given in RFC 2125 [93] and is called the PPP **B**andwidth **A**llocation **P**rotocol (*BAP*). The proposed approach is able to manage the number of links in a multi–link bundle. It coordinates the adding and the dropping of individual links in a multi–link bundle. The channel–management can be based on link utilization or the available resources. In case it is based on the link utilization each terminal monitors its outgoing and ingoing traffic. Based on the traffic measurements the BAP implementation is able to drop and add channels from the current multi–link bundle. Channel–management based on available resources is implemented to drop channels from the bundle if high priority traffic is established or the maximum number of channels is reached. Both approaches (traffic or resource based) can be controlled by the sender, receiver or both of them. Mechanisms have be implemented, which resolve situations where both entities request a drop or add command for the channel bundle.

3.4.2 Distributed–Queuing Request Update Multiple Access

Distributed–**Q**ueueing **R**equest **U**pdate **M**ultiple **A**ccess (DQRUMA) is an efficient protocol for provisioning and allocating resources in wireless packet networks. DQRUMA schedules transmissions in both the downlink as well as the uplink direction. It uses separate frequencies for downlink and uplink transmissions, i.e., **F**requency **D**ivision **D**uplex (*FDD*). In the original DQRUMA [94, 95, 96, 97, 98] the terminal's resource reservation requests contend for a random access slot. Multiple CDMA code channels are assigned to the requesting terminal for one or multiple **D**ata **L**ink **L**ayer (*DLL*) time slots. At the end of a time slot each terminal may be assigned a new set of code channels. The total number of available channels is estimated from the SNIR. The channels are assigned to the individual terminals by a Bandwidth–on–Demand Fair–Sharing Round–Robin Schedule (BoD–FSRR). Figure 3.6 gives the frame structure of DQRUMA for MC–CDMA. The frame consists of the following slots: *Request Access*, *Piggybacking Request*, *Packet–Transmit Channel* for the uplink and *Acknowledgment of Request Access*, *Packet–Transmit*

Permission, and *Packet–Transmit Channel* for the downlink.

There is an offset τ_{offset} between the uplink and downlink of a given frame. This allows an immediate feedback to terminals that send reservation requests into the random access contention. In case a reservation request collides (i.e., is not successful), the corresponding terminal might send a new request in the random access slot of the next frame. Collisions occur since there is only a limited number of codes — which may be smaller than the number of terminals — available for the request access. An improvement that reduces the probability of collisions in the request access of DQRUMA was proposed in [99]. Terminals which are successful in the random access refrain from sending a new request when the central scheduler is not able to assign the requested channel. Terminals that are already in the reservation mode may piggyback their reservation requests (i.e., in the slot preceding their MC–CDMA slot, which is reserved for the terminal and thus free from collisions). In both approaches, random access and piggyback, the scheduler notifies the terminals eligible for transmission in the packet transmit slot. The simulation results presented in [94, 95, 96, 97, 98] rely on the assumption that the base station has perfect knowledge of the terminal's bandwidth needs. The proposed Bod–FSRR protocol consists of two phases. In the first phase the protocol executes a Fair–Sharing Round Robin policy. The protocol maintains a counter for each service class. The counter is incremented whenever a slot is assigned to the corresponding service class. The counter is decremented periodically; different service classes have different decrement periods. A negative counter indicates that the service class' QoS requirements are violated; consequently, the scheduler grants the corresponding back-logged transmission requests.

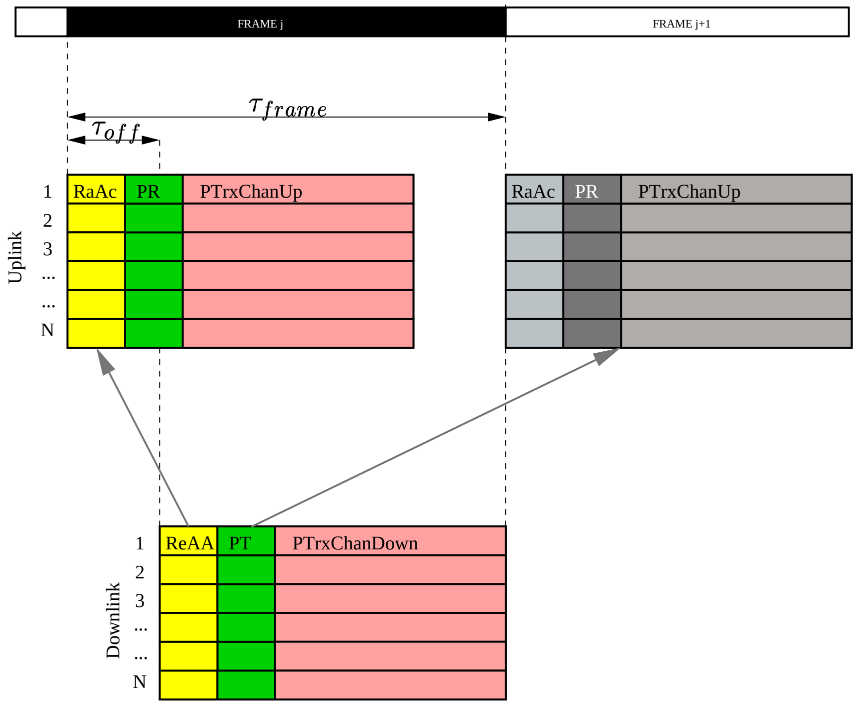


Figure 3.6: DQRUMA frame structure.

This procedure is repeated until the system capacity is exhausted, or all counters are non–negative. In case the first procedure does not use up the available capacity, the second procedure is executed. According to this second procedure, the best–effort component of DQRUMA, is based on round-robin scheduling. Time slots are assigned to the terminals with transmission requests in round–robin fashion and their counters are incremented, until the system capacity is exhausted.

The **M**aximum **C**apacity **P**ower **A**llocation (*MCPA*) approach relies on the assumption that there is no self–interference between the CDMA channels of a given wireless MC–CDMA terminal. The self–interference is eliminated through judicious selection of the spreading codes [100]. The power control unit calculates the uplink transmission power for each channel such that all channels are received with the same power level at the base station. All downlink channels have the same transmission power. This puts terminals with one channel at a disadvantage over terminals with multiple channels: a well–known problem [94]. Consider a CDMA based wireless cell with N active channels. Suppose that a given terminal uses n channels; clearly, for this terminal $N - n$ channels contribute to the SNR. For a terminal using one channel, on the other hand, $N - 1$ channels contribute to the SNR. The maximum capacity in a MC–CDMA wireless cell is achieved when the transmission power for each channel is adjusted such that (i) the total (combined) transmission power of all channels is minimized, and (ii) the required SNR for each channel is met. It is shown in [94] that the transmission power of the terminals with more than two channels is governed by the transmission power levels of terminals with one channel. The transmission power levels of terminals with one channel, in turn, depends on the distribution of the active channels of the wireless cell among the terminals.

3.4.3 Load and Interference based Demand Assignment

Existing CDMA systems of the second generation such as the IS–95 (Rev A) [101] digital cellular system by Qualcomm already support voice and data traffic at a basic rate. Further improvements on IS–95 (Rev B) [102] lead to the usage of Multi–Code CDMA, where mixed traffic can be supported in combination with the **L**oad and **I**nterference Based **D**emand **A**ssignment (LIDA) approach [19]. A high speed capable WT can assign up to seven additional channels, called the **S**upplemental **C**ode c**H**annel (*SCH*) on top of the **F**undamental **C**ode c**H**annel (*FCH*), which results in a overall number of eight usable channels. The channels are controlled by the infrastructure (e.g. by the base station) and assigned to the WT through a dedicated signaling channel. The decision whether channels will be assigned or not depends on pilot tone measurement. In [20] a **B**urst **A**dmission based on **L**oad and **I**nterference (BALI) approach is introduced. Once channels are assigned to the WT by the LIDA approach the BALI approach uses these channels to empty the queue on sender side. Whenever the queue is filled with multiple packets, the WT requests multiple channels to empty the queue. Ejzak etal. propose BALI for systems based on the IS–95 standard. BALI is similar to DQRUMA in that it aggregates CDMA code channels to accommodate bursty traffic.

3.4.4 ATM–based Multi–Code CDMA Transport for MPEG Video Transmission

Video transmission over multi–code CDMA systems is studied in [80]. The scheme proposed in [80] is based on the wireless **A**synchronous **T**ransfer **M**odus (*ATM*) approach. Multi–code CDMA is used because it is well suited for bandwidth allocation supporting **V**ariable **B**it **R**ate (*VBR*) MPEG–2 video. The approach presented in [80] requires a significant amount of coordination among the videos being transmitted (for instance, the video streams are aligned such that a (typically large) Intracoded (I)–frame of one video stream does not coincide with the I–frame of another video stream). The concept of I–frames is introduced in Section 5.1.1.

3.4.5 Wireless JSQ

In [103, 62, 7] we developed a high performance streaming protocol for the real–time delivery of prerecorded continuous media over wireless links, which is henceforth referred to as **W**ireless **J**oin the **S**hortest **Q**ueue (*WJSQ*) . The protocol is equally well suited for streaming in the downlink (base station to clients) direction, as well as the uplink (clients to base station) direction. The protocol allows for immediate commencement of playback as well as near instantaneous client interactions, such as pause/resume and temporal jumps. The protocol achieves this high performance by exploiting two special properties of *prerecorded* continuous media: (1) the client consumption rates over the duration of the playback are known before the streaming commences, and (2) while the continuous media stream is being played out at the client, parts of the stream can be prefetched into the client's memory. The prefetched reserves allow the clients to continue playback during periods of adverse transmission conditions on the wireless links. In addition, the prefetched reserves allow the clients to maintain a high perceptual media quality when retrieving bursty **V**ariable **B**it **R**ate (*VBR*) encoded streams.

The prerecorded continuous media streams are prefetched according to a specific **J**oin the **S**hortest **Q**ueue (*JSQ*) policy, which strives to balance the prefetched reserves in the wireless (and possibly mobile) clients within a wireless cell. The JSQ prefetch policy exploits rate adaptation techniques of wireless data packet protocols [104]. The rate adaptation techniques allow for the dynamic allocation of transmission capacities to the ongoing wireless connections. In the **C**ode **D**ivision **M**ultiple **A**ccess (CDMA) IS–95 (Revision B) standard [9], for instance, the rate adaptation is achieved by varying the number of codes (i.e., the number of parallel channels) used for the transmissions to the individual clients. Multiple channels per WT is achieved with **M**ulti–**C**ode CDMA (*MC–CDMA*) . The total number of code channels used for continuous media streaming in the wireless cell may be constant. The JSQ prefetch policy dynamically allocates more transmission capacity to wireless clients with small prefetched reserves while allocating less transmission capacity to the clients with large reserves. The ongoing streams within a wireless cell collaborate through this lending and borrowing of transmission capacities. In Figure 3.7 we give the streaming of VBR MPEG–1 videos to clients. The figure shows the prefetch buffer contents of three clients in kBytes. The plots illustrate the collaborative nature of the WJSQ prefetch policy in conjunction with the rate adaptation of the wireless

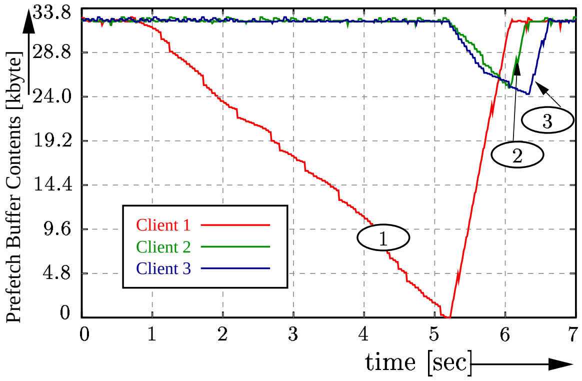


Figure 3.7: Sample path plot: Prefetch buffer contents (in kBytes) of 3 clients as a function of time: Client 1 starts over [7].

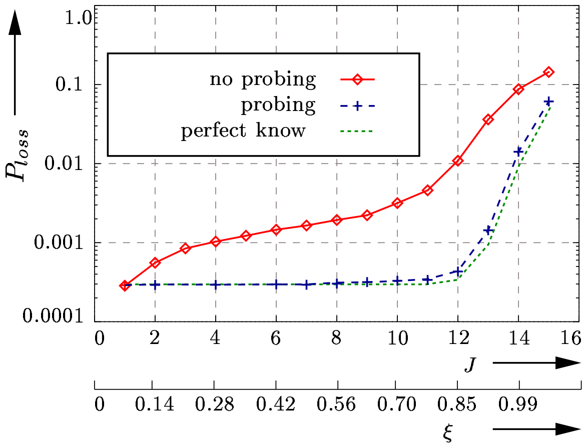


Figure 3.8: Client starvation probability P_{loss} as a function of the bandwidth efficiency ξ (obtained by varying the number of clients J) for VBR video and prefetch buffer $B = 32$ kByte [7].

communication system. We observe from Figure 3.7 that at time $t = 5.1$ sec the buffer content of client 1 drops to zero. This is because the video stream of client 1 ends at this time; the client selects a new video stream and starts over with an empty prefetch buffer. Note that already at time $t = 1.0$ second all frames of the "old" video stream have been prefetched into the client's buffer and the client continued to consume frames without receiving any transmissions. When the client starts over with an empty prefetch buffer, the JSQ prefetch policy gives priority to this client and quickly fills its prefetch buffer. While the prefetch buffer of client 1 is being filled, the JSQ prefetch policy reduces the transmissions to the other clients; they "live off" their prefetched reserves until client 1 catches up with them. Notice that the buffer of client 1 is then filled faster than the buffers of clients 2 and 3. This is because the JSQ prefetch policy strives to balance the lengths of the prefetched video segments (in seconds of video runtime) in the clients' buffers; clients 2 and 3 just happen to have video segments with lower bit rates in their buffers in the time period from 5.8 sec to 6.4 sec.

Channel probing is used to judiciously utilize the transmission capacities of the wireless links, which typically experience location–dependent, time–varying, and bursty errors. Figure 3.8 shows the client starvation probability P_{Loss} as a function of the bandwidth efficiency ξ. The client starvation probability P_{Loss} achieved with channel probing is (i) generally over one order of magnitude smaller than without channel probing, and (ii) only slightly larger than with perfect knowledge of the channel state. Our extensive numerical studies indicate that this collaboration is highly effective in reducing playback starvation at the clients while achieving a high bandwidth efficiency. For bursty VBR video with an average rate of 64 kbps and typical wireless communication conditions our prefetching protocol achieves client starvation probabilities on the order of 10^{-4} and a bandwidth efficiency of 90% with client buffers of 128kBytes.

4 Simultaneous MAC Packet Transmission

In this chapter we summarize the shortcomings of the currently used approaches and hence provide the motivation for the development of a new scheme to support multimedia traffic over the wireless channels. We introduce a new scheme, which is called **S**imultaneous MAC **P**acket **T**ransmission (*SMPT*). MAC stands for **M**edium **A**ccess **C**ontrol (*MAC*). SMPT is based on the capability of **W**ireless **T**erminals (*WTs*) to transmit on multiple channels. This can be achieved by using **Multi**–**C**ode CDMA (*MC–CDMA*). MC–CDMA has some advantages over other multi–rate CDMA techniques and therefore it will be used for the following approach. A set of different approaches are introduced for correlated and uncorrelated link characteristics. Their drawbacks and advantages are discussed. The impact of a delayed feedback channel on the performance of SMPT is discussed. Finally the SMPT system design is given.

4.1 Synopsis and Motivation

In [57] the authors have mentioned that to maximize the channel capacity no overwhelming signaling has to be introduced. Therefore the algorithms should be *simple*, *distributed*, *autonomous*, and *asynchronous*. Thus we want to analyze the related work presented in Chapter 3 with regard to these attributes and their possibility of QoS support. DQRUMA, LIDA/BALI, WJSQ, MPEG–ATM are based on a centralized entity to assign the MC–CDMA resources and therefore are neither *distributed* nor *autonomous*. These approaches can only be applied to cellular systems. While WJSQ and LIDA/BALI are *simple* and therefore easy to implement without much hardware complexity, DQRUMA and MPEG–ATM are not. The four approaches differ in the QoS support. LIDA/BALI supports only channel dependent burst allocation. This means in case of a non empty queue the approach tries to empty the queue as fast as possible without taking the required QoS into consideration. The QoS support of WJSQ takes the receiver side queue into consideration. Therefore this approach is well suited for streaming services. The MPEG–ATM only supports MPEG based video services and is not suited for different traffic services. DQRUMA has applied a Bandwidth–on–Demand Fair–Sharing Round–Robin Scheduler. This allows to support any traffic. Measurements of the SNIR in the cell are made to detect the absolute number of supportable channels in the next uplink slot. This measurement can only be done by the means of a centralized entity. The PCMA approaches are not based on MC–CDMA, but they are *simple*, *distributed*, *autonomous*, and *asynchronous.* The QoS support is based on the physical layer, but no multi–rate support is applied. Moreover the advantage of data link layer approaches is the possibility to achieve a certain knowledge about the requirements of packets, while the physical layer approaches are limited to the QoS support of streams of packets. In Table 4.1 the characteristics of these algorithms are given.

It has already been pointed out earlier that in the future there will be an increased demand for multimedia services. These multimedia applications typically need stable throughput, small jitter and a limited loss rate. Until recently, multimedia protocols were designed for wired networks. But the ongoing development of omnipresent mobile communication environment makes solutions necessary that are suitable for wireless networks. In contrast to wired networks wireless links oscillate between *good* and *bad* states. This means they have temporary outage periods where the wireless channel is prone to correlated bursts of errors [40, 105]. From the application point of view a *bad* channel condition is recognized by a reduced throughput, higher loss rate, higher jitter or higher delay. Instead of attempting to tolerate the unstable service (e.g. adaptive applications [10, 11]) we introduce a mechanism that helps to stabilize the **Q**uality **o**f **S**ervice (*QoS*) of a **C**ode **D**ivision **M**ultiple **A**ccess (CDMA) system. To stabilize the service it is necessary to find a mechanism which reduces the loss rate without increasing the jitter, reducing the throughput or increasing the delay. Furthermore this algorithm should not waste bandwidth during good channel states.

In order to reduce the loss rate seen by the application either **A**utomatic **R**epeat Re**Q**uest (*ARQ*), **F**orward **E**rror **C**orrection (*FEC*) or hybrid mechanisms can be used to transmit packets. ARQ mechanisms are sometimes referred to as closed–loop mechanisms [85, 106], because packets that are not received successfully at the destination are retransmitted by the request from the destination. FEC mechanisms, on the other hand, are referred to as open–loop mechanisms. No feedback from the destination is needed (in case *static* FEC is used). The cost of using ARQ is an increased delay and a reduced throughput in case of a *bad* channel state because of the repeated retransmissions of erroneous packets. The retransmission of a single packet influences the delay of all succeeding packets of the transmitting queue. In case of real time multimedia applications segments with a delay higher than a specific limit are considered to be lost. The cost of using FEC is a reduced throughput, because of FEC overhead which has to be transmitted as well. Increasing the redundancy due to strong FEC coding during a *bad* channel state might correct errors but reduces the throughput compared to the *good* state. Since bandwidth is a scarce resource in wireless systems the redundancy due to FEC codes might be a function of the channel state (e.g. use FEC only during a *bad* channel state).

Errors on the wireless link reduce the throughput, because some **L**ink–layer **P**acket **D**ata **U**nit (*LPDU*) are prone to errors, cannot be decoded correctly and therefore lead to losses. Due to the variation on the wireless link, the throughput becomes unstable, i.e. varies over time. With *Send and Wait*, the simplest **A**utomatic **R**epeat Re**Q**uest (*ARQ*) mechanism (discussed detail in Section 3.3) each erroneous LPDU is retransmitted. The subsequent LPDUs in the transmission queue have to wait until the corrupted packet has been transmitted successfully. Figure 4.1 depicts an example of such a sequential transmission of eleven LPDUs in 16 time slots using *Send and Wait.* We assume that an ideal feedback channel exists and therefore the acknowledgment for each packet can be

Table 4.1: Characteristic of the related work.

	simple	autonomous	distributed	use MC–CDMA	QoS
DQRUMA [94, 95, 96, 97, 98]	no	no	no	yes (PN)	Fair Share Round Robin
LIDA/BALI [19, 20]	yes	no	no	yes (PN)	channel dependent burst allocation
WJSQ [103, 62, 7]	yes	no	no	yes (orth.)	depend on receiver side queue
MPEG–ATM [80]	no	no	no	yes (PN)	only for MPEG video
PCMA 1/2 [57]	yes	yes	yes	no	only for streams on physical ayer no multi–rate support

received before the next sending process starts. It is quite obvious that the retransmissions of packet number three, six, and eight increase the delay for the stored LPDUs. Moreover, if we suppose that the eleven LPDUs represent one higher **P**acket **D**ata **U**nit (*PDU*) (e.g. an IP segment), each retransmission leads to a higher transmission delay of this PDU.

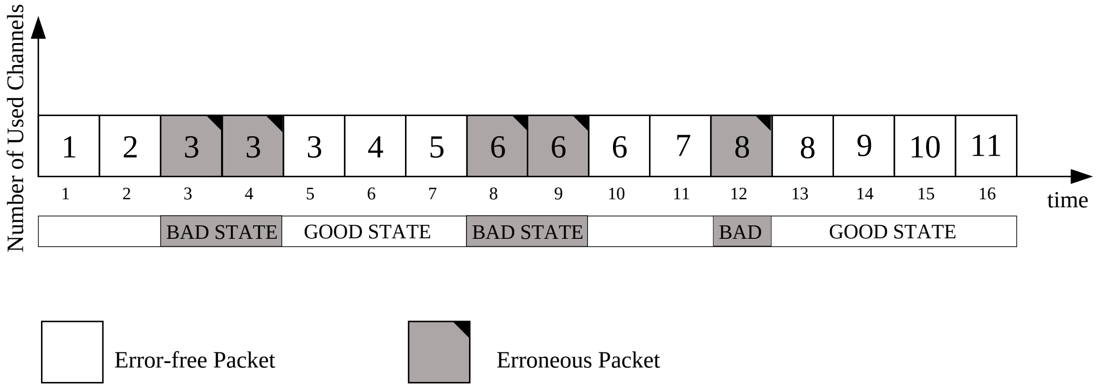


Figure 4.1: Sequential transmission mode for 11 LPDUs on the wireless link.

We conclude that due to retransmissions, the delay and the delay–jitter for a single LPDU as well as for segments of higher protocol layers increase. Highly variable throughput and an increased delay–jitter, are not acceptable for many multimedia applications as it is claimed in [86, 107] and Section 2.2. Thus, in the following we want to introduce a method that overcomes the instability of the wireless link and therefore reduces the delay and delay–variations of single LPDUs as well as for higher protocol segments. In contrast to DQRUMA, LIDA/BALI, WJSQ, MPEG–ATM the approach has to work in a distributed fashion and unlike PCMA it has to support heterogenous QoS. All in all we conclude that we want to develop a data link approach, which is simple, autonomous, distributed, and support multi–rate communication.

4.2 SMPT Approach

Using the simplest ARQ mechanism *Send and Wait*, that is discussed in [108] and suggested in recent wireless LAN standards [109, 110], each erroneous LPDU is retransmitted while following stored LPDUs have to wait until the LPDU has been transmitted successfully. The transmission of eleven LPDUs using the sequential method is shown in Figure 4.1. Because LPDUs numbered $3, 6, 8$ are prone to errors for a long time period, the stored LPDU packets have to wait until these packets have been transmitted successfully. So the effective bit rate decreases from B_{good} to B_{bad} and simultaneously the jitter increases. We assume that a resulting bit rate B_{bad} is not acceptable for the required throughput specified by the QoS parameters. Furthermore the increased jitter is not acceptable for the application.

SMPT is a method that overcomes the described problem. During bad channel conditions we compensate the bit rate degradation $\omega = B_{good} - B_{bad}$ by using multiple CDMA channels. Bundling k multiple CDMA channels will offer a bit rate of $B_{joint,k}$. In order to recognize the changes of the link quality an information feedback is needed. For the following considerations we assume a feedback channel that provides the acknowledgments right before the next scheduling cycle (before the next DLC frame). Note, the immediate acknowledgment is no SMPT feature, later on we show the impact of a delayed acknowledgment. One possible SMPT transmission method is shown in Figure 4.2. We refer to this transmission method as the *Basic* SMPT approach. The LPDUs are transmitted sequentially using one CDMA channel as long as no errors occurs. However, when a packet is corrupted no acknowledgment will be received at the sender (e.g. LPDU three is error–prone and therefore not acknowledged). In such a case, SMPT retransmits the corrupted LPDU once again. Simultaneously the next stored LPDU (number four) is transmitted over another parallel CDMA channel. If subsequent LPDUs are also erroneous new CDMA channels are allocated. This can be seen at time slot five. Henceforth we assume that in a single time slot τ_{slot} all the parallel transmitted packets will either be corrupted or received properly. The additional CDMA channels are released, if all packets, which were influenced by the error prone link, have been successfully transmitted. This is the case at time slot six.

If we compare the sequential transmission method with SMPT by means of the examples given in Figure 4.1 and 4.2, we see that: 1.) the sequential approach is able to transmit eleven LPDUs successfully in our example with five retransmissions and 2.) the SMPT approach transmits 16 LPDUs successfully with eight retransmissions. Note that the same channel error pattern is used for the sequential approach as well as for the SMPT approach.

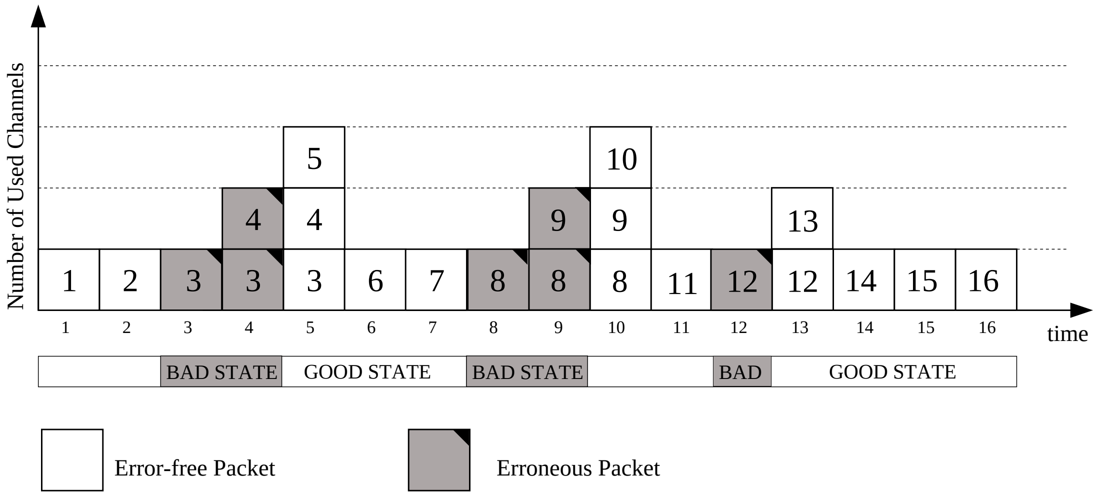


Figure 4.2: Retransmissions in code domain.

Generally, the joint bit rate $B_{joint,k}$ using k CDMA channels of bit rate B_{bad} is smaller than the sum of the bit rates of all channels, because the higher number of used channels results in a higher bit error probability [15, 111]. The degradation of bit rate allocating k simultaneously used CDMA channels is expressed by Δ_k as shown in Equation 4.1.

$$B_{joint,k} = k \cdot B_{bad} - \Delta_k \tag{4.1}$$

The effective bit rate $B_{joint,k}$ depends on Δ_k, which is composed of three terms namely $\Delta_{Codes,k}$, Δ_{ECS}, and Δ_τ. With an increasing number of allocated channels the noise level will also increase resulting in a lower effective bit rate. The effect of this parameter depends on the overall system. Therefore mechanisms are needed which protect the system against bad influence due to the SMPT approach. For example a **B**ase **S**tation (*BS*) , which has full knowledge about the dimension of the system might assign only a limited number of channels to each WT. Alternatively a WT might stop using excessive number of channels if the throughput does not improve. This leads to the assumption that an optimal number of channels per WT can be found, such that the influence on other WTs is low while on the other hand assures stability of bandwidth. When the wireless link quality is low, we can also protect the packets by additional error correction schemes as shown in Figure 9.3, 9.1 and 9.2. For example we switch on FEC in addition to ARQ algorithms, if the bit error probability increases. These error correction schemes reduce the effective bit rate per channel if the link becomes good again. Dependent on the system design, additional channels, which are required to stabilize the link quality, are not available immediately. There is a delay τ involved in establishing a new channel. τ depends on the following parameters τ_{decs}, τ_{aloc} and τ_{sync}. τ_{decs} represents the time expired before the degradation of the bit rate is noticed. This parameter depends of the delay on the feedback channel. τ_{aloc} represents the time that is needed to *allocate* a new channel. As long as MC–CDMA is assumed and the channels are managed by the WT (preassigned) the time τ_{aloc} equals to zero. For VSG systems τ_{aloc} is larger than zero. Even for an error–free signaling channel the change in the spreading gain has to be signaled and can be activated in the next time frame. For an error–prone signaling channel τ_{aloc} becomes even larger. Furthermore, if we assume a centralized channel management, τ_{aloc} increases. τ_{sync} reflects the *synchronization* time on the channels. Our assumptions about the CDMA system let τ_{sync} reduce to zero, because the synchronization for all additional channels was already taken care of by the first channel. Hence, $\tau = \tau_{decs} + \tau_{aloc} + \tau_{sync}$. The influence of τ in the resulting bit rate is denoted by Δ_τ. Summarizing the aspects described above a resulting bit rate $B_{joint,k}$ is given in Equation 4.2.

$$B_{joint,k} = k \cdot B_{bad} - \underbrace{\Delta_{Codes,k} - \Delta_{ECS} - \Delta_\tau}_{\Delta_k} \tag{4.2}$$

It cannot always be assured, that $B_{joint,k}$ will reach the required bit rate B_{good}. This depends on the design parameters of the wireless system and used CDMA technologies. In the following work we concentrate on ARQ schemes, which lead to $\Delta_{ECS} = 0$. Furthermore we assume Δ_τ to be zero for all investigations. Thus, the main degradation for $B_{joint,k}$ stem from $\Delta_{Codes,k}$.

4.3 SMPT for Multimedia Services

In this section we give a short definition of delay and jitter as given in RFC 1193 [107]. Afterwards one possible scheme to gather transport layer information at the data link layer, which would help to optimize the SMPT transmission approaches is given. The gathered information, if available, can be used to achieve higher capacity in the cell. To achieve higher capacity an appropriate abort criteria has to be found for higher protocol segments.

4.3.1 Definition of Delay and Jitter

Depending on the application and the under–lying transport protocol each network–oriented message or IP segment has certain delay and jitter constraints. For example the sender–side transport entity passes one segment to the IP–based network layer at time t_0. If an error–free sequential transmission is assumed, the segment will arrive at the receiver–side network layer at time $t_1(L_{\text{IP}})$, which depends on the segment length L_{IP} and the bit rate. For transmission over the wireless link the segment will be divided into N LPDUs (DLC packets). Each DLC packet can be transmitted in one time slot of length τ_{TDD}. Thus, under the assumption that only one LPDU can be send in τ_{TDD} the minimal delay is given by

$$D_{\text{min}}(L_{\text{IP}}) = t_1(L_{\text{IP}}) - t_0 = N \cdot \tau_{\text{TDD}}. \tag{4.3}$$

Because of the error–prone wireless link, retransmissions of corrupted packets are required, which increase the segment's delay. Suppose that the receiver–side transport entity accepts only segments that arrive with a delay not higher than a maximum permissible delay

$$D_{max} = \min\{\tau_{\text{delay}}, D_{\text{min}}(L_{\text{IP}}) + \tau_{\text{jitter}}\} \tag{4.4}$$

where τ_{delay} denotes the deterministic delay bound, and τ_{jitter} denotes the deterministic delay–jitter bound as it is defined in [107], we define D_{max} as the **T**ransmission **W**indow (*TW*) in which a segment has to be transmitted successfully from the sender to the receiver. The delay D_i of segment i is given by

$$D_i = t_2 - t_0 \leq \tau_{\text{delay}}, \tag{4.5}$$

where t_2 reflects the real arrival time of the LPDU. And the jitter J_i of segment i can be described as

$$J_i = t_2 - t_1(L_{\text{IP}}) \leq \tau_{\text{jitter}}. \tag{4.6}$$

τ_{jitter} and τ_{delay} are given by the application or transport layer requirement. This information can be gathered from the network layer segment (e.g. evaluating the RTP header). If a segment arrives at the receiver–side at time t_2, such that $D_i \leq D_{max}$, the segment was transmitted successfully. For successfully transmitted segments the delay–jitter J_i is measured. Note, that the delay–jitter of successful segments can never be larger than τ_{jitter}. In Figure 4.3 the terms D_{min}, D_i, D_{max}, J_i, and J_{max} are given for an example of an error–free and an error–prone transmission of one segment consisting of eight packets. In the specific example $J_{max}+D_{min}$ is smaller than D_{max}, because of the segment length of eight. For this example J_{max} is the determining factor. If the segment would be three packets larger the determining factor would be D_{max}.

4.3.2 Gathering Transport Layer Information at the Link Layer

In this section we demonstrate how layer four information of transport layers with **T**ransport **C**ontrol **P**rotocol (*TCP*) and **U**ser **D**atagram **P**rotocol (*UDP*) can be exploited in order

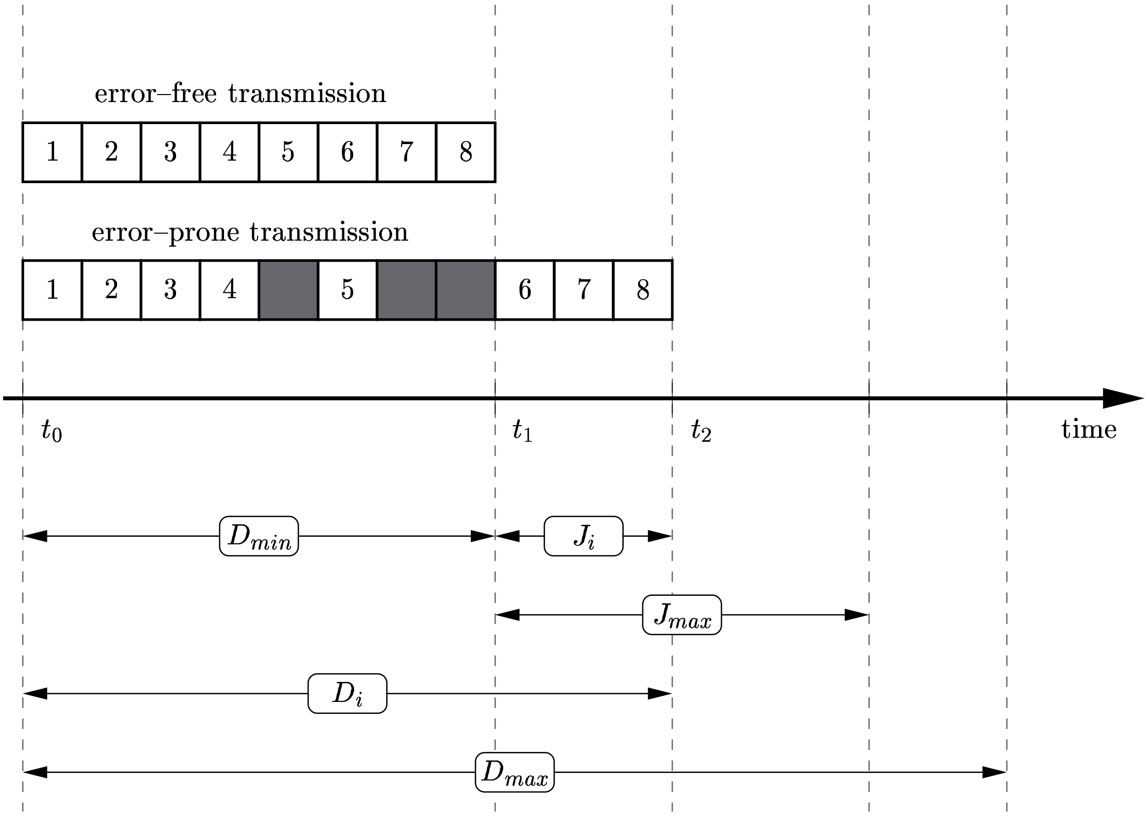


Figure 4.3: Definition of maximum delay and maximum jitter for a given network segment.

to influence layer two mechanisms at the wireless host and at the base station. We presented this work in [112]. As mentioned before TCP performance degrades for increasing delay variation of the segments. In order to investigate improvements of TCP performance by SMPT we distinguish the following two cases: For the WT acting as transmitter, the data link layer may obtain knowledge of the **R**etransmission **T**ime **O**ut (*RTO*) timer. If the data link layer has full knowledge of this timer[1] it can dimension the layer two transmission window for transmitting the packets of one corresponding TCP segment. Figure 4.4 shows the approach how the DLC layer will adjust its TW under consideration of the calculated RTO after the first measured **R**ound **T**rip **T**ime (*RTT*) timer. Because of the missing a priori knowledge for the time duration T_{Fix} and T_D[2], T_U will be initially set to $T_U = RTO_{estimated} - \max\{T_{Fix}\} - max\{T_D\}$. $max\{T_{Fix}\}$ and $max\{T_D\}$ are known from the previous measurements. In the case of WT terminated TCP connections the following alternatives for dimensioning of TW can be distinguished: using a fixed window or dynamically adjusting the TW based on an estimation of the RTO timer of the transmitter. Estimation of the RTO timer could be performed in the wireless terminal, or in the base station. As an estimation of the RTO timer is not trivial, it appears to be a suitable approach to exploit the performance improvement possible by using a fixed

[1] This knowledge can be obtained either by accessing TCP state, or by recalculation in the DLC layer.

[2] As given in Figure 4.4 T_{Fix} represents the time of the TCP segment in the backbone (after the base station) and T_D is the time assumed for the downlink transmission of the TCP segment.

layer 2 transmission window (TW).

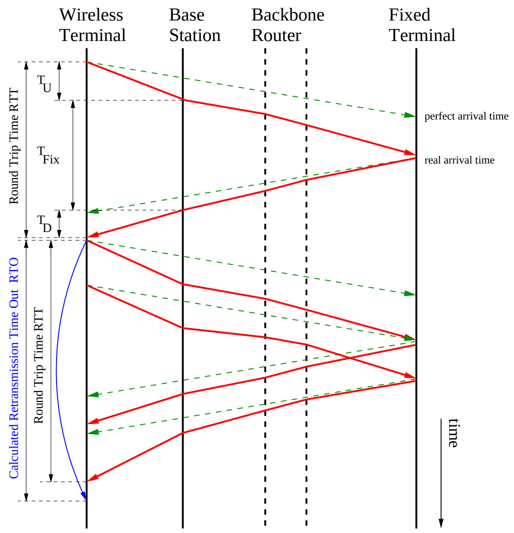


Figure 4.4: Message sequence chart for calculation of the retransmission time out.

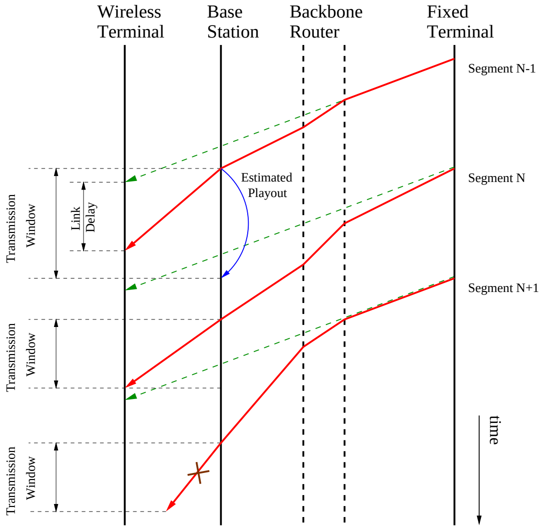


Figure 4.5: Message sequence chart for RTP traffic flow evaluating time stamps.

For RTP/UDP we differentiate between the uplink and the downlink case: In the case of WT oriented RTP connections a transport layer information in the wireless terminal may be used for scheduling at layer two. A fixed TW is used for stabilizing delay in the transmission over the wireless link. For the case of WT terminated RTP connections we consider the following alternatives: using a fixed TW, controlling TW by functions located in the wireless terminal, in the base station, or a combination of both. In the WT, filtering can be used to detect RTP segments upon arrival of the first MAC packets. The SMPT layer of the wireless terminal can determine a suitable TW, based on RTP timestamp information (which is also available before complete reception of the RTP segment), together with information on the delay budget for the RTP flow and the actual arrival time (see Figure 4.5). The value for TW determined by the WT is signaled to the base station. Performance improvements can be obtained by stopping the transmission of RTP packets that would arrive too late at the wireless terminal, and by signaling to the base station the TW which is most suitable for a given segment, and for which segments the service quality benefits significantly from using parallel communication channels for their transmission. Under certain conditions (e.g. no IPSEC is used), an alternative scheme can be used, where a transparent RTP agent is located at the base station. This agent contains informations about RTP connections by collecting state information for a number of RTP sessions up to a given maximum. RTP timestamp information, together with the actual arrival time of a packet can be used to dimension the TW for SMPT dynamically. By evaluating time stamps, information can be obtained about whether an individual packet is delayed more or less than the average delay calculated over a given window. Segments that arrive at the base station with a small delay have more time to be transmitted over the wireless link (allowing a larger TW), while segments with a large delay have to be transmitted using several CDMA channels in parallel. The RTP agent can be described as a QoS balancer, improving last hop performance of RTP packets that have been delayed higher than average in the fixed network, at the cost of RTP packets that have been delayed less. The advantages of this approach will be shown in one of the next chapters by simulations. Henceforth we assume that we have perfect knowledge of τ_{jitter} and τ_{delay}.

4.3.3 Abort Criteria

If an exploitation of the delay and jitter requirements of higher protocol layers is feasible, the sender–side transmitter will abort transmission of segments that fail meeting the delay or jitter requirements. This will help to maximize the SNIR value in the CDMA cell. The sender–side data link entity aborts transmission of LPDUs of a segment, if the segment is likely to fail to meet the pre–specified delay bounds. All corresponding packets are removed from the queue and the segment is lost. Even if for the aborting WT the segment is lost, this procedure leads to an improved overall system performance. The sender–side data link entity will proceed with the next stored segment. The question is now when a WT has to abort the transmission of a segment. For the sequential approach this is easy. The number of remaining LPDUs of one segment has to be smaller than the number remaining DLC frames for a given transmission window. In case of the sequential transmission the remaining DLC frames reflect the remaining capacity. For SMPT the remaining capacity is not that easy to calculate. Figure 4.6 shows the problem of calculating the capacity. While the number of remaining DLC frames is given by the transmission window, the number of parallel channels depends on the error pattern for this time period and the SMPT mechanism.

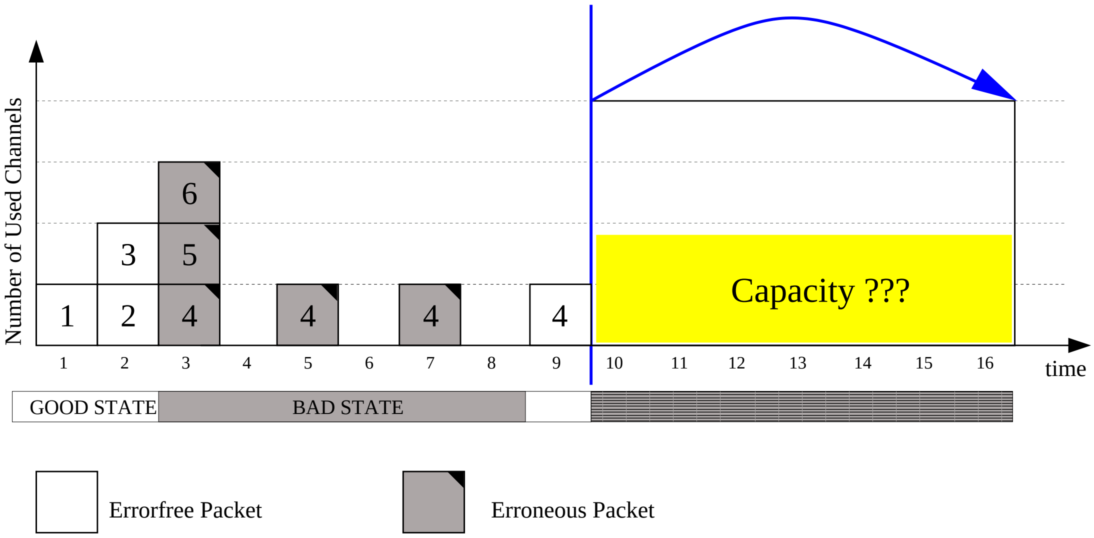


Figure 4.6: Abort criteria and the calculation of the remaining capacity.

In order to perform a fair comparison of the studied SMPT approaches and the sequential transmission mechanism, we fix a basis to decide whether to abort or to continue the transmission of the LPDUs of a given segment. If the strictly sequential transmission (using only one CDMA code channel) of the remaining LPDUs over an error–free channel is not able to meet the segment's delay bound, the transmission is aborted. We note, that for the different SMPT approaches further QoS improvements can be achieved by optimizing the abort criterion. If all LPDUs of a segment are transmitted successfully, the receiving data link layer reassembles the LPDUs to the segment. The successfully received segment is passed immediately to the application via the network and transport layers.

4.4 Enhanced SMPT Mechanisms Considering Correlated Link Errors

We assumed independent random packet errors on the wireless link so far. However, fading channels often show dependent packet losses. For such scenarios the presented *Basic* SMPT approach leads to inefficient channel usage. Figure 4.7 shows the *Basic* SMPT approach with a maximum number of four parallel channels. In case of *correlated* errors (time slot number three up to eight are error–prone and represent the *bad* channel state) we notice that several channels are used without success leading to a further degradation of the SNIR for all WTs in the cell. It has to be noticed that either all multiple channels are error–prone or none. Thus, we have to develop mechanisms that can deal with correlated packet errors under consideration of the total SNIR. The best case scenario would be to do not send any packet when the channel is in the bad state. Even though the a priori knowledge is not available, the goal is to detect the bad channel state and to react on such a situation properly. We introduce four enhanced SMPT mechanisms that can deal with correlated packet errors in this section.

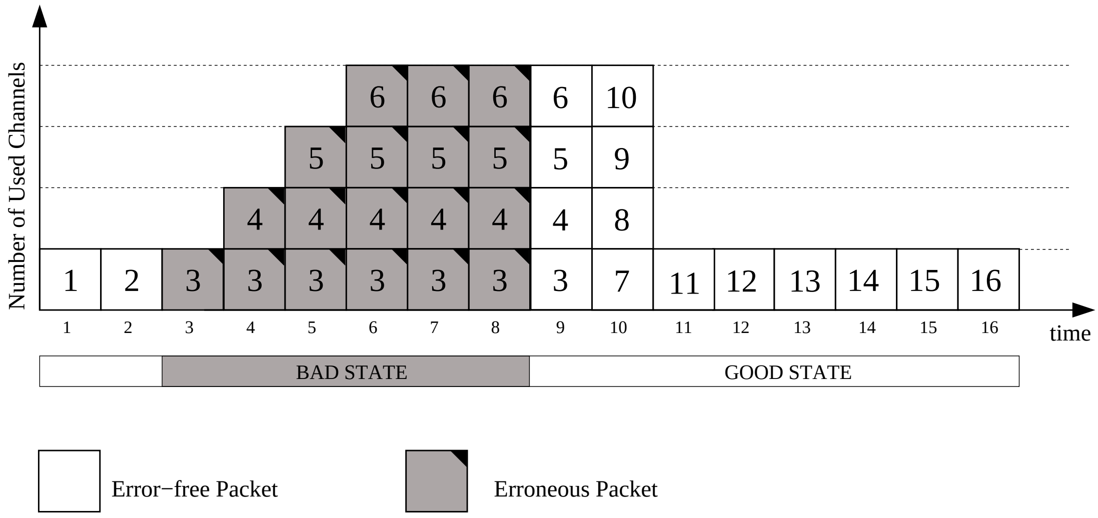


Figure 4.7: Behavior of the basic SMPT approach in presence of burst errors.

4.4.1 The Approaches

In [105] Zorzi and Rao claimed that classical ARQ strategies lead to a considerable waste of energy due to the large number of retransmissions in the outage period of the wireless channel. They introduced a new protocol which switches between *probing* and *normal* mode depending on the channel state. We have used this aspect and introduced a new enhancement to the SMPT protocol. As soon as the sender side data link entity realizes that the channel state has changed from *good* to *bad* it sends probing packets. The probing mode continues until one probing packet is transmitted successfully. The probing rate R_{probe} (number of probing packets per time slot) has an significant impact on the system performance. If the rate is too high to much channel capacity can be wasted during long *bad* channel durations. On the other side if the rate is too low the change in the channel quality is notified with a delay. Throughout this work the probing rate R_{probe} is not changed and set to one. Thus, in each time slot one probing packet is sent. The probing packet may be a *normal* LPDU (e.g. the last corrupted LPDU). We assume in this work, that a successfully transmitted packet is equivalent to an indicator of a *good* state, while a lost packet is equivalent to an indicator of a *bad* state. The work of Zorzi and Rao implicates for the following enhanced SMPT approaches that multiple channels should be used only in the *good* channel state. We will see in the following that there will still be situations in which this target cannot be guaranteed. But these situations have to be minimized. Once a *bad* channel state is detected, the protocol will probe the channel as proposed in [105]. In fact after having recognized a *good* state of the channel we have different approaches to assign additional channels. Several of these approaches are described in the following examples. We basically categorize the approaches based on the *Self Healing* and the *Start Up* mechanisms. The investigated SMPT approaches are named *Slow Start*, *Fast Start*, *Slow Healing*, and *Fast Healing.* For illustration purpose we show the transmission of 16 LPDUs representing one higher protocol layer segment in a TW of 16 time slots. Note, for all examples we assume the same error characteristic on the wireless link and the maximum number of parallel channels per WT is set to four. Figure 4.8, 4.9, 4.10, and 4.11 illustrates the different enhanced SMPT transmission strategies. We refer to Figure 4.1 for comparison with the sequential transmission.

4.4.1.1 Start Up

The *Start Up* method sends on additional channels whenever the channel is in the *good* state and there is something in the queue. The *Fast Start* is depicted in Figure 4.8. In our example this leads to a successful transmission of the first eight packets in the first two time slots. After each uplink time slot the successful transmitted LPDUs are acknowledged. When the bad phase starts at time slot three the mechanism will send four further packets (9,10,11,12). Even if the channel state is *bad*, the sender transmits on all parallel channels, because the sender has not realized the changed state of the wireless channel, yet. If no acknowledgment is received before the LPDUs are scheduled for time slot four, the sender stops the *Fast Start* phase and changes to the *Probing* phase. In the *Probing* phase only one packet (in this case LPDU nine) is retransmitted. The retransmission stops if this LPDU is transmitted successfully and an acknowledgment is received. Simultaneously the sender switches back to the *Fast Start* phase using all parallel channels.

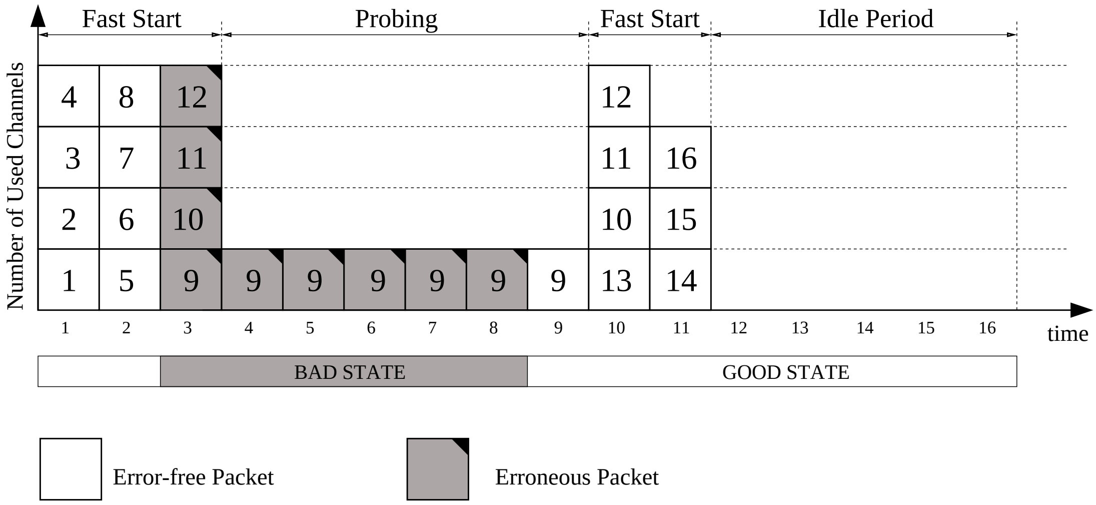


Figure 4.8: *Fast Start* SMPT mechanism.

The *Slow Start* approach is depicted in Figure 4.9. In contrast to the *Fast Start* method it will use the channels in a more moderate way. Using the *Slow Start* approach the sender starts to transmit the first LPDU. After having received the acknowledgment for this LPDU, it assumes to be still in a *good* channel state and transmit the next time two LPDUs in the next slot and so on. When the *bad* channel begins, the sender will not receive the acknowledgment for the three LPDUs. In this case, like the *Fast Start* approach, the sender switches to the *Probing* phase. After recognizing a *good* channel state again it starts to build up the ramp once again until all LPDUs are transmitted.

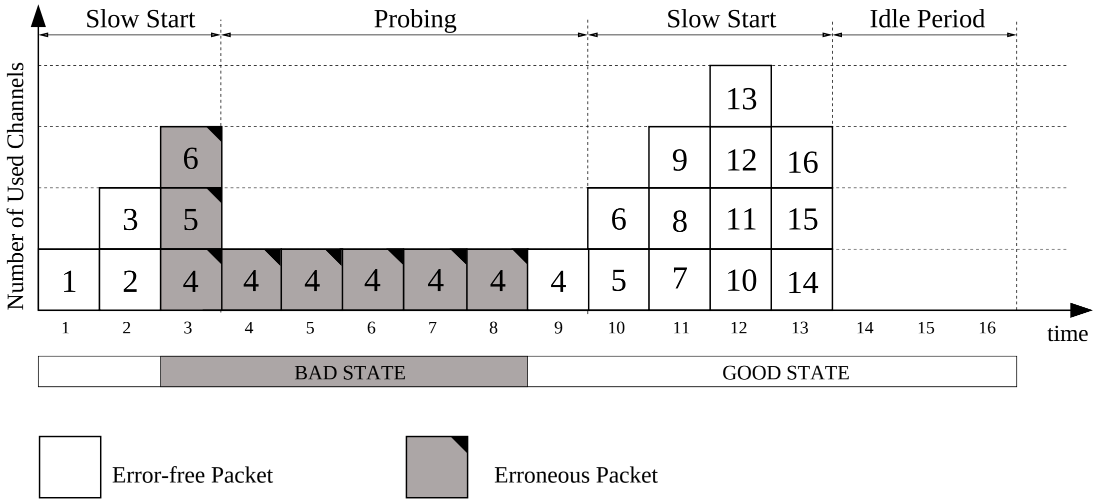


Figure 4.9: *Slow Start* SMPT mechanism.

As illustrated in these two examples the *Start Up* methods can lead to periods, during which nothing has to be sent. We refer to these periods as idle periods. While the idle periods are good for the purpose of **P**ower **S**aving (*PS*) they inherently have also one small drawback. The idle periods are achieved by sending preventive LPDUs. Therefore it is possible to achieve a negative jitter value (see definition in Section 4.3.1). For flow based transport protocols (e.g. TCP) this can lead to wrong decisions in the transport protocols. As an example we refer to the TCP timer problem. In case of *good* channel conditions the segment is transmitted much faster than with *bad* channel conditions. This variance could lead to a situation where the TCP mechanism detects congestion and decreases its transmission rate. To protect these protocols by making wrong decisions, the receiver–side data link entity passes the segment to the network layer with a certain delay, which is set to the idle period.

The main advantage of both approaches would be to successfully transmit higher protocol segments in the tolerable **T**ransmission **W**indow (*TW*) , when only a few WTS are in the cell. However, in the *Fast Start* approach the aggressive method using all possible parallel channels ignores the channel conditions and hence could be harmful to itself as well as to the other WTs in the same cell. This drawback was solved by the *Slow Start* method. Increasing the parallel channels incrementally allows to *plumb* the channel conditions. In case that too many channels were used leading to a situation where no communication is possible the *Slow Start* method will immediately switch back to the usage of the initial channel. Note, both approaches depend heavily on a perfect feedback channel with no delay. In Section 4.4.3 we will illustrate the impact of a perfect but delayed feedback channel.

4.4.1.2 Self Healing

In contrast to the *Start Up* approach, *Self Healing* is used only if the sequential transmission (using one channel) falls behind due to channel errors. The *Self Healing* mechanism reduces the accumulated delay–jitter using the capability to send on additional channels. There are two main approaches of the self healing process. The *Fast Healing* (see Figure 4.10) approach uses all available resources immediately after detecting the *good* channel state, while the *Slow Healing* process uses the resources incrementally (see Figure 4.11). For both approaches, we assume that only one channel is used after any error on the wireless link is detected.

The goal of the *Self Healing* SMPT mechanisms is to *repair* the accumulated delay–jitter always after an error–prone link. It will never send LPDUs preventively like it is done with *Start Up* approach. Therefore no idle periods exist and no protection for flow based transport protocols has to be taken into consideration. It seems that the *Slow Healing* approach has the best capability to adapt itself to the conditions in the wireless cell. The example in Figure 4.11 shows a successful *repair* of the jitter. In case an error occurs while building up the ramp the number of channels is reset immediately to one. This behavior is similar to the TCP fairness approach, where each terminal reduces its bandwidth after a congestion and slowly increases it afterwards.

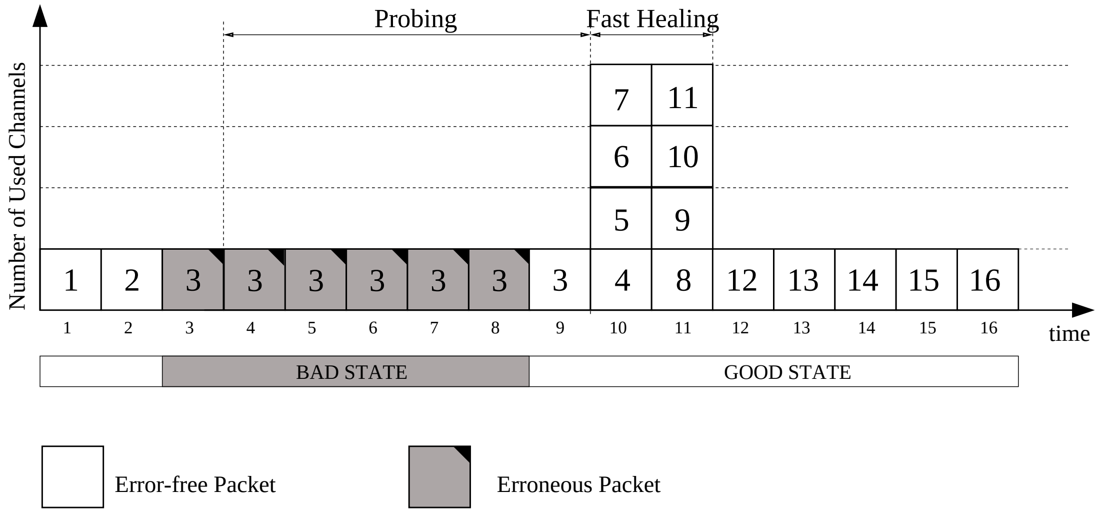


Figure 4.10: *Fast Healing* SMPT mechanism.

We want to point out that in all the above examples the delay [3] and the delay–jitter is zero for the higher protocol layer segments (composed out of 16 LPDUs).

4.4.2 Impact of the Channel Acquisition Delay

So far we assumed that parallel channels can be switched on at any time they are needed. The goal of this section is now to show the impact of the channel acquisition delay. The channel acquisition delay is the time needed to switch on one or a set of parallel channels (denoted with Δ_τ in Section 4.2). For illustration purpose we take the *Slow Healing* mechanism and assume an acquisition delay of two DLC–frame times. The impact is given in Figure 4.12. Up to time slot nine there is no difference in the transmission strategy between the *Slow Healing* mechanism given in Figure 4.11) (no acquisition delay) and Figure 4.12. After the first successful transmission of the probing packet the *Slow Healing* phase starts and the SMPT mechanisms tries to transmit on a second channel. Because of the acquisition delay the second channel is available for the first time at time slot twelve. Meanwhile (time slot ten and eleven) the mechanism is transmitting sequentially. But even with a second channel the jitter can not be healed and therefore a third channel is required. The third channel is available at time slot 14. Sending twice with three channels in parallel the effect of the bad channel duration from time slot three up to time slot eight could be fully healed. In conclusion we see that the acquisition delay prolongates the *Slow Healing* phase. The effect of this behavior has to be evaluated. A performance evaluation for impact of the channel acquisition delay is given by simulations in Section 8.2.1.4.

[3] Assuming that negative delay and delay–jitter values can be corrected to zero.

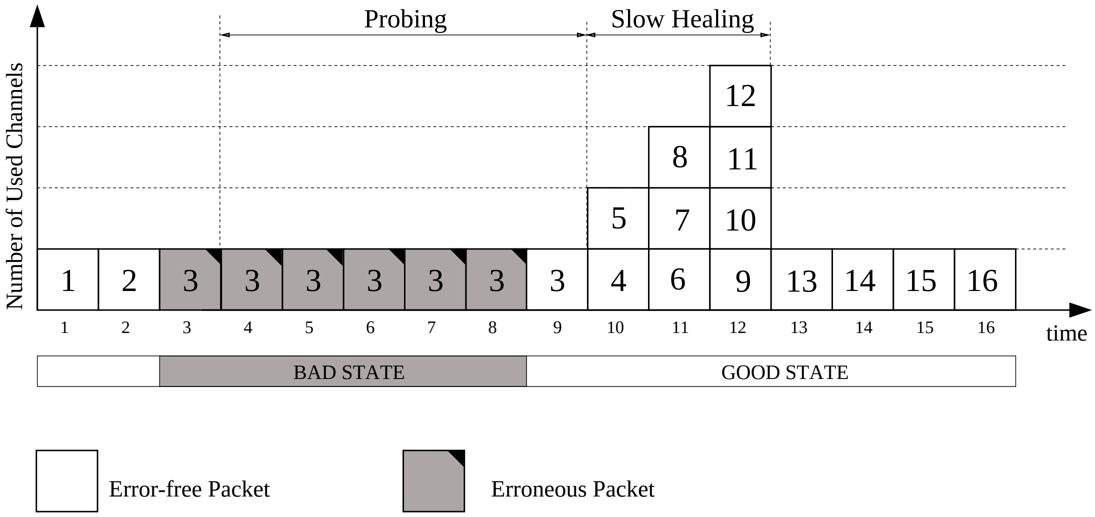


Figure 4.11: *Slow Healing* SMPT mechanism.

4.4.3 Impact of the Feedback Channel

We have already mentioned in Section 4.4.1 that the delay of the feedback channel will have an impact on the performance of the different SMPT strategies. Here we show up the situation were the delay of the feedback channel is chosen to be two time slots. Two time slots appear to be the maximum value for a delayed feedback channel. For illustration purpose we focus on only two strategies namely *Fast Start* and *Slow Healing*. The error pattern on the wireless link in all Figures is kept as in the examples before in Section 4.4.1.

Figure 4.13, shows the *Fast Start* mechanism. The mechanism starts transmitting on four channels. The mechanism shows no difference to the ideal feedback example in the first three time slots. In contrast to the previous *Fast Start* example the bad channel condition is not recognized immediately and therefore all channels are used in the fourth time slot as well. In the next slot no LPDU is sent, because the whole segment is assumed to be sent. The bad channel is recognized at the beginning of the sixth time slot since the acknowledgments for the LPDUs (9,10,11,12) are missing. In such a situation the mechanism switches to the *Probing* mode transmitting only one LPDU. The probing mode is maintained until the first LPDU is acknowledged. Because the sender has no knowledge, when the channel becomes *good* again, it periodically sends LPDU nine, ten, and eleven like it would be done using *Selective Repeat* ARQ. This procedure will increase the performance of the SMPT mechanism in contrast to send constantly packet number nine like we did in the preceding examples. Before time slot 12 the change in channel condition is recognized. The *Probing* phase is stopped and the sender transmits on all four channels again following the *Fast Start* approach. Obviously the idle time shrinks from five to two time slots for this example. Furthermore the number of error–prone transmissions increases.

Figure 4.14 depicts the *Slow Healing* mechanism. The impact of the delayed feedback channel is the reason that the bad channel state is first recognized only after time slot five. The sender goes to the *Probing* mode and repeats the first unacknowledged LPDU (in this

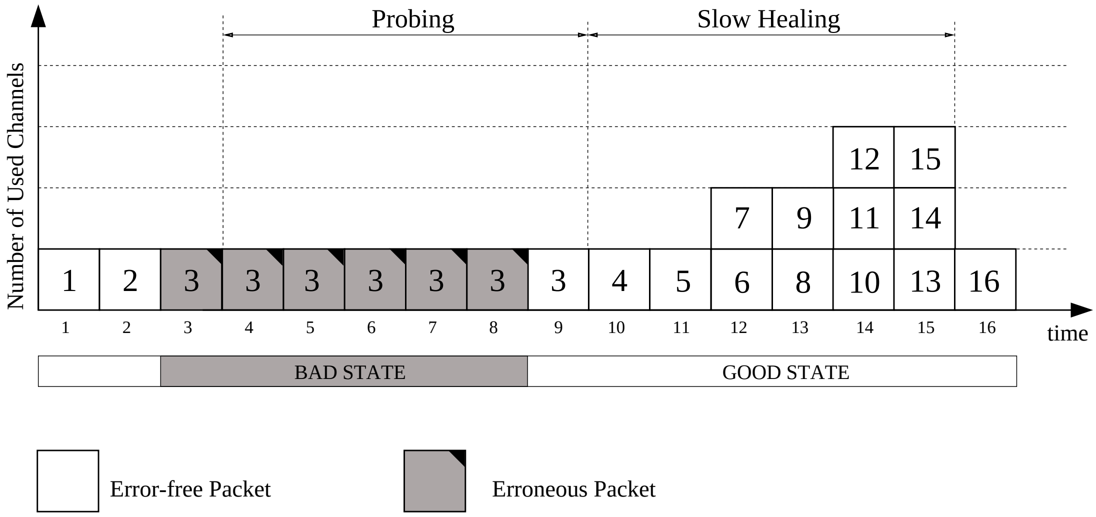


Figure 4.12: *Slow Healing* SMPT mechanism with a signaling delay of two DLC–frame times.

case LPDU number three). The status of LPDU four and five (whether they were received successfully or not) is not given at this time instant at sender–side. Having not received the acknowledgment of LPDU four and five these LPDUs are used to probe further. Even if the channel becomes *good* again at time instant nine, the sender will end its *Probing* phase first after time slot eleven, because the acknowledgment for LPDU transmitted at time slot nine was received. Now, the sender assumes that the channel condition is *good* and starts to build up the ramp. If we compare Figure 4.13 with Figure 4.8, we see that for this specific example the number of error–prone transmissions, the number of parallel used channels, and the length of the TW is the same. This suggests that for error bursts at the beginning of the segment the impact of the delayed feedback can be neglected. Thus, we claim that the *Slow Start* method is less vulnerable to the delayed feedback channel than the *Fast Start* method. We will demonstrate this by simulations.

4.5 Code Distribution and Signaling

We mentioned earlier that the code (or spreading sequences) distribution among the WTs has a great impact on the system performance. Hence, we now focus on the code distribution mechanism among the WTs. In Section 3.4.2 we introduced DQRUMA, a centralized approach where the codes were under the full control of the base station. In contrast to DQRUMA our design goal for SMPT was to build a distributed protocol. Therefore there is no special need for a centralized entity. We distinguish the following two cases:

1. Each channel is described by a unique PN sequence (*random sequences*) or
2. Each WT is assigned a PN sequence and the channels of one WT can be distinguished from the other by using a set of orthogonal codes (*random subspaces*). The

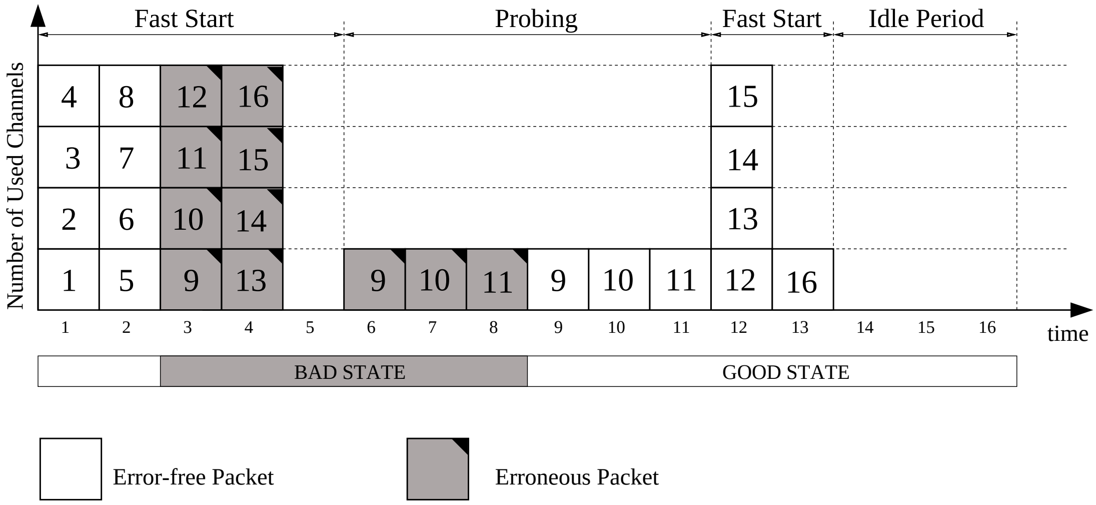


Figure 4.13: *Fast Start* SMPT mechanism with delayed feedback channel of two time slots.

advantages of these two approaches are given in Section 2.3.3.1.

In cellular communication system the WT will register at the base station using a common know resource (e.g. **R**andom **A**ccess **CH**annel (*RACH*) or **N**etwork **A**ccess and **C**onnectivity **Ch**annel). Doing so the BS will acknowledge the request and simultaneously provide a code sequence for the possible data transmission. In Figure 4.15 the **M**essage **S**equence **C**hart (*MSC*) for the code distribution mechanism is given. First the WT sends the *registration* request. The BS provides an acknowledgment and piggyback the code sequence information. Note, that it could also be possible that the WT suggest a code sequence.

In the first case, where only PN–sequences are used, the BS provides a set of codes dependent on the maximal number of usable channels per WT. For the second case the BS provides only one code sequence to distinguish between the WTs. Each WT has its own set of orthogonal codes. In combination of the set of orthogonal codes and the provided PN sequence the WT can create its own sequences. In both cases the signaling occurs only at the beginning of the combination.

Only for the *controlled overlapping* an increased signaling has to be taken under consideration. For an optimal SNIR value the spreading sequences have to be adjusted every time the distribution of active channels changes. To decrease the amount of signaling it might be reasonable to allow small degradations in the SNIR value before new spreading sequences have to be announced through signaling.

4.6 SMPT System Design

Figure 4.16 shows system design of a SMPT sender. The source generates an information stream. This information stream is segmented into LPDUs. A Header is added to each LPDU. The header contains error control and reordering informations. All LPDUs are stored in a limited buffer. The information about the QoS requirements in terms of delay

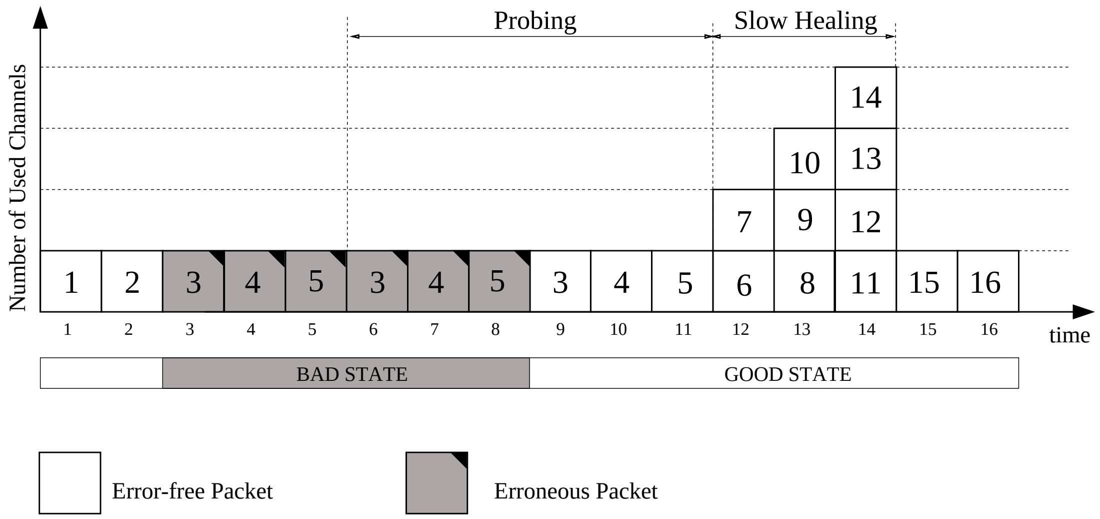


Figure 4.14: *Slow Healing* SMPT mechanism with delayed feedback channel of two time slots.

and delay–variation is given by the source itself or by the segmentation entity to the SMPT scheduler (see the gray dashed control flow). If the QoS requirements are given by the source they are certainly more precise. Even though if the segmentation entity provides *only* the information for the SMPT scheduler about the service type (e.g. TCP or UDP based) and the information at which time the segment has entered the DLC layer, a performance gain can be achieved (see Chapter 8). In Section 4.3.2 we give an insight of how we provide more detailed information to the SMPT scheduler.

The SMPT scheduler, following the mechanisms explained in Section 4.2 and 4.4 transmits the LPDUs using multiple CDMA channels. In Figure 4.16 we assume CDMA channels of the same WT are separated with a set of orthogonal codes and the WT is identified by an unique PN–sequence (see Section 2.3.3.1). We refer to this scheme as *Random Subspaces.*

At the receiver side (see Figure 4.17) this procedure is repeated inversely and the SMPT receiver reorders the LPDUs. All successfully received LPDUs are stored in a buffer. If all LPDUs of one appropriate segment have been stored, the reassembling entity rebuilds the segment and passes it to the higher protocol layer. In Figure 4.18 the interaction between BS and WTs is given.

In general a CDMA radio front-end can support several channels in parallel. It is obvious that there exists a platform dependent number of channels which can be used in parallel by one WT. Depending on the system properties this maximum number can even equal one. As an example for the second generation of mobile communication systems we describe the features of the IS-95 air interface according to the new IS-95B [102] standard. For a more detailed treatment of the IS-95A standard and theoretical analysis of IS-95 air interface, we refer interested readers to [113, 114]. The IS-95 air interface standard, after the first revision in 1995, was titled IS-95A [101]. The standard specifies the air interface for cellular, 800MHz frequency band. ANSI J-STD-008 specifies the PCS version (i.e. , the air interface for 1900MHz). TSB74 specifies the Rate Set 2 (14.4Kbit/s)

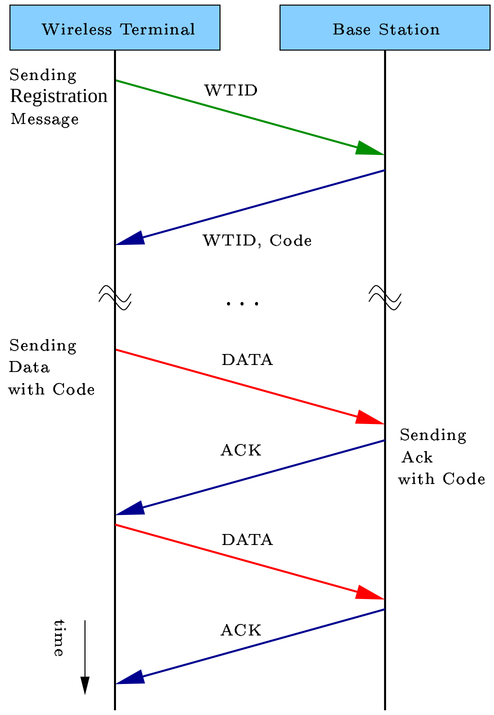


Figure 4.15: Message sequence chart for code distribution mechanism.

standard. IS-95B merges the IS-95A, ANSI J-STD-008 (PCS version for 1900MHz) [115], and TSB74 (14.4Kbit/s) standards. Furthermore, it specifies the high-speed data operation using up to eight parallel traffic channels, resulting in a maximum bit rate of 115.2Kbit/s. A traffic channel consists of a single fundamental code channel and seven supplemental code channels. Data transmitted on the uplink channels are grouped into 20ms frames. As an example for third generation mobile communication systems we have chosen the **U**niversal **M**obile **T**elecommunications **S**ystem (*UMTS*). The UMTS **W**ideband **C**ode **D**ivision **M**ultiple **A**ccess (WCDMA) standard for Europe and Japan adapts data rates by employing code aggregation in conjunction with variable spreading gains and code puncturing [73]. The **U**MTS **T**errestial **R**adio **A**ccess (*UTRA*) has two

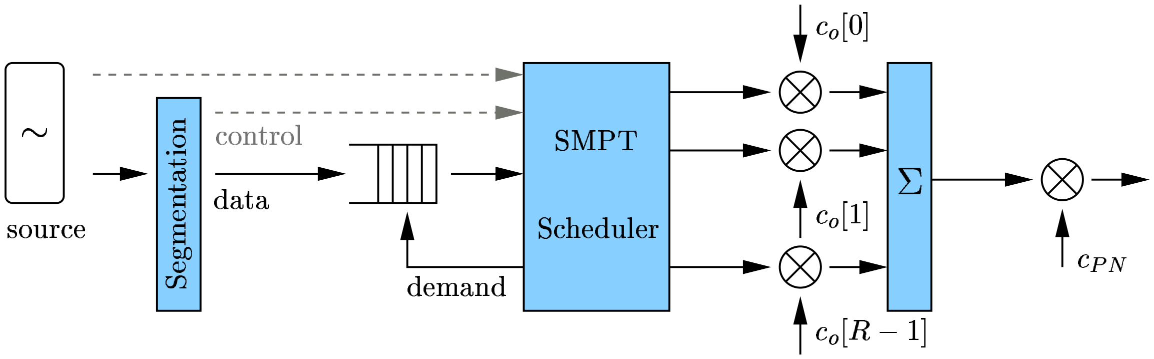


Figure 4.16: System design of SMPT sender.

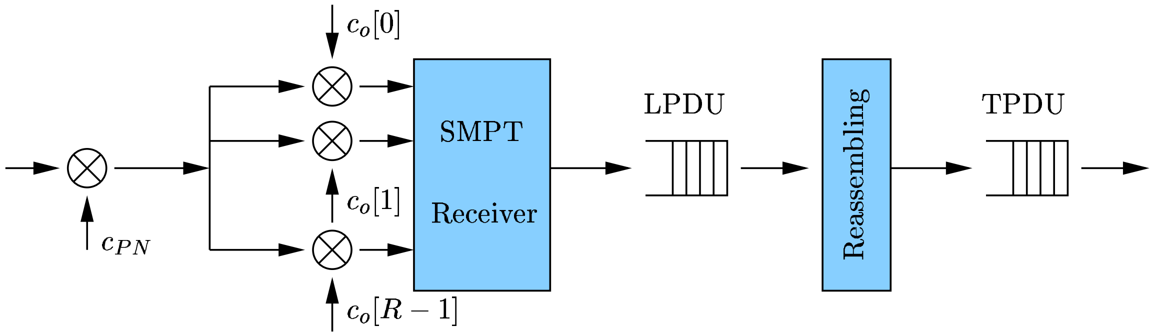


Figure 4.17: System design of SMPT receiver.

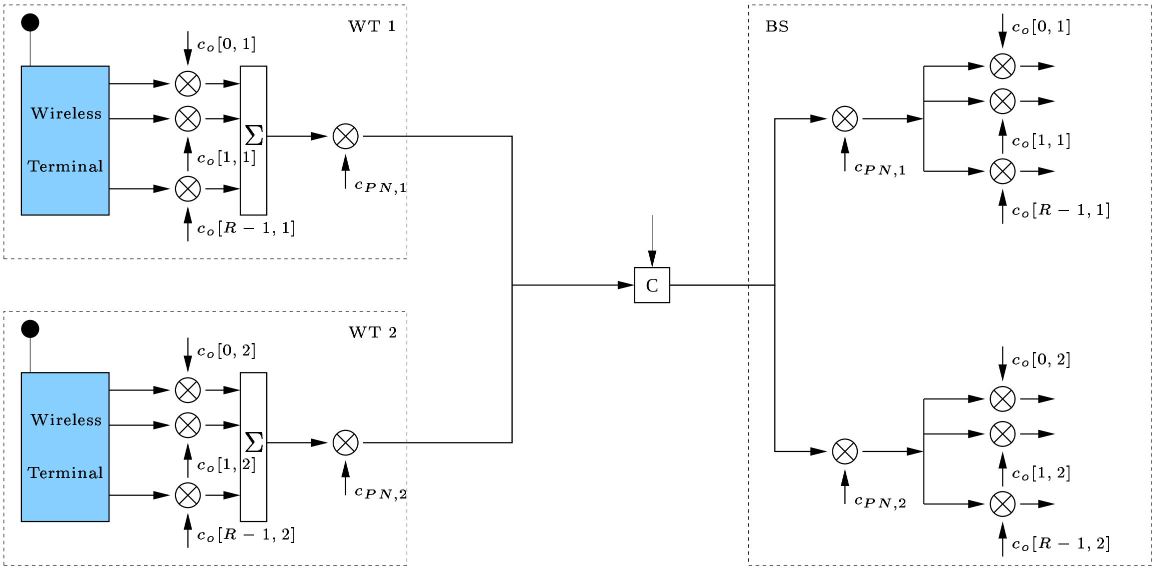


Figure 4.18: System design of SMPT sender and receiver.

modes: **F**requency **D**ivision **D**uplex (*FDD*)–mode and **T**ime **D**ivision **D**uplex (*TDD*)–mode [116]. The frame length for both modes is set to 10ms. In the TDD mode time is divided into 10 msec frames, which are subdivided into 16 mini–slots of 625 μsec each. A mini–slot is spread with a unique code and may carry either forward or backward traffic. Each mini–slot has a total of 15 code channels, which may be dynamically assigned to the individual clients. So far it is not clear how many channels will be provided by one WT. Even if 16 channels are possible, low cost WT will only support a fraction of it. In the FDD mode the spreading gain between 1 and 16 can be allocated for the up–link.

5 MPEG–4 and H.263 Video Traces for Network Performance Evaluation

Networking researchers use video frame size traces for video traffic studies [117], and video traffic modeling [118, 119], as well as for the development and evaluation of protocols and mechanisms for packet–switched networks [120, 121, 122, 123, 124, 125, 126], wireless networks [127], and optical networks [128]. The cited works are just a small sample of the hundreds of works that have made use of the video traces over the last couple of years. In fact, most researches currently use the MPEG–1 encodings of Garret [129], Rose [130], Krunz etal. [131], or Feng [132, 133]. These frame size traces give the sizes (in bit or byte) for each encoded video frame. MPEG–1 is not well suited for wireless transmission because of its large bandwidth requirement. In contrast to MPEG–1, MPEG–4 and H.263 encoded video is expected to account for large portions of the traffic in future wired and wireless networks. To date, the statistical analysis of MPEG–4 and H.263 encoded video has received very little attention in the literature. There are only few studies that evaluate networking protocols and resource management schemes with MPEG–4 and H.263 encoded video. This is partly due to a lack of sufficiently long frame size traces of MPEG–4 and H.263 encoded videos. We have generated frame size traces from MPEG–4 and H.263 encodings of over 25 video sequences of 15, 20, 30, or 60 minutes length each. The video recordings differ due to the length of the original video sources. Different quality levels as well as rate and non rate adaptive video traces are under investigation. Because of the limited bandwidth on the wireless channel we investigate only video material that is based on digital frame format QCIF. In this chapter we present a summary of the results presented in [134, 46, 135].

This chapter is structured as follows. In Section 5.1 we give an overview of digital video as well as MPEG–4 and H.263 compression. In Section 5.2 we describe the generation of the frame size traces. We give an overview of the encoded video sequences and discuss the capturing of the uncompressed video information. We describe our MPEG–4 and H.263 encoding procedure in detail. In Section 5.3 we conduct a thorough statistical analysis of the generated MPEG–4 and H.263 frame size traces. In the Appendix we review the statistical methods used in the analysis of the traces. In Section 5.4 we investigate the scalability of video streams for rate and non rate adapted video. In Section 5.5 we introduce our software which allows us to make the effect of error–prone video transmission visible.

5.1 Overview of Digital Video

First, we give a brief overview of digital video (we refer the interested reader to [136] for a more detailed discussion). Let us start with an analog video signal generated by an analog video camera. The analog video signal consists of a sequence of video frames. The video frames are generated at a fixed frame rate (30 frames per second in the **N**orth American **T**elevision **S**tandard **C**ommittee (*NTSC*) format and 25 frames per second for PAL). For each video frame, the video camera scans the frame line by line (with 455 lines in NTSC and 525 in PAL). To obtain a digital video signal the analog video signal is passed to a digitizer. The digitizer samples and quantizes the analog video signal. Each sample corresponds to a **p**icture **el**ement (*pel*). The most common digital frame formats are CIF with 352x288 pels (i.e., 352 pels in the horizontal direction and 288 pels in the vertical direction), SIF with 352x240 pels, and QCIF with 176x144 pels. In all three frame formats, each video frame is divided into three components. These are the luminance component (Y), and the two chrominance components hue (U) and intensity (V). Since the human eye is less sensitive to the color information than to the luminance information, the chrominance components are sampled at a lower resolution. Typically, each chrominance component is sampled at half the resolution of the luminance component in both the horizontal and the vertical direction. (This is referred to as 4:1:1 chroma sub sampling.). In the QCIF frame format, for instance, there are 176x144 luminance samples, 88x72 hue samples, and 88x72 intensity samples in each video frame, when 4:1:1 chroma subsampling is used. Finally, each sample is quantized; typically, 8 bits are used per sample.

As an observation we notice that the YUV video format was introduced to make color TV signals backward compatible with black–and–white TV sets, which can only display the luminance (brightness) components. Computer monitors, on the other hand, use typically the RGB video format, which contains red, green, and blue components for each pel.

Before we discuss the specific features of MPEG–4 and H.263 we discuss briefly some of their common aspects. Both encoding standards employ the **D**iscrete **C**osine **T**ransformation (*DCT*) to reduce the spatial redundancy in the individual video frames. Each video frame is divided into Macro Blocks (MBs). A Macro Block consists of 16x16 samples of the luminance component and the corresponding 8x8 samples of the two chrominance components. The 16x16 samples of the luminance component are divided into four blocks of 8x8 samples each. The DCT is applied to each of the six blocks (i.e., four luminance blocks and two chrominance blocks) in the macro block. For each block the resulting DCT coefficients are quantized using an 8x8 quantization matrix, which contains the quantization step size for each DCT coefficient. The quantization matrix is obtained by multiplying a base matrix by a quantization parameter. This quantization parameter is typically used to tune the video encoding. A larger quantization parameter results in coarser quantization, which in turn results in a lower quality as well as smaller size (in bit) of the encoded video frame. The quantized DCT coefficients are finally variable–length–coded, for a more compact representation.

Both, MPEG–4 and H.263 employ predictive encoding to reduce the temporal redundancy, that is, the temporal correlation between successive video frames. A given macroblock is either intracoded (i.e., without reference to another frame) or intercoded (i.e., with reference to a preceding or succeeding frame). To intercode a given macroblock, a motion search is conducted to find the best matching 16*16 sample area in the preceding (or succeeding) frame. The difference between the macroblock and the best matching area is DCT coded, quantized, and variable–length–coded, and then transmitted along with a motion vector to the matching area.

5.1.1 Overview of MPEG–4 Video Compression

In this section we provide a brief overview of MPEG–4 video coding; we refer the reader to [137, 138, 139, 140, 136] for details. MPEG–4 provides very efficient video coding covering the range from the very low bit rates of wireless communication to bit rates and quality levels beyond **H**igh **D**efinition **TV** (*HDTV*). In contrast to the "frame–based" video coding of MPEG–1 and H.263, MPEG–4 is object based. Each scene is composed of **V**ideo **O**jects (*VOs*) that are coded individually. (If scene segmentation is not available or not useful, e.g., in very simple wireless video communication, the standard defines the entire scene as one VO.) Each VO may have several scalability layers (i.e., one base layer and one or several enhancement layers) which are referred to as **V**ideo **O**ject **L**ayers (*VOLs*) in MPEG–4 terminology. Each VOL in turn consists of an ordered sequence of snapshots in time, referred to as **V**ideo **O**ject **P**lanes (*VOPs*). For each VOP the encoder processes the shape, motion, and texture characteristics.

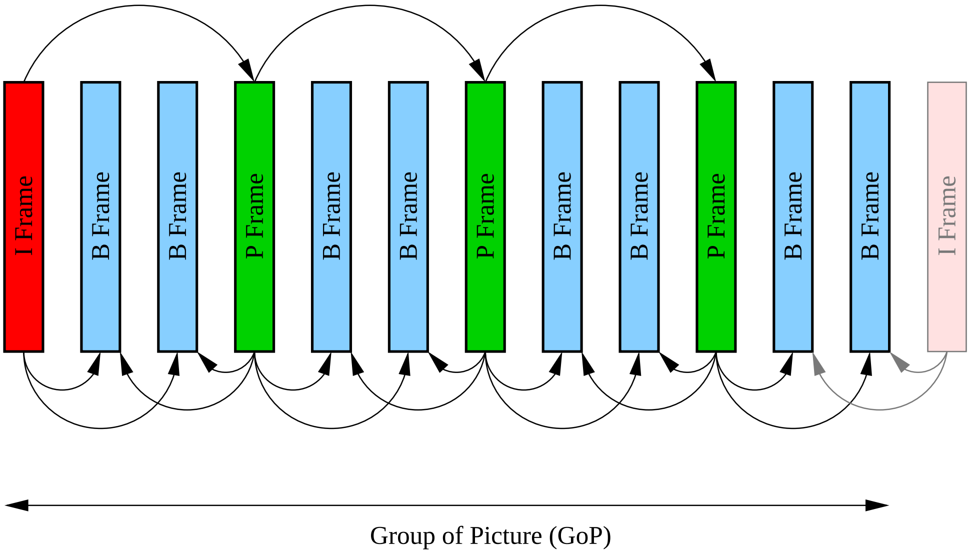


Figure 5.1: A typical GoP structure.

The shape information is encoded by bounding the VO with a rectangular box and then dividing the bounding box into Macro Blocks (MBs). Each MB is classified as lying (*i*) inside the object, (*ii*) on the object's border, or (*iii*) outside the object (but inside the bounding box). The "border" MBs are then shape coded. The texture coding is done on a per–block basis similar to the "frame–based" standards, such as MPEG–1 and H.263.

In an Intracoded (I) VOP the absolute texture values in each MB are **D**iscrete **C**osine **T**ransformation (*DCT*) coded. The DCT coefficients are then quantized and variable–length–coded. In forward Predicted (P) VOPs each MB is predicted from the closest match in the preceding I (or P) VOP using motion vectors. In Bi–directionally predicted (B) VOPs each MB is predicted from the preceding I (or P) VOP and the succeeding P (or I) VOP. The prediction errors are DCT coded, quantized, and variable–length–coded. The I, P, and B VOPs are arranged in a periodic pattern referred to as **G**roup **o**f **P**icture (*GoP*). A typical GoP structure is IBBPBBPBBPBB as given in Figure 5.1. The hierarchiacal structure of MPEG–4 encoded video streams imposes difficulty on sending this content over wireless links. In [141] it is claimed that small packet error probabilities of 3% translate into frame loss probabilities of 30%. For the transmission the shape, motion, and texture information is multiplexed at the MB level, i.e., for a given MB the shape information is transmitted first, then the motion information, and then the texture information, then the shape information of the next MB, and so on. To combat the frequent transmission errors typical for wireless communication, MPEG–4 provides a number of error resilience and error concealment features; we refer the reader to [137, 138, 139, 140, 136] for details.

5.1.2 Overview of H.263 Video Compression

The basic structure of the H.263 video source coding algorithm [142, 143, 140, 136] has been adopted from ITU–T Recommendation H.261 [144]. It uses (1) inter picture prediction to reduce the temporal redundancy and (2) **D**iscrete **C**osine **T**ransformation (*DCT*) coding of the residual prediction error to reduce the spatial redundancy. After the DCT coding, the prediction error is quantized and the resulting symbols are variable–length–coded and transmitted. For the interpicture prediction each video frame is divided into macro block and one motion vector is transmitted per macro block. In contrast to H.261, half pixel prediction is used for the motion vectors in H.263. The bit rate of the compressed video stream is controlled by adjusting several encoder parameters, such as quantizer scales and the frame rate. H.263 provides four advanced coding options. Unrestricted motion vectors, advanced prediction, and PB–frames are options that improve the inter–picture prediction. The fourth option is to use the more efficient arithmetic coding instead of variable–length–coding. These four options improve the video quality at the expense of increased video codec complexity. We refer the reader to [142, 143, 136] for details. Roughly speaking, the unrestricted motion vector option allows motion vectors to point outside the video frame. The edge pels are used instead of prediction pels that lie outside the frame. This allows for more efficient compression, especially when there is motion near the frame border and the frame format is small. With advanced prediction the motion predicted blocks overlap and a pel is interpreted as the weighted average of the overlapping blocks. This reduces artifacts in the decoded video frames and increases the perceived video quality. The PB–frames option increases the frame rate without significantly increasing the bit rate. A PB–frame consists of two consecutive frames that are encoded as one entity. Specifically, a PB–frame consists of a P–frame, which is predicted from the preceding P–frame, and a B–frame, which is bi–directionally predicted from the preceding P–frame and the P–frame being part of the PB entity. When the reconstruction of the PB–frame is complete, the B–frame is displayed first and then the P–frame.

5.2 Video Trace Generation

In this section we describe the generation of the video frame size traces. This process is illustrated in Figure 5.2, which we refer to throughout this section.

5.2.1 Overview and Capturing of Video Sequences

We played the videos [1] listed in Table 5.1 from VHS tapes using a Video Cassette Recorder (VCR). For ease of comparison with the existing MPEG–1 traces we included *Star Wars IV*, which has been MPEG–1 encoded by Garrett [129], and several of the movies that have been MPEG–1 encoded by Rose [130] in our video selection. (For a given movie, the German and English releases are sometimes edited according to different criteria and may therefore differ in the scene content. For this reason we indicate whether we encoded the German or English version.) For each video we captured the (uncompressed) YUV information with `bttvgrab` (Version 0.15.10) [145] and stored it on disk. The YUV information was grabbed at a frame rate of 25 frames/sec in the QCIF format, that is, with a luminance resolution of 176x144 picture elements (pels) and 4:1:1 chrominance subsampling at a color depth of 8 bits. We chose the QCIF format because we are particularly interested in generating traces for the evaluation of wireless networking systems. We expect that hand–held wireless devices of next–generation wireless systems will typically have a screen size that corresponds to the QCIF video format. We note that `bttvgrab` is a high–quality grabber. It is designed to not leave out a single video frame and to overcome temporal delays by buffering several frames. To avoid potential hardware problems due to buffer build–up when grabbing long video sequences, we grabbed the 60 minutes video sequences in segments of 22,501 frames ($\approx$ 15 minutes of video run time). Any 22,501 frame segment of any video gave exactly 855,398,016 Bytes of uncompressed YUV information (= 38,016 Bytes/frame). The stored YUV frame sequences were used as input for both the MPEG–4 encoder and the H.263 encoder. We emphasize that we did not encode in real–time; thus there was no encoder bottleneck.

5.2.2 Encoding Approach for MPEG–4

For each video we encoded the YUV information into an MPEG–4 bit stream with the MOMUSYS MPEG–4 video software [146], which has been adopted by MPEG in the MPEG–4 standard, Part 5 — Reference Software. We set the number of video objects to one, i.e., the entire scene is one video object. The width of the display is set to 176 pels, the height is set to 144 pels. We used a pel depth of 8 bits per pel. We did not use rate control in the encoding. The single video object was encoded into a single video object layer. We set the video object layer frame rate, i.e., the rate at which video object

[1] To avoid any conflict with copyright laws, we emphasize that all image processing, encoding, and analysis was done for scientific purposes. The encoded video sequences have no audio stream and are not publicly available. We make only the frame size traces available to researches.

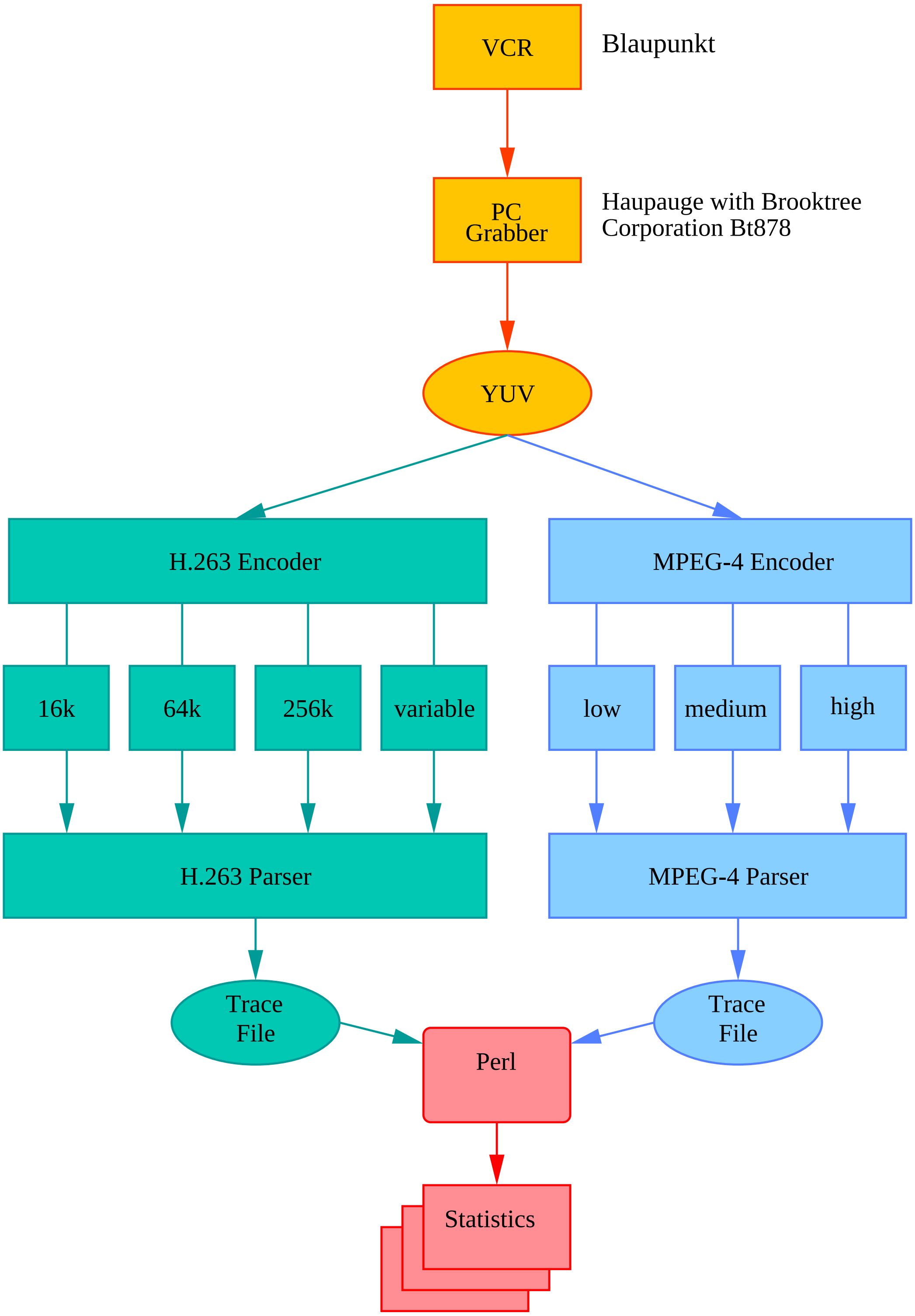


Figure 5.2: Generation of frame size traces.

Table 5.1: Overview of encoded video sequences grouped into movie, cartoons, sport events, TV sequences, and set–top

Movies (rental tapes, German/English movie versions)
Jurassic Park I (G)
Silence of the Lambs (E)
Star Wars IV (E)
Mr. Bean (G)
Star Trek: First Contact (G)
Form Dusk Till Dawn (G)
The Firm (G)
Starship Troopers (G)
Die Hard III (G)
Cartoons (rental tapes, German/English versions)
Walt Disney:Aladdin (G)
Walt Disney:Robin Hood (G)
Walt Disney:Susi und Strolch (G)
Simpsons (G)
Futurama (G)
South Park (G)
Sports Events (recorded from German cable TV)
Formula 1: Formula 1 car race
Soccer: Soccer game (European championship 1996)
Sci: Sci Alpin
TV sequences (recorded from German cable TV)
ARD News: German news (Tagesschau)
ARD Talk: German Sunday morning talk show (Presseclub)
N3 Talk: German late night show (Herman und Tietjen)
Video Clips: German VIVA Video Clips
Boulevard Bio: German late night show
Set–top
Office–Cam: Office camera observing person in front of terminal
Lecture Room–Cam: Talk about Mobile Communication by Dr. Dr. Boche
Parking–Cam: Security Cam on Campus parking

planes are generated, to 25 frames/sec. The **G**roup **o**f **P**icture (*GoP*) pattern was set to IBBPBBPBBPBB. We encoded each video at three different quality levels: *low*, *medium*, and *high*. For the low quality encoding the quantization parameters were fixed at 10 for I frames (VOPs), 14 for P frames, and 18 for B frames. For the medium quality encoding the quantization parameters for all three frame types were fixed at 10. For the high quality encoding the quantization parameters for all three frame types were fixed at 4. We refer the interested reader to the technical report [46] for a complete listing of the parameters settings used in the encodings.

We note that the MOMUSYS MPEG–4 encoder is limited to encoding segments with a length of at most 1,000 video frames. Therefore, we encoded the YUV frame sequences in segments of 960 frames (= 80 GoPs) each. When encoding a given 80 GoP segment, the last two B frames of the 80th GoP are bi–directionally predicted from the third P frame of the 80th GoP and the I frame of the 81st GoP. Since the 81st GoP is not encoded, the last two B frames of the 80th GoP are not encoded either. As a consequence our frame size files were missing two B frames per 960 encoded video frames (= 38.4 seconds of video run time). As a remedy we inserted two B frames at the end of each segment of 958 (actually encoded) frames. We set the size of the inserted B frames to the average size of the B frames in the 958 frame segment. This error could be attributed to the limitations of the MOMUSYS MPEG–4 Reference Software, can be neglected. Parameters for the encoding process with the tmn software are given in Table 5.2.

Table 5.2: Sample of the encoder parameters of the MPEG–4 MoMuSys software.

Parameter	Value	Info
Config. file for VO 1	config.dat	
Frame rate of source sequences on disk (Hz)	25.0	grabbing frequency
Hor. size of Y image on disk	176	QCIF
INTRA PERIOD	12	→ IBBPBBPBBPBB
Initial value for quantizer - INTRA	4,10,10	fixed without rate control
Initial value for quantizer - P VOPs	4,10,14	fixed without rate control
Initial value for quantizer - B VOPs	4,10,18	fixed without rate control
MPEG-4 Version	2	Version2
Number of layers	1	single layer coding
Number of VO	1	
Number of VTCs	0	
Type of rate contro	0	
VO Id	0	
VOL Id	0	
VOL frame rate (Hz)	25.0	
VOL start frame	1	
VOL end frame	960	
Value of M	3	→ IBBPBBPBBPBB
Ver. size of Y image on disk	144	QCIF
bits per pixel	8	
quantizer precision	5	

5.2.3 Encoding Approach for H.263

We encoded the uncompressed YUV information into an H.263 bit stream with the `tmn` encoder (Version 2.0) [147]. (We did not use the H.263 encoder of `bttvgrab` because of it not being fully compliant with the H.263 standard; it inserts additional sequencing and synchronization information into the H.263 bit stream.) We emphasize that we did not encode in real–time; thus there was no encoder bottleneck. We set the `tmn` encoder parameters to encode in the QCIF (176x144 pel) video format at a fixed reference frame rate of 25 frames/sec. We did not enable unrestricted motion vectors, syntax–based arithmetic coding, and advanced prediction, since we observed that these features bring only little improvement in the video quality while slowing down the encoder dramatically. We did enable PB–frames. We encoded each video at four different target bit rates: (1) 16 kbit/sec, (2) 64 kbit/sec, (3) 256 kbit/sec, and (4) Variable Bit Rate (VBR), i.e., without setting a target bit rate. Parameters for the encoding process with the tmn software are given in Table 5.3.

Table 5.3: Sample of the encoder parameters of the tmn software.

Parameter	Value
Coding format	QCIF (176x144pels)
Integer pel search window	15 pels
Quantization parameter QP	5
QP for first frame	5
Frames to skip between each encoded frame	0
Reference frame rate	25 frames/sec
USE of PB-frames	ON

5.2.4 Extracting Frame Sizes

Finally, we obtained the frame sizes (in bytes) of the individual encoded video frames by directly parsing the encoded MPEG–4 and H.263 bit streams.

We note that Ryu [148] encoded and analyzed a one hour CNN news video and a one our C–SPAN video using the MBONE video tool `vic` [149]. The `vic` tool employs an encoding scheme that is roughly equivalent to H.261. Similarly, Dolzer and Payer [150] encoded and analyzed a political talk show using the `vic` tool. In both works the `vic` packet stream was sent over a packet–switched network and the video's frame sizes were extracted from the packet time stamps using `tcpdump`. It may be argued that this indirect measurement of the frame sizes is less accurate due to the influence of the underlying packet–switched network. For this reason we extract the frame sizes directly from the encoded bit streams.

5.3 Statistical Analysis

5.3.1 Statistical Analysis of MPEG–4 Traces

In this section we conduct a thorough statistical analysis of the generated MPEG–4 frame size traces. For the analysis we introduce the following notation. Let N denote the number of video frames in a given trace. Let t denote the frame period (display time) of a given frame. Note that for almost all our MPEG–4 traces approximately $N = 90.000$ and $t = 40$ msec, which corresponds to a video runtime of about 60 minutes. Let $X_n,\ n = 1, \ldots, N$, denote the number of bits in frame n, that is, the frame size of frame n. Let G denote the number of frames per **G**roup **o**f **P**icture (*GoP*). Let $Y_m,\ m = 1, \ldots, N/G$, denote the number of bits in GoP m, that is, the size of GoP m. Clearly, $Y_m = \sum_{n=(m-1)G+1}^{mG} X_n$. For illustration Table 5.4 gives the first lines of the trace of the MPEG–4 trace file of *Star Wars IV* encoding with high quality. The trace gives on line $n,\ n = 1, \ldots, N$, the cumulative display time T_{n-1} (up to frame $n-1$), the type (I, P or PB) of frame n, and the frame size X_n in bytes.

Tables 5.5 and 5.6 give an overview of the statistical properties of the generated MPEG–4 traces. (To conserve space we give the statistics of all three quality levels only for the *Jurassic Park I*, *Silence of the Lambs*, and *Star Wars IV* videos. We refer the interested reader to the technical report [46] for the omitted results.)

The compression ratio is defined as the ratio of the size of the entire uncompressed YUV video sequence (in bit) to the size of the entire MPEG–4 compressed video sequence (in bit). The Mean $\bar{X}$ gives the average frame size. The Coefficient of Variation (defined as the standard deviation S_X of the frame size divided by the average frame size $\bar{X}$) is a typical metric for the variability of the frame sizes; the larger the coefficient of variation the more variable are the frame sizes. (We refer the reader to the Appendix for the formal definitions of the mean and the coefficient of variation.) Comparing encodings at different quality levels we observe that lower quality encoding achieves higher compression ratios, as is to be expected. Interestingly, these higher compression ratios come at the expense of increased variability of the encoded video streams. We observe that relatively high compression ratios are achieved for the *Star Wars IV* movie even for high quality encodings. This is probably due to the long scenes with dark backdrops and little contrast in this movie. For the *Formula 1* and *Soccer* videos, on the other hand, only relatively small compression ratios are achieved. These videos feature many small objects that move rapidly. This results in high mean bit rates and relatively small peak–to–mean ratios of the encoded frame sizes. Comparing the frame statistics and the GoP statistics we observe that smoothing the videos over one GoP (= 0.48 sec of video runtime) is quite effective in reducing the variability and the peak rate.

In the following we provide plots to illustrate the statistical properties of the following three MPEG–4 traces: (a) *Star Wars IV* encoded at high quality, (b) *Jurassic Park I* encoded at medium quality, and (c) *Silence of the Lambs* encoded at low quality. Figure 5.3 gives the bit rate traces, i.e., the bit rate (in bit/s) as a function of the time t.

We observe from the plots that the *Star Wars IV* encoding at high quality is relatively smooth. The *Silence of the Lambs* encoding at low quality, on the other hand, exhibits extreme changes in the frame sizes. Inspecting this trace closely, we are able to identify periods during which the frame sizes stay roughly at a fixed level; these periods appear to correspond to distinct scenes in the movie.

Figure 5.4 gives the histograms of the frame size X_n. The histogram plots reflect again the general tendency that lower quality encodings (i.e., a higher compression ratios) result in more variability of the encoded video stream. The difficulty in modeling the frame size distributions is illustrated by the histogram for the *Silence of the Lambs* encoding at low quality; it has a pronounced gap around the frame size of 60 Byte.

Figure 5.5 gives the autocorrelation coefficient $\rho_X(k)$ (see Appendix for the formal definition) of the frame size sequence $X_n,\ n = 1, \ldots, N$, as a function of the lag k (in frames). The frame size correlations exhibit a periodic spike pattern that is

Table 5.4: Excerpt of MPEG–4 trace file of *Star Wars IV* encoding with high quality.

```
Frame No.       Frametype       Time[ms]        Size [byte]
-------------------------------------------------------------
...
9               B               280             269
10              B               320             284
11              I               480             1463
12              B               400             273
13              B               440             329
14              P               600             386
15              B               520             278
16              B               560             247
17              P               720             490
18              B               640             266
19              B               680             333
20              P               840             409
21              B               760             280
22              B               800             360
23              I               960             1472
24              B               880             319
25              B               920             349
26              P               1080            424
27              B               1000            266
28              B               1040            314
29              P               1200            460
30              B               1120            327
31              B               1160            304
32              P               1320            398
33              B               1240            289
34              B               1280            293
35              I               1440            1517
36              B               1360            287
37              B               1400            314
38              P               1560            442
39              B               1480            317
40              B               1520            289
41              P               1680            338
42              B               1600            272
43              B               1640            303
44              P               1800            339
45              B               1720            327
46              B               1760            313
47              I               1920            1504
48              B               1840            329
49              B               1880            352
...
```

Table 5.5: Overview of frame statistics of MPEG–4 traces

		Compr.	Frame Size			Bit Rate	
		ratio	Mean	CoV	Peak/Mean	Mean	Peak
Quality	Trace	YUV:MP4	$\bar{X}$	$S_X/\bar{X}$	$X_{\max}/\bar{X}$	$\bar{X}/t$	$X_{\max}/t$
			[kbyte]			[Mbps]	[Mbps]
High	*Jurassic Park I*	9.92	3.8	0.59	4.37	0.77	3.3
	Silence of the Lambs	13.22	2.9	0.80	7,73	0.58	4.4
	Star Wars IV	27.62	1.4	0.66	6.81	0.28	1.9
Medium	*Jurassic Park I*	28.4	1.3	0.84	6.36	0.27	1.7
	Silence of the Lambs	43.43	0.88	1.21	13.6	0.18	2.4
	Star Wars IV	97.83	0.39	1.17	12.1	0.08	0.94
Low	*Jurassic Park I*	49.46	0.77	1.39	10.61	0.15	1.6
	Silence of the Lambs	72.01	0.53	1.66	21.39	0.11	2.3
	Star Wars IV	142.52	0.27	1.68	17.57	0.053	0.94
	Mr. Bean	66.60	0.57	1.55	13.25	0.11	1.5
	First Contact	110.94	0.34	1.50	17.08	0.069	1.2
	Formula 1	43.51	0.87	1.12	8.05	0.17	1.4
	Office–Cam	84.20	0.45	2.87	11.48	0.09	1.0
	The Firm	131.79	0.29	1.73	16.42	0.06	0.9
	Starship Troopers	61.15	0.62	1.21	11.69	0.12	1.5
	Die Hard III	49.48	0.77	1.21	10.62	0.15	1.6
	Aladdin	75.53	0.50	1.35	12.40	0.10	1.2
	Robin Hood	40.42	0.90	1.21	10.56	0.19	2.0
	Susi und Strolch	96.78	0.39	1.52	16.04	0.08	1.3
	Simpsons	32.66	1.20	1.39	17.23	0.23	4.0
	Futurerama	45.12	0.84	1.63	22.81	0.17	3.8
	Southpark	73.46	0.52	1.98	35.06	0.10	3.6
	Ski	40.65	0.94	1.12	9.76	0.19	1.8
	VIVA	38.17	1.00	1.14	9.06	0.20	1.8
	Boulevard Bio	65.95	0.58	1.74	11.16	0.12	1.3
	Lecture Room–Cam	179.62	0.21	2.29	16.22	0.04	0.7
	Parking–Cam	57.10	0.67	2.88	11.65	0.13	1.6

superimposed on a decaying slope. The periodic spike pattern reflects the repetitive GoP pattern. The large positive spikes are due to (the typically large) I frames. An I frame is followed by two (typically small) B frames, which appear as small negative spikes. The subsequent P frame (typically of mid–size) shows up as a small positive spike. The decaying slope is characteristic of the long term correlations in the encoded video. To get a clearer picture of these long term correlations we show in Figure 5.6 the autocorrelation coefficient $\rho_Y(k)$ of the GoP size sequence Y_m, $m = 1, \ldots, N/G$, as a function of the lag k (in GoPs). We observe from the figure that the GoP autocorrelation function of the *Jurassic Park I* encoding at medium quality decays roughly exponentially. This indicates that the GoP size process is memoryless. The other two curves clearly decay slower than an exponential function. This slow decay of the GoP autocorrelation is particularly pronounced for the *Silence of the Lambs* encoding at low quality, which has an autocorrelation coefficient of roughly 0.2 for a lag of 230 GoPs (approximately 110 sec).

The time–dependent statistics are important for network and traffic engineering since

Table 5.6: Overview of GoP statistics of MPEG–4 traces

		GoP Size			Bit Rate	
		Mean	CoV	Peak/Mean	Mean	Peak
Quality	Trace	$\bar{Y}$	$S_Y/\bar{Y}$	$Y_{\max}/\bar{Y}$	$\bar{Y}/(Gt)$	$Y_{\max}/(Gt)$
		[kbyte]			[Mbps]	[Mbps]
High	*Jurassic Park I*	46	0.47	3.15	0.77	2.4
	Silence of the Lambs	35	0.71	6.22	0.58	3.6
	Star Wars IV	17	0.38	4.29	0.28	1.2
Medium	*Jurassic Park I*	16	0.57	3.92	0.27	1.0
	Silence of the Lambs	11.0	0.99	10.07	0.18	1.8
	Star Wars IV	4.7	0.52	6.29	0.08	0.49
Low	*Jurassic Park I*	9.20	0.53	4.05	0.15	0.62
	Silence of the Lambs	6.30	0.92	10.48	0.11	1.10
	Star Wars IV	3.20	0.46	5.31	0.053	0.28
	Mr. Bean	6.9	0.46	5.08	0.11	0.58
	First Contact	4.1	0.54	6.03	0.069	0.41
	Formula 1	10.0	0.32	2.77	0.17	0.48
	Office–Cam	5.4	0.09	1.80	0.09	0.16
	The Firm	3.5	0.52	4.48	0.06	0.26
	Starship Troopers	7.5	0.50	4.05	0.12	0.50
	Die Hard III	9.2	0.56	4.90	0.50	0.66
	Aladdin	6.0	0.55	4.88	0.10	0.49
	Robin Hood	11.0	0.40	3.34	0.19	0.63
	Susi und Strolch	4.7	0.49	7.68	0.08	0.60
	Simpsons	14.0	0.49	7.50	0.23	1.70
	Futurerama	10.0	0.57	11.50	0.17	1.90
	Southpark	6.2	0.92	19.95	0.10	2.10
	Ski	11.0	0.47	3.14	0.19	0.59
	VIVA	12.0	0.58	4.29	0.20	0.85
	Boulevard Bio	6.9	0.43	3.26	0.12	0.38
	Lecture Room–Cam	2.5	0.20	3.40	0.04	0.14
	Parking–Cam	8.0	0.11	1.94	0.13	0.26

correlations in the video traffic can have a significant impact on the performance of packet–switched networks. Several studies [151, 152, 153, 154] have found that the losses and/or delays of queuing systems are considerably larger for positively correlated input traffic than uncorrelated input traffic. It has also been demonstrated [155] that carefully designed VBR traffic models are able to capture the relevant range of correlations and predict the system performance accurately. For these reasons it is important to analyze the long range correlations of the video traces. These long range correlations are formally characterized as self–similarity or long–range dependence (LRD) [156]. Intuitively, long–range dependent traffic is bursty (highly variable) over a wide range of time scales. *23.107: UMTS assumes that streaming and conventional classes will be* ***non*** bursty. The cumulative effect of the correlations for large lags is significant and gives rise to the large losses and/or delays found for long–range dependent traffic (even though the correlations for large lags may be individually small).

The Hurst parameter is a succinct metric for the long–range dependence (i.e., the degree of self–similarity). Generally speaking, time series without long range dependence

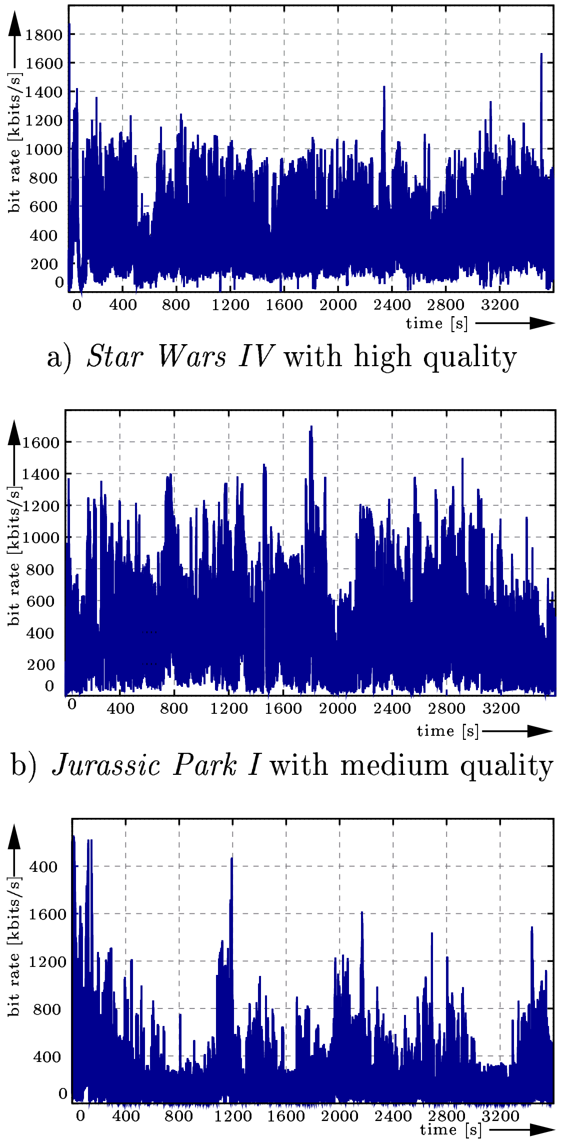
Table 5.7: Hurst parameters of MPEG–4 traces estimated from pox diagram of R/S as a function of the aggregation level a.

		Aggregation level a [frames]								
Quality	Trace	1	12	50	100	200	400	600	700	800
High	*Jurassic Park I*	0.973	0.830	0.795	0.774	0.737	0.666	0.705	0.591	0.622
	Silence of the Lambs	1.007	0.894	0.872	0.868	0.894	0.819	0.765	0.728	0.771
	Star Wars IV	0.903	0.838	0.808	0.785	0.776	0.756	0.752	0.727	0.722
Medium	*Jurassic Park I*	0.948	0.821	0.776	0.756	0.722	0.630	0.664	0.549	0.579
	Silence of the Lambs	0.997	0.891	0.867	0.866	0.896	0.849	0.781	0.736	0.760
	Star Wars IV	0.847	0.846	0.822	0.798	0.787	0.770	0.823	0.815	0.803
Low	*Jurassic Park I*	0.881	0.824	0.771	0.752	0.729	0.628	0.642	0.538	0.580
	Silence of the Lambs	0.935	0.887	0.863	0.858	0.882	0.842	0.777	0.744	0.764
	Star Wars IV	0.770	0.844	0.814	0.795	0.785	0.769	0.833	0.817	0.805
	Mr. Bean	0.816	0.871	0.823	0.824	0.813	0.796	0.750	0.790	0.791
	First Contact	0.828	0.834	0.802	0.795	0.774	0.744	0.740	0.746	0.779
	Formula 1	0.736	0.739	0.685	0.586	0.550	0.574	0.542	0.565	0.606
	Office–Cam	0.441	0.919	0.796	0.860	0.872	0.900	0.853	0.929	0.887
	The Firm	0.842	0.911	0.854	0.857	0.842	0.792	0.766	0.755	0.791
	Starship Troopers	0.871	0.873	0.843	0.830	0.798	0.764	0.745	0.712	0.726
	Die Hard III	0.904	0.895	0.892	0.877	0.839	0.849	0.915	0.874	0.835
	Aladdin	0.847	0.895	0.894	0.898	0.912	0.870	0.913	0.841	0.849
	Robin Hood	0.812	0.842	0.797	0.793	0.803	0.814	0.745	0.731	0.753
	Susi und Strolch	0.807	0.845	0.829	0.803	0.766	0.726	0.725	0.721	0.659
	Simpsons	0.769	0.886	0.877	0.851	0.837	0.784	0.791	0.830	0.717
	Futurerama	0.745	0.852	0.813	0.802	0.777	0.883	0.799	0.859	0.870
	Southpark	0.704	0.831	0.816	0.821	0.892	0.783	0.737	0.850	0.789
	Ski	0.848	0.787	0.734	0.727	0.696	0.697	0.751	0.711	0.764
	VIVA	0.904	0.916	0.943	0.954	0.937	0.884	0.849	0.823	0.835
	Boulevard Bio	0.782	0.909	0.914	0.925	0.948	1.006	1.005	0.993	1.025
	Lecture Room–Cam	0.491	0.777	0.728	0.745	0.769	0.766	0.780	0.815	0.827
	Parking–Cam	0.470	0.979	0.919	1.035	1.022	0.959	0.852	0.877	0.915

have a Hurst parameter of 0.5. Hurst parameters between 0.5 and 1 indicate long range dependence. Additionally, larger Hurst parameters indicate a higher degree of long range dependence. We estimated the Hurst parameters of the frame size traces from pox plots of the R/S statistic, as outlined in the Appendix. For each frame size trace we generated pox plots of R/S for different aggregation levels a, that is, we averaged the frame size traces over non–overlapping blocks of a frames and then plotted the pox diagram of R/S according to the algorithm given in Table 5.7. Figure 5.7 gives some pox plots of R/S for an aggregation level of $a = 1$. The Hurst parameter is estimated from the slope of the "street of points" in the pox plot. Table 4 gives the Hurst parameters of the MPEG–4 frame size traces as a function of the aggregation level a. Generally, Hurst parameters larger than 0.5 for all aggregation levels are a strong indication of long range dependence.

We estimated the Hurst parameters from pox plots of the R/S statistic as outlined in the Appendix. Figure 5.7 gives some pox plots of R/S for an aggregation level of a = 1. We observe from the table that the encodings of *Silence of the Lambs*, *Star*

Wars IV, *Mr. Bean*, *First Contact*, *From Dusk Till Dawn*, and *The Firm* have Hurst parameters larger than 0.72 for all aggregation levels. This indicates a high degree of long range dependence. The *Formula 1* and *ARD news* encodings have large Hurst parameters for aggregation levels of 50 frames and less; for aggregation levels of 200 frames and larger, however, the Hurst parameters are around 0.5. These results are thus not a strong indication of long range dependence. It is also interesting to note that the Hurst parameters for the *Jurassic Park I* encodings do not give a strong indication of long range dependence properties. This corroborates the observation that the GoP autocorrelation functions decay almost exponentially; thus indicating the memoryless property.

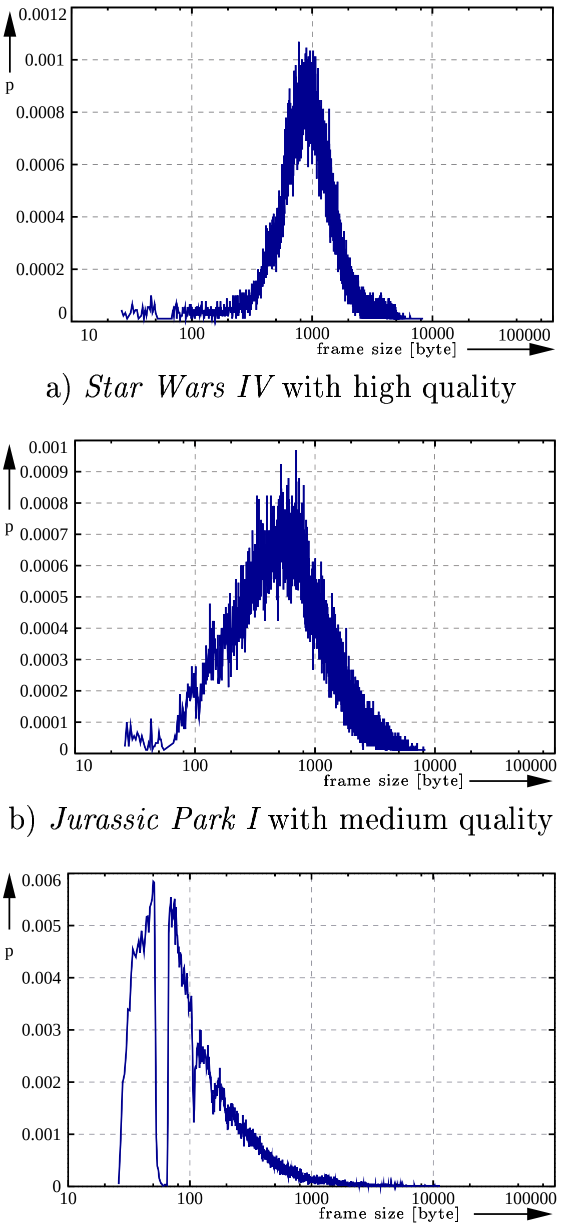


a) *Star Wars IV* with high quality

b) *Jurassic Park I* with medium quality

c) *Silence of the Lambs* with low quality

Figure 5.3: MPEG–4 bit rate versus time.

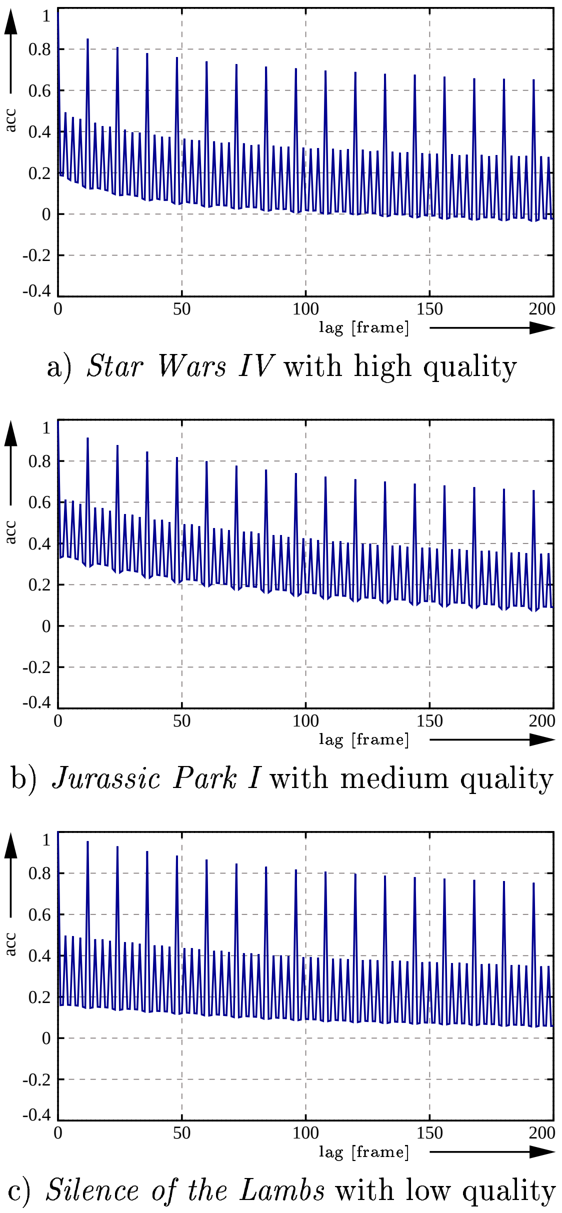


a) *Star Wars IV* with high quality

b) *Jurassic Park I* with medium quality

c) *Silence of the Lambs* with low quality

Figure 5.4: MPEG–4 frame size histograms.

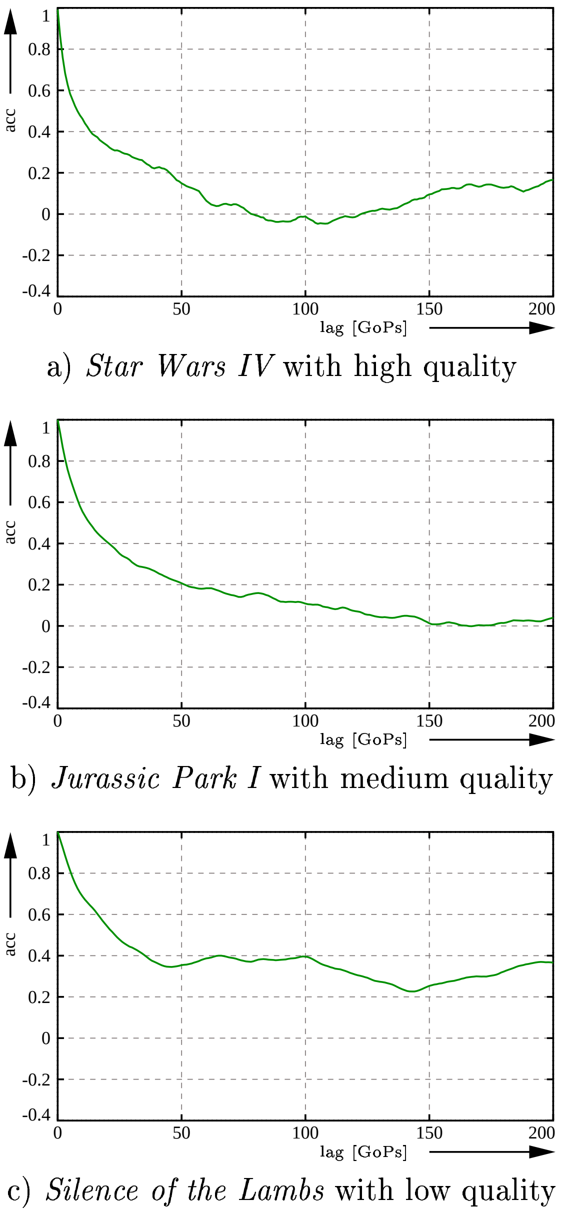


a) *Star Wars IV* with high quality

b) *Jurassic Park I* with medium quality

c) *Silence of the Lambs* with low quality

Figure 5.5: Autocorrelation of MPEG–4 frame size traces.

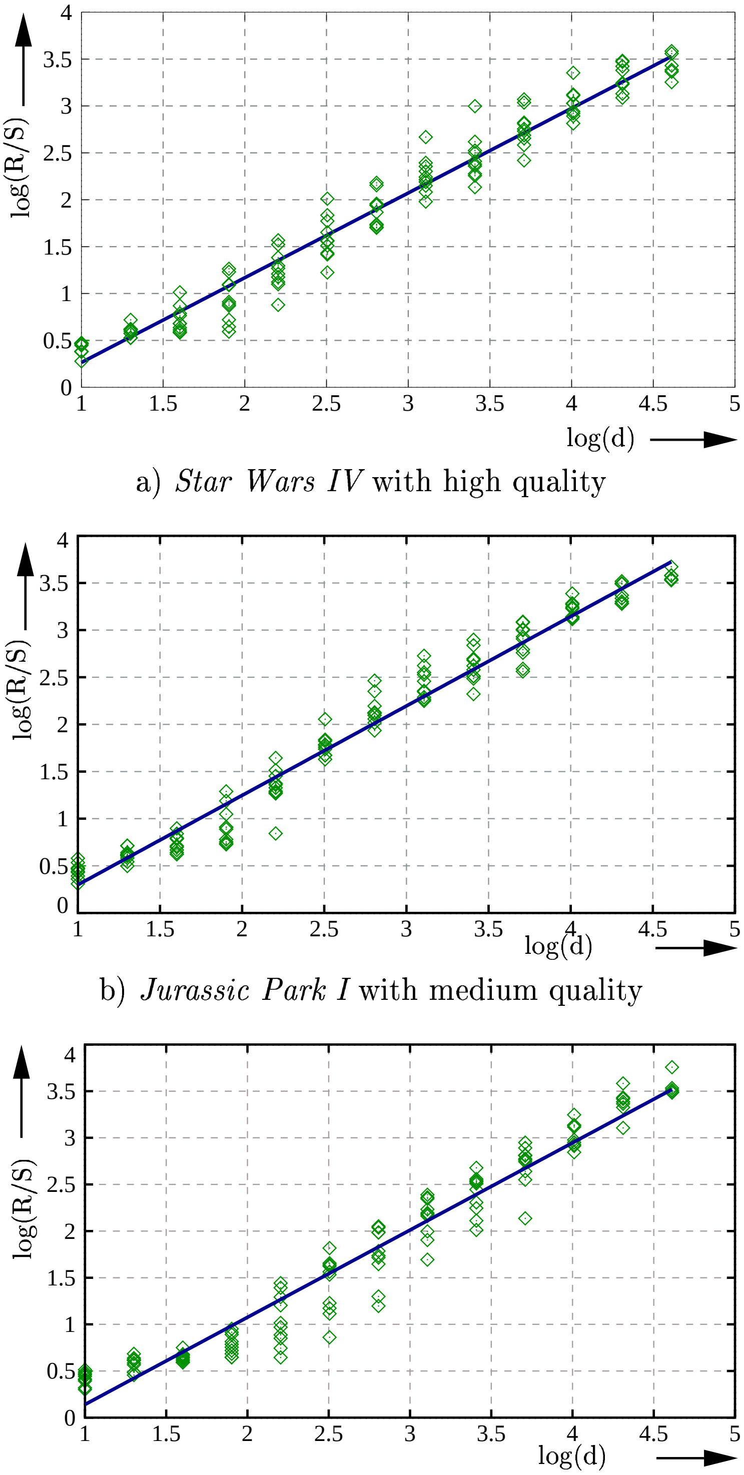


a) *Star Wars IV* with high quality

b) *Jurassic Park I* with medium quality

c) *Silence of the Lambs* with low quality

Figure 5.6: Autocorrelation of MPEG–4 GoP size traces.



a) *Star Wars IV* with high quality

b) *Jurassic Park I* with medium quality

c) *Silence of the Lambs* with low quality

Figure 5.7: Pox plots of R/S for MPEG–4 traces with aggregation level $a = 1$.

5.3.2 Statistical Analysis of H.263 Traces

In this section we conduct a thorough statistical analysis of the generated H.263 frame size traces. Let N denote the number of frames in a given video trace. Let $X_n,\ n = 1, \ldots, N$, denote the number of bits in frame n, i.e., the frame size of frame n. Let $t_n,\ n = 1, \ldots, N$, denote the frame period (display time) of frame n in msec. Let $T_n,\ n = 1, \ldots, N$, denote the cumulative display time up to (and including) frame n, i.e., $T_n = \sum_{k=1}^{n} t_k$ (define $T_0 = 0$).

As illustrated by the trace file, the T_n's are integer multiples of the basic (reference) frame period $\Delta = 40$ msec of the H.263 encoder. Notice, however, that some frames are skipped by the encoder striving to meet the specified target bit rate [136, p. 67]. This results in variable frame periods. Figure 5.8 gives the probability mass functions $P(t_n = l \cdot \Delta),\ l = 1, 2, \ldots$, of the frame periods of three generated H.263 traces. We observe from the plots the general tendency that smaller target bit rates result in larger frame periods, i.e., more frames are skipped. On the other hand, for VBR encodings, i.e., without specified target bit rate, the H.263 encoder typically does not skip any frames. Nevertheless, as we observe from Figure 5.8 a), most encoded frames have a frame period of 2Δ. This is because the encoder produces mostly PB frames, i.e., two consecutive frames are encoded as one entity. If no frame is skipped, the PB frame has a frame period of 2Δ when emitted by the encoder; at the decoder, however, the B frame is displayed first for a period of Δ and then the P frame for a period of Δ.

Table 5.8 gives an overview of the statistics of the frame sizes X_n. (To conserve space we give the statistics for all four target bit rate settings only for the *Jurassic Park I*, *Silence of the Lambs*, and *Star Wars IV* videos. We refer the interested reader to the technical report [46] and for the omitted results.)

First, we note that the H.263 encoder meets a given target for the average bit rate of the encoded video stream. To see this recall that the uncompressed YUV video stream has a bit rate of 38,016 byte/frame · 25 frames/sec = 950.4 kbit/sec. Also, recall that the compression ratio is defined as the ratio of the sum of the sizes of all unencoded YUV frames of the video to the sum of the sizes of all encoded frames emitted by the encoder. (Keep in mind that the H.263 encoder may (*i*) skip frames and (*ii*) encode two frames into one PB frame; therefore the number of encoded frames N may be smaller than the number of unencoded YUV frames.) To achieve a given target for the average bit rate of the encoded video stream the encoder enforces the same compression ratio for all videos. Even though, for a given target rate all encoded videos have the same average bit rate, their average frame sizes are different. For the 16 kbps target rate, for instance, the *Soccer* encoding has an average size of 655 bytes, while the *Silence of the Lambs* encoding has an average frame size of 370 bytes. Nevertheless, the encoder meets the target bit rate by skipping more frames of the *Soccer* video; i.e., the average frame period of the *Soccer* encoding is larger.

Comparing the 256 kbps target rate encodings with the VBR encodings we observe that some VBR encodings have higher compression ratios than the corresponding 256 kbps target rate encodings. The VBR encoding of *Star Wars IV*, for instance, has a compression ratio of 65, while the 256 kbps encoding has a compression ratio of 29.7. The more efficient VBR encoding, however, has a larger variability of the frame sizes. It is important to note that for variable frame period H.263 encoded video, the frame sizes are only one component of the video stream statistics. For the complete picture we need to consider the frame sizes in conjunction with their associated frame periods. Clearly, if the larger frame sizes of the VBR H.263 encodings were associated with larger frame periods, and vice versa, then the larger frame periods could be used to smooth out the larger frames. We shall see shortly that this is to a limited extend possible.

Figure 5.9 gives the histograms of the frame size X_n for three traces. We observe that the 256 kbps target rate encoding of *Jurassic Park I* has a pronounced bi–modal distribution of the frame sizes. This is because the encoder typically produces (*i*) P frames with an average size of roughly 3 kbytes, and (*ii*) PB frames with an average size of 6 kbytes. Similar observations hold for the depicted frame size histogram of the 16 kbps target rate encoding of *Silence of the Lambs*, as well as the other encodings.

To get the complete picture of the H.263 video stream statistics we define for a given H.263 frame size trace, two different traces that associate the frame sizes X_n with the frame periods t_n. (This will also facilitate the analysis of the H.263 video correlations and long range dependence characteristics.) First, we consider a "stuffed" frame size trace $F_m,\ m = 1, \ldots, T_N/\Delta$, obtained by "stuffing" zeros for the skipped frames into the generated frame size trace $X_n,\ n = 1, \ldots, N$. Formally,

$$F_m = \begin{cases} X_n & \text{for } m = \frac{T_n}{\Delta},\quad n = 1, \ldots, N \\ 0 & \text{for } m \notin \{\frac{T_1}{\Delta}, \ldots, \frac{T_N}{\Delta}\}\ . \end{cases}$$

The stuffed frame size trace reflects the traffic characteristics at the encoder output, where the frames of sizes X_n are emitted at discrete instants T_n.

Secondly, we introduce the rate trace $r(t),\ 0 \leq t \leq T_N$. We convert the discrete frame size trace $X_n,\ n = 1, \ldots, N$, to a fluid flow by transmitting the frame of size X_n at the constant rate X_n/t_n over its frame period, i.e.,

$$r(t) = \frac{X_n}{t_n} \quad \text{for } T_{n-1} < t \leq T_n,\ n = 1, \ldots, N.$$

The fluid flow characterization is an approximation of a system that transmits the frame of size X_n in many small packets that are equally spaced over the frame period of length t_n. For infinitesimally small packets this approximation gives a fluid flow of rate X_n/t_n over the frame period of frame n. This fluid flow approximation is popular in teletraffic studies since it significantly simplifies mathematical analysis [157]. Note that $r(t)$ changes its value only at integer multiples of the reference frame period Δ. A more convenient representation of $r(t)$ is thus obtained by "sampling" at Δ–spaced intervals. We define the sampled rate trace as

$$R_m = r(m \cdot \Delta),\ m = 1, \ldots, T_N/\Delta.$$

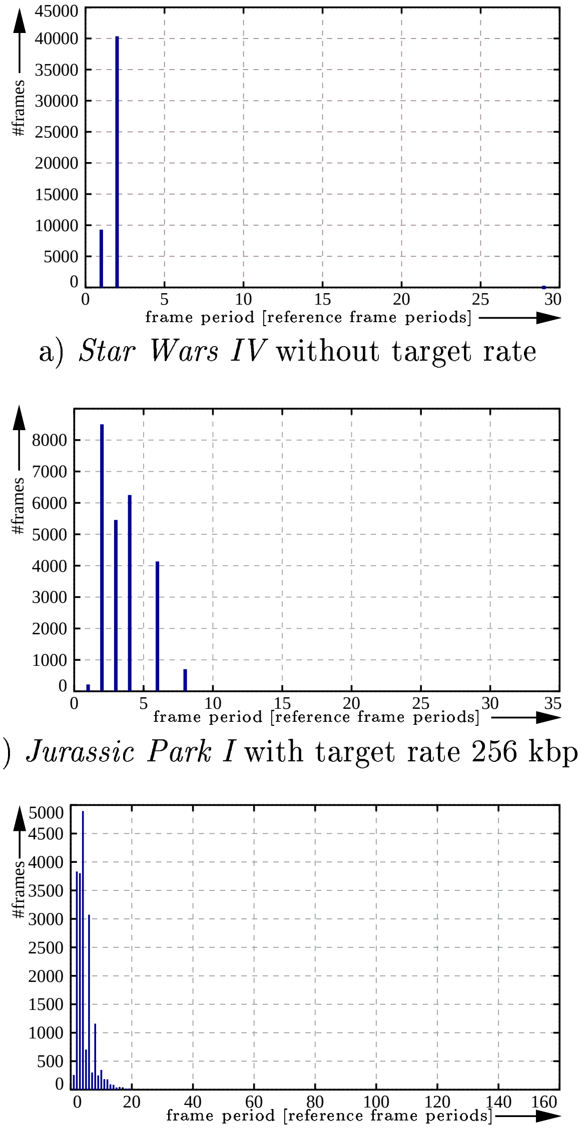
In the following we study the statistical properties of the "stuffed" frame size traces $F_m,\ m = 1, \ldots, T_N/\Delta$, and the sampled rate traces $V_m,\ m = 1, \ldots, T_N/\Delta$, obtained from the generated H.263 frame size traces. Table 5.9 gives an overview of the statistics of the "stuffed" frame size traces and the sampled rate traces. First, we compare the frame size statistics from Table 5.8 with the sampled rate trace statistics in Table 5.9. We observe that for the target rate encodings transmitting each encoded frame at a constant rate (fluid rate) over its frame period significantly reduces the variability of the encoder output. This is because some extremely large frames are associated with large frame periods. Nevertheless, the peak–to–mean ratios of the rate traces with fixed target rates are typically five and larger. On the other hand, for VBR encodings the rate traces have larger variability than the frame size traces (see Table 5.8). We also observe from the statistics of the "stuffed" frame size traces that transmitting each frame at a constant rate over one reference period of length Δ gives extremely variable encoder output for small target rates.

As alluded to above, some VBR encodings have significantly smaller average bit rates than the 256 kbps target rate; see, for instance, the *Star Wars IV* encoding. The more efficient VBR encoding, however, entails more variability in the encoded video stream. Loosely speaking, with VBR encoding the encoder produces high output rates when they are needed to encode complex scenes without reducing the video quality. We note, however, that a detailed study of the video stream statistics in conjunction with the perceived video quality is beyond the scope of this article.

Figure 5.10 gives the "stuffed" frame size trace F_m as a function of the index m (in reference frame periods of length Δ) for the generated H.263 traces. Figure 5.11 gives the autocorrelation coefficient of the "stuffed" frame size trace $\rho_F(k)$ and the autocorrelation coefficient of the sampled rate trace $\rho_R(k)$ as a function of the lag k (in reference frame periods of length Δ) for the generated H.263 traces over 14 reference frame periods of length Δ. We observe that the "stuffed" frame size traces have rather "jerky" autocorrelation functions. This is because the zeros in the "stuffed" traces give negative spikes. The sampled rate traces, on the other hand, have smooth autocorrelation functions. To get a better picture of the long term correlations we give in Figure 5.12 the autocorrelation functions over 500 reference frame periods. We observe that the autocorrelation function of the VBR encoding (i.e., without specified target rate) of *Star Wars IV* decays very slowly; for a lag of $d = 500\Delta$ the correlation coefficient of the sampled rate trace is roughly 0.18. (The "jerky" autocorrelation function of the "stuffed" frame size trace gives rise to the light gray shading in this plot.) The autocorrelations of the depicted target rate encodings, on the other hand, decay quickly to zero.

Table 5.10 gives the Hurst parameters of the sampled rate traces as a function of the aggregation level a. Figure 5.13 gives pox plots of R/S for an aggregation level of a = 1. We notice from the pox plots given here and on our web sites that two problems arise when applying the R/S statistic to the H.263 traces. First, some pox plots for the aggregation level $a = 1$ have outliers for small lags d. One strategy could have been to remove those outliers; this would have given larger estimates for the Hurst parameter for the aggregation level $a = 1$. We chose not to do so in order to keep the least–squares fit estimation simple and automated. In interpreting the results in Table 5.10 we ignore the column $a = 1$ and focus on the larger aggregation levels instead. Secondly, the pox plots for aggregation levels of $a = 200$ and larger for encodings with a specified target bit rate, typically do not settle down around a straight "street". We suspect that this is due to the fact that the H.263 encoder typically skips many frames to meet a specified rate target. As a result these traces might not have a sufficiently large number of values to estimate the Hurst parameter for large aggregation levels.

Nevertheless, we observe that all VBR encodings have Hurst parameters above 0.7 for all aggregation levels of $a = 12$ and higher. This indicates a high degree of long range dependence in the VBR traces. We also observe from the table that the encodings with a specified target bit rate have Hurst parameter above 0.7 for the aggregation levels $a = 12$, $a = 50$, and $a = 100$. This gives some indication of long range dependence properties. However, more studies on the long range dependence properties of H.263 traces are needed.

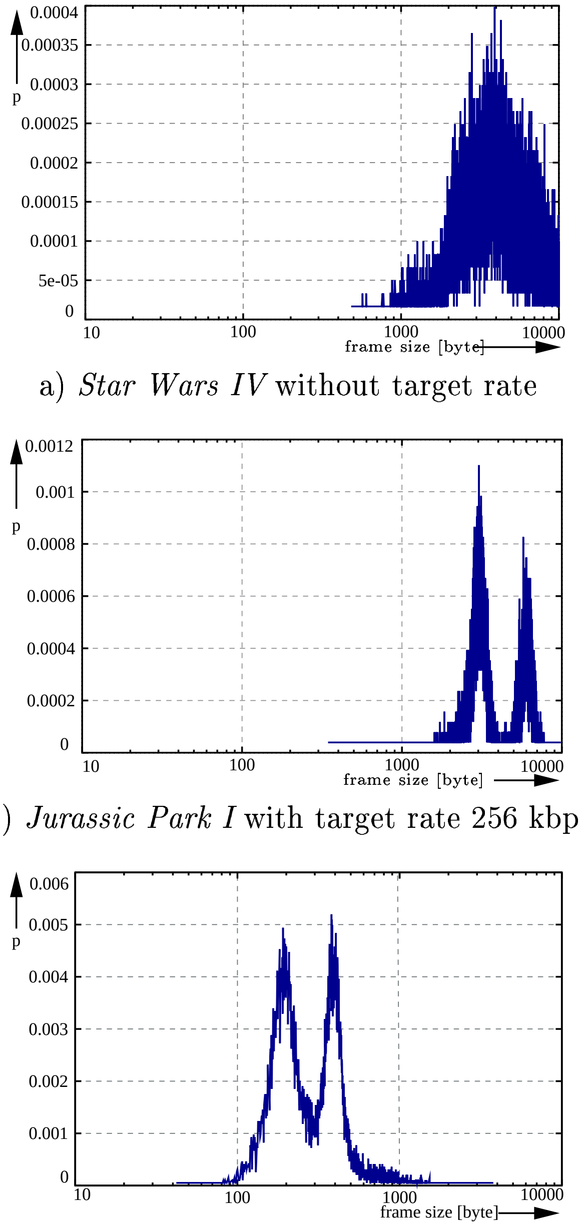


a) *Star Wars IV* without target rate

b) *Jurassic Park I* with target rate 256 kbps

c) *Silence of the Lambs* with target rate 16 kbps

Figure 5.8: Probability mass functions of frame periods of H.263 traces.



a) *Star Wars IV* without target rate

b) *Jurassic Park I* with target rate 256 kbps

c) *Silence of the Lambs* with target rate 16 kbps

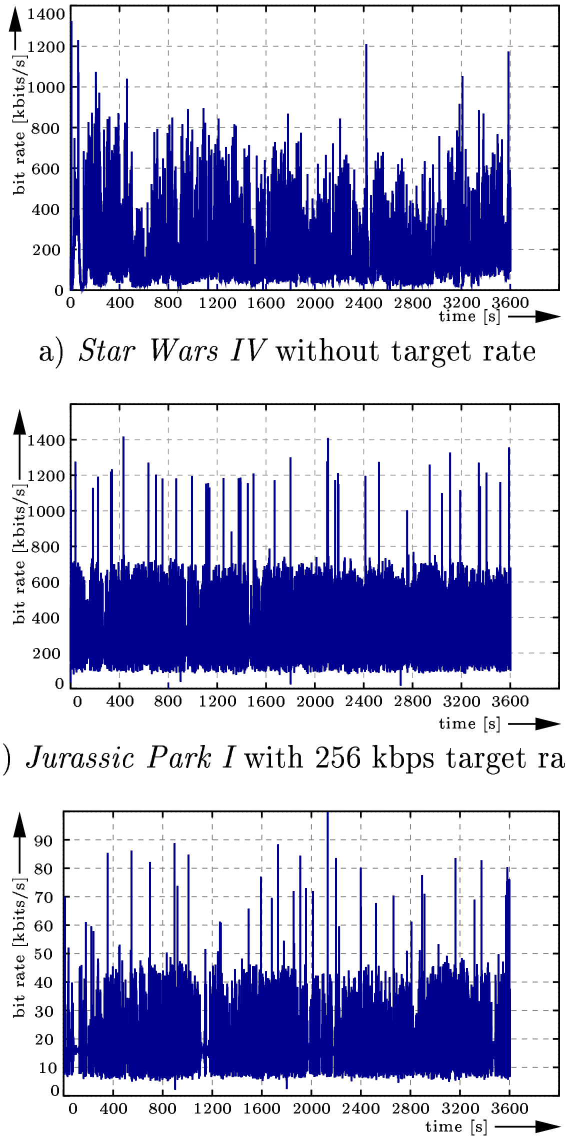
Figure 5.9: H.263 frame size histograms.

Table 5.8: Overview of frame size statistics of H.263 traces.

		Comp. ratio	Mean	CoV	Peak/Mean
Rate	Trace	YUV:H.263	$\bar{X}$ [byte]	$S_X/\bar{X}$	$X_{\max}/\bar{X}$
16 kbps	*Jurassic Park I*	476.36	476.36	0.67	20.83
	Silence of the Lambs	476.43	369.85	0.67	33.90
	Star Wars IV	476.43	326.02	0.61	11.32
64 kbps	*Jurassic Park I*	118.96	1132.02	0.36	7.89
	Silence of the Lambs	118.95	1129.94	0.41	11.10
	Star Wars IV	118.95	1153.32	0.43	7.11
	Mr. Bean	118.96	1115.56	0.40	3.90
	First Contact	118.95	1099.94	0.46	5.33
	From Dusk Till Dawn	118.95	1056.67	0.42	5.62
	The Firm	118.96	1186.73	0.35	3.53
	Formula 1	118.92	862.65	0.36	5.60
	Soccer	118.94	905.35	0.37	5.13
	ARD News	118.83	1236.72	0.38	5.74
	ARD Talk	118.94	1268.00	0.35	3.38
	N3 Talk	118.96	1145.44	0.39	4.24
	Office–Cam	118.94	1564.69	0.12	3.32
	The Firm	118.96	1184.41	0.36	3.82
	Starship Troopers	118.96	983.62	0.39	8.16
	Die Hard III	118.95	971.13	0.39	6.80
	Aladdin	118.96	931.06	0.39	4.78
	Robin Hood	118.95	969.43	0.38	6.52
	Susi und Strolch	118.95	991.21	0.41	4.11
	Simpsons	118.96	1090.63	0.41	6.24
	Futurerama	118.95	1171.28	0.40	6.22
	Southpark	118.93	1057.86	0.47	5.90
	Ski	118.96	869.19	0.37	6.30
	VIVA	118.96	1001.75	0.41	7.08
	Boulevard Bio	118.95	1331.49	0.34	3.72
	Lecture Room–Cam	118.96	1714.80	0.19	1.91
	Parking–Cam	118.96	1586.42	0.08	4.73
256 kbps	*Jurassic Park I*	29.73	4533.67	0.35	2.61
	Silence of the Lambs	29.73	4453.81	0.39	5.00
	Star Wars IV	29.73	4563.53	0.33	3.58
VBR	*Jurassic Park I*	17.08	3993.31	0.64	4.55
	Silence of the Lambs	25.19	2703.46	0.99	10.27
	Star Wars IV	65.79	1048.21	0.66	8.58

Table 5.9: Overview of statistics of H.263 ”stuffed” frame size traces and sampled rate traces.

		”Stuffed” Frame Size Trace			Sampled Rate Trace		
		Mean	CoV	Peak/Mean	Mean	CoV	Peak/Mean
Rate	Trace	$\bar{F}$ [byte]	$S_F/\bar{F}$	$F_{\max}/\bar{F}$	$\bar{R}$ [kbit/s]	$S_R/\bar{R}$	$R_{\max}/\bar{R}$
16 kbps	*Jurassic Park I*	79.81	2.48	111.96	16	0.35	5.8
	Silence of the Lambs	79.80	2.38	157.14	16	0.37	6.3
	Star Wars IV	79.81	2.15	46.24	16	0.42	6.1
64 kbps	*Jurassic Park I*	319.59	1.73	27.96	64	0.42	5.7
	Silence of the Lambs	319.60	1.77	39.23	64	0.42	5.7
	Star Wars IV	319.62	1.81	25.66	64	0.45	5.2
	Mr. Bean	319.58	1.74	13.61	64	0.4	8.1
	First Contact	319.62	1.78	18.35	64	0.43	5.8
	From Dusk Till Dawn	319.62	1.70	18.59	64	0.39	6.3
	The Firm	319.59	1.78	13.10	64	0.45	5.4
	Formula 1	319.69	1.43	15.11	64	0.32	5.7
	Soccer	319.63	1.49	14.54	64	0.33	6.0
	ARD News	319.91	1.85	22.21	64	0.42	5.3
	ARD Talk	319.65	1.86	13.41	64	0.43	5.7
	N3 Talk	319.58	1.77	15.20	64	0.41	5.4
	Office–Cam	319.63	1.99	16.24	64	0.44	4.9
	The Firm	319.59	1.78	14.17	64	0.45	5.3
	Starship Troopers	319.58	1.60	25.10	64	0.36	5.7
	Die Hard III	319.61	1.58	20.66	64	0.32	5.6
	Aladdin	319.59	1.54	13.92	64	0.35	6.5
	Robin Hood	319.61	1.57	19.78	64	0.38	5.9
	Susi und Strolch	319.60	1.62	12.73	64	0.39	5.8
	Simpsons	319.57	1.73	21.30	64	0.45	5.8
	Futurerama	319.61	1.80	22.81	64	0.45	5.5
	Southpark	319.65	1.75	19.51	64	0.46	6.2
	Ski	319.58	1.45	17.13	64	0.30	5.7
	VIVA	319.59	1.63	22.20	64	0.36	5.5
	Boulevard Bio	319.60	1.91	15.49	64	0.43	5.8
	Lecture Room–Cam	319.59	2.13	10.24	64	0.48	6.2
	Parking–Cam	319.58	2.00	23.47	64	0.49	5.1
256 kbps	*Jurassic Park I*	1278.72	1.72	9.24	256	0.42	5.5
	Silence of the Lambs	1278.61	1.73	17.4	256	0.42	5.9
	Star Wars IV	1278.80	1.72	12.76	256	0.47	5.3
VBR	*Jurassic Park I*	2225.26	1.23	8.16	450	0.69	7.7
	Silence of the Lambs	1509.14	1.60	18.41	300	1.10	17.0
	Star Wars IV	589.63	1.24	15.25	120	0.76	11.0

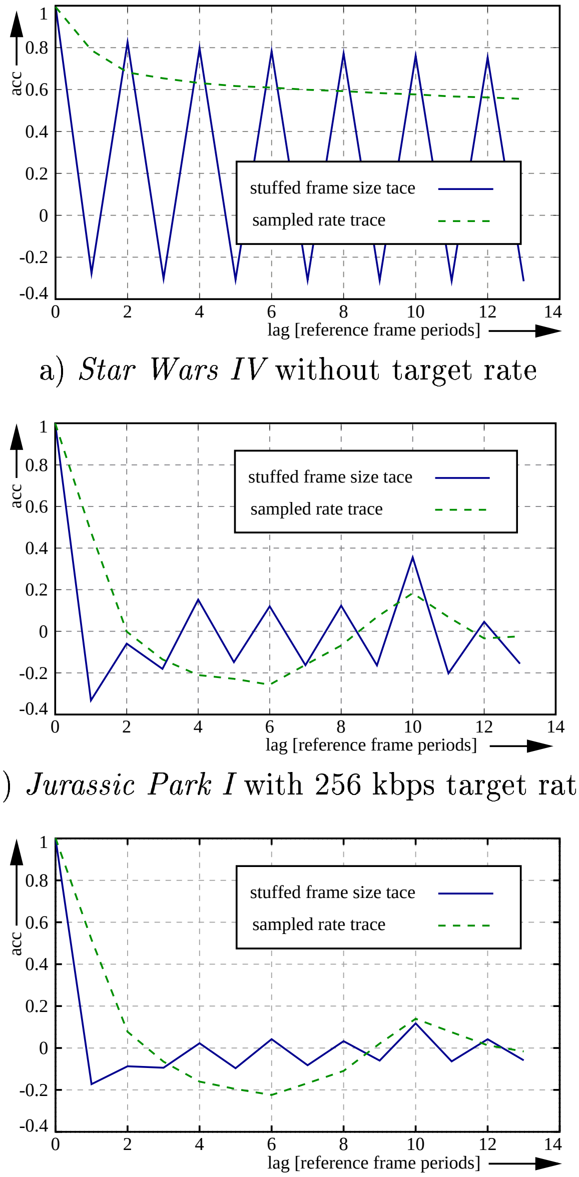


a) *Star Wars IV* without target rate

b) *Jurassic Park I* with 256 kbps target rate

c) *Silence of the Lambs* with 16 kbps target rate

Figure 5.10: H.263 bit rate rate versus time plot.

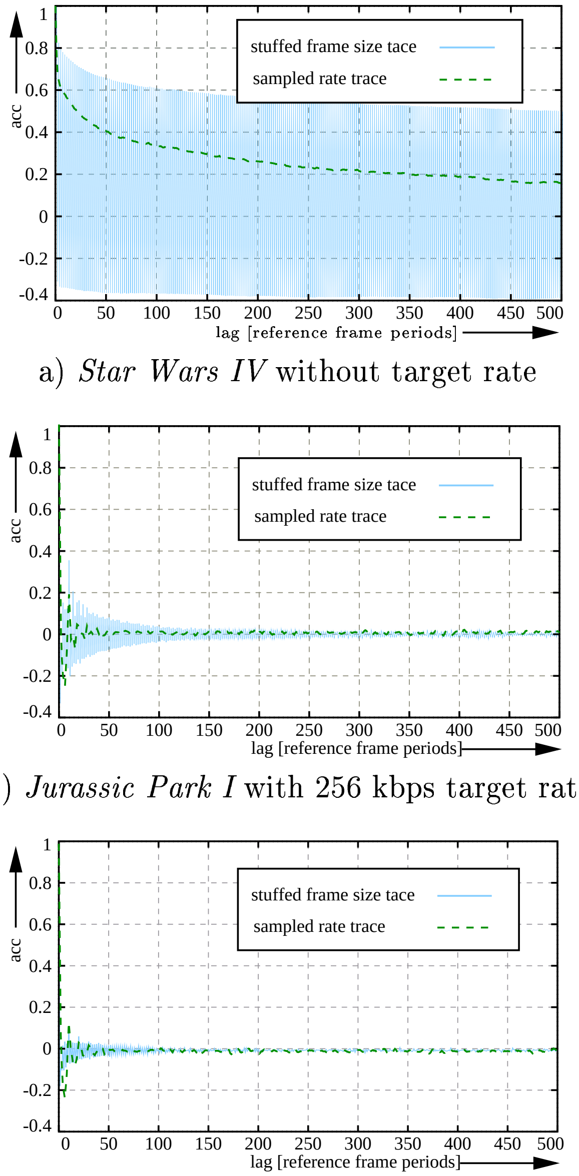


a) *Star Wars IV* without target rate

b) *Jurassic Park I* with 256 kbps target rate

c) *Silence of the Lambs* with 16 kbps target rate

Figure 5.11: Autocorrelation of H.263 ”stuffed” frame size traces and sampled rate traces over 14 reference frame periods.

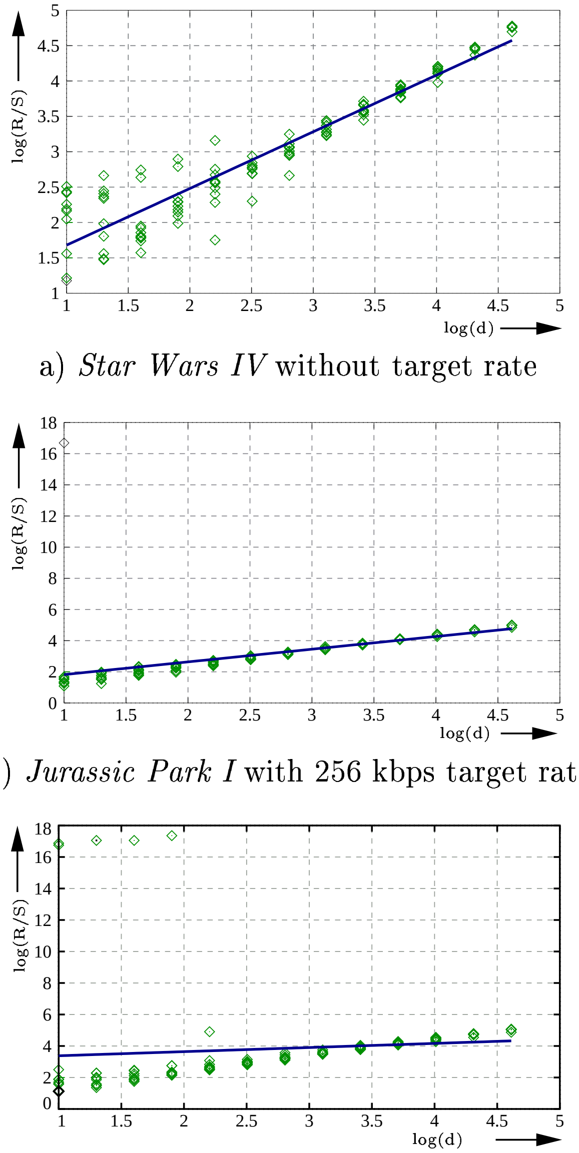


a) *Star Wars IV* without target rate

b) *Jurassic Park I* with 256 kbps target rate

c) *Silence of the Lambs* with 16 kbps target rate

Figure 5.12: Autocorrelation of H.263 "stuffed" frame size traces and sampled rate traces over 500 reference frame periods.



a) *Star Wars IV* without target rate

b) *Jurassic Park I* with 256 kbps target rate

c) *Silence of the Lambs* with 16 kbps target rate

Figure 5.13: Pox plots of R/S for H.263 "stuffed" frame size traces and sampled rate traces with aggregation level $a = 1$.

Table 5.10: Hurst parameters of H.263 sampled rate traces estimated from pox diagram of R/S as a function of the aggregation level a.

		Aggregation level a [reference frame periods Δ]								
Rate	Trace	1	12	50	100	200	400	600	700	800
16 kbps	*Jurassic Park I*	0.945	0.930	0.843	0.770	0.630	0.581	0.083	0.352	0.503
	Silence o/t Lambs	0.262	0.657	0.835	0.721	0.594	0.488	0.091	0.215	0.361
	Star Wars IV	0.947	0.944	0.872	0.762	0.621	0.534	0.157	0.384	0.511
64 kbps	*Jurassic Park I*	1.005	0.963	0.832	0.748	0.608	0.503	0.209	0.323	0.527
	Silence o/t Lambs	0.459	0.985	0.886	0.790	0.608	0.465	0.101	0.272	0.467
	Star Wars IV	0.963	0.952	0.903	0.802	0.620	0.509	0.150	0.388	0.465
	Mr. Bean	0.960	0.926	0.812	0.728	0.589	0.304	0.411	0.275	0.436
	First Contact	0.646	0.979	0.872	0.730	0.603	0.311	0.480	0.295	0.513
	Dusk Till Dawn	0.805	0.904	0.831	0.682	0.540	0.364	0.409	0.261	0.572
	The Firm	0.933	0.925	0.848	0.758	0.642	0.385	0.457	0.322	0.463
	Formula 1	0.772	0.910	0.772	0.616	0.403	0.134	0.0107	0.054	0.571
	Soccer	0.954	0.889	0.863	0.752	0.579	0.315	0.370	0.200	0.433
	ARD News	0.713	0.866	0.867	0.752	0.800	0.366	-0..157	0.544	-0.047
	ARD Talk	0.955	0.921	0.819	0.691	0.593	0.449	0.135	0.428	0.333
	N3 Talk	0.761	0.940	0.864	0.765	0.630	0.399	0.416	0.306	0.478
	Office–Cam	0.786	0.985	0.743	0.563	0.396	0.224	-0.186	-0.002	0.143
	The Firm	0.930	0.923	0.807	0.694	0.512	0.216	0.046	-.225	0.152
	Starship Troopers	0.917	0.890	0.777	0.656	0.394	0.090	00.034	-0.002	0.100
	Die Hard III	0.730	0.960	0.851	0.651	0.441	0.045	0.002	-.100	0.082
	Aladdin	0.974	0.895	0.794	0.633	0.492	0.134	00.079	-.181	-0.045
	Robin Hood	0.662	0.883	0.743	0.613	0.480	0.097	0.031	-.186	0.143
	Susi und Strolch	0.143	0.964	0.805	0.676	0.462	0.207	00.089	-.193	0.074
	Simpsons	0.605	0.964	0.644	0.405	-0.008	0.212	-0.117	0.035	-0.530
	Futurerama	0.900	0.913	0.669	0.456	0.231	0.163	-0.101	00.040	-0.593
	Southpark	0.967	0.902	0.720	0.472	0.131	0.264	-0.057	0.029	-0.650
	Ski	0.721	0.895	0.778	0.591	0.348	0.135	00.034	-.130	0.074
	VIVA	0.937	0.926	0.763	0.612	0.369	0.092	00.115	-.126	0.024
	Boulevard Bio	0.918	0.906	0.831	0.736	0.555	0.272	0.127	-.089	0.103
	Lecture Room–Cam	0.663	0.938	0.783	0.603	0.369	0.103	00.015	-.101	0.086
	Parking–Cam	0.409	0.975	0.757	0.532	0.303	-0.054	-0.178	-0.050	-0.139
256 kbps	*Jurassic Park I*	0.815	0.960	0.883	0.770	0.575	0.449	0.129	0.318	0.497
	Silence o/t Lambs	0.961	0.950	0.867	0.757	0.598	0.509	0.112	0.226	0.435
	Star Wars IV	0.777	0.960	0.872	0.743	0.585	0.495	0.111	0.328	0.497
VBR	*Jurassic Park I*	0.863	0.862	0.902	0.900	0.867	0.770	0.805	0.817	0.867
	Silence o/t Lambs	0.575	0.758	0.778	0.815	0.793	0.753	0.727	0.738	0.755
	Star Wars IV	0.633	0.893	0.882	0.882	0.882	0.859	0.856	0.876	0.865

5.4 Scalability of Video Streams

In this section an investigation of the scalability of video traces for simulation is given. Even if we have a larger set of video traces, anybody might be interested to have video traces with slightly different traffic parameters. E.g. many researchers use the MPEG–1 video traces presented by Rose [130] and scale the bit rate down or up to their specific requirements. We will show in the following that scaling can not be applied for all kinds of video traces.

In case the mean bit rate of a video trace file is *scaled* in that way that it fits the simulation requirements, each frame has to be changed in the trace file. To achieve a certain bit rate R_{target} all frames within the trace files are simply multiplied with a given factor α. If R_{trace} is the bit rate of the trace file, α is given by

$$\alpha = \frac{R_{\text{target}}}{R_{\text{trace}}}. \tag{5.1}$$

By following this procedure we will investigate the impact of scaling on rate adaptive as well as non rate adaptive video trace[2].

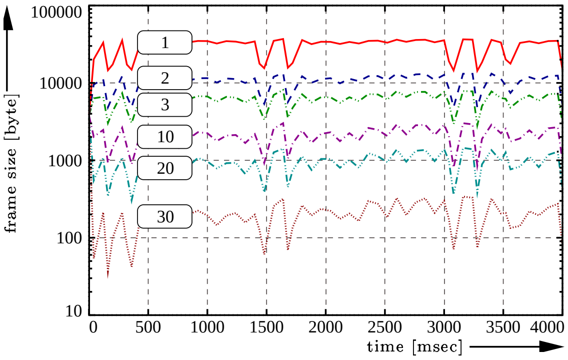


Figure 5.14: Non rate adapted video stream for different quality levels.

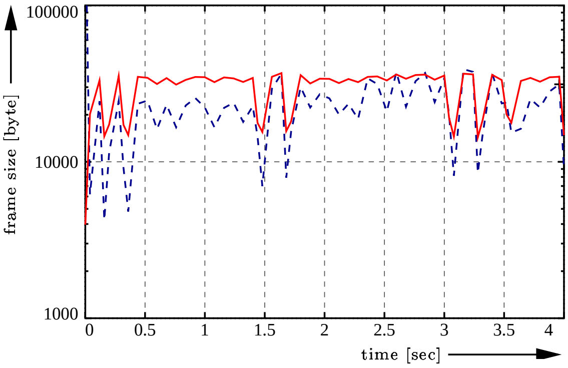


Figure 5.15: Effect of scaled video for non rate adaptive video streams.

In Figure 5.14 the frame size over time is given for non rate adapted video streams with different quality levels. The quality levels 1, 2, 3, 10, 20, 30 refer to the quantization level of the DCT matrix. Obviously a higher quantization results in a lower bit rate. But, even if the total bit rate decreases the characteristics of the video streams are nearly the same. Therefore, in Figure 5.15 we compare a video stream with quality level 1 with a scaled video stream of quality level 30. Both streams have the same mean bit rate ($\alpha_{q30 \to q1}$=114) and nearly the same traffic characteristic. The frame sizes of the *scaled* video stream seem to be smaller. The reason is that the first frame (I–frame) has the same size for all quality levels (see Figure 5.14). The I–frame of the scaled video stream become much larger by the scaling process, while the other frames becomes smaller. The effect is negligible for

[2] The presented results are based on the reference video stream *Coastguard.yuv* taken from the official MPEG–4 ftp site. We note that our results do not depend on this specific video sequence, but can also be achieved with *akiyo.yuv*, *Foreman.yuv*, etc.

larger observation times. Nevertheless, because of this example and further examples that are not shown here (can be seen on the web page [135]) scaling for non rate adaptive video streams can be applied.

On the other hand we investigate the scaling for rate adapted video stream. In this case the encoder produces video streams with a given target bit rate. In our example we encode the same video sequence with 16 kbps, 64 kbps, and 256 kbps. In Figure 5.16 the resulting frame sizes are given over the time. In Figure 5.17 the 16 kbps and 64 kbps encoded streams are scaled up to a 256 kbps stream with a factor of α=16 and α=4, respectively. This time the scaling process results in different traffic patterns for some time instance. Especially the low rate encode stream of 16 kbps differs dramatically if it is scaled up to a 256 kbps stream. Therefore we conclude that *scaling* of rate adaptive video streams can lead to significant differences in the traffic pattern.

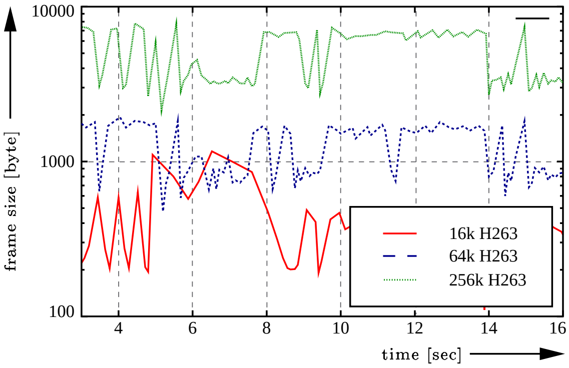


Figure 5.16: Rate adapted video stream for different rates.

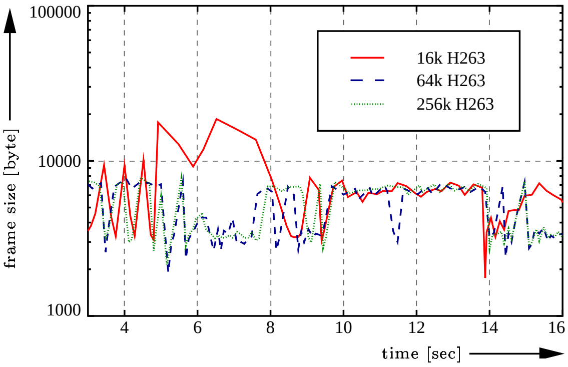


Figure 5.17: Effect of scaled video for rate adaptive video streams.

We conclude that scaling for non rate adapted video traces can be applied achieving small difference in the traffic pattern, while the scaling of rate adaptive video traces lead to total different traffic pattern. We have encoded video streams with different quality levels in our library available for the public of frame size traces [135]. Other trace files, like [130], have to be scaled with the resulting errors presented above.

5.5 Error–Prone Video Transmission

To make the effect of error–prone transmission over the wireless link visible, we combined the original video bit stream with the errors faced in the simulation. Afterwards we looked at the error–prone video sequence to have a first feeling of the impact of the erroneous video transmission. Therefore the following setup, given in Figure 5.18, was chosen. As explained earlier, the trace files are derived from the original video bit stream by a parser. The trace files are used within the simulations. Within the simulation it is possible to mark successfully transmitted frames. Non successfully transmitted frames are written into an `error-list`. By an extended version of the parser (GenErr.cc) it is possible to skip video frames from the original bit stream given by the `error-list`.

In Figure 5.19 and 5.20 a comparison of the video sequence of the *Aladdin* movie after

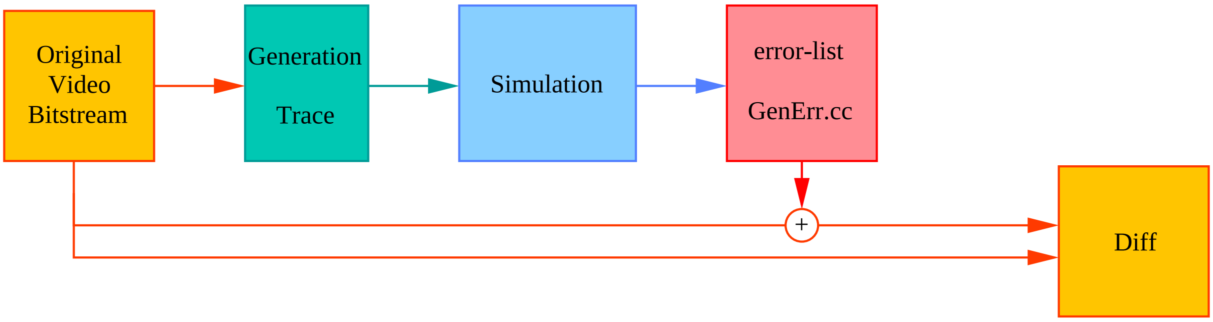


Figure 5.18: Comparison–Setup.

two different transmission schemes (sequential and **S**imultaneous **MAC** **P**acket **T**ransmission (*SMPT*) see chapter 4) is given for one specific time instant. The picture obtained with an error–free transmission is given for comparison.

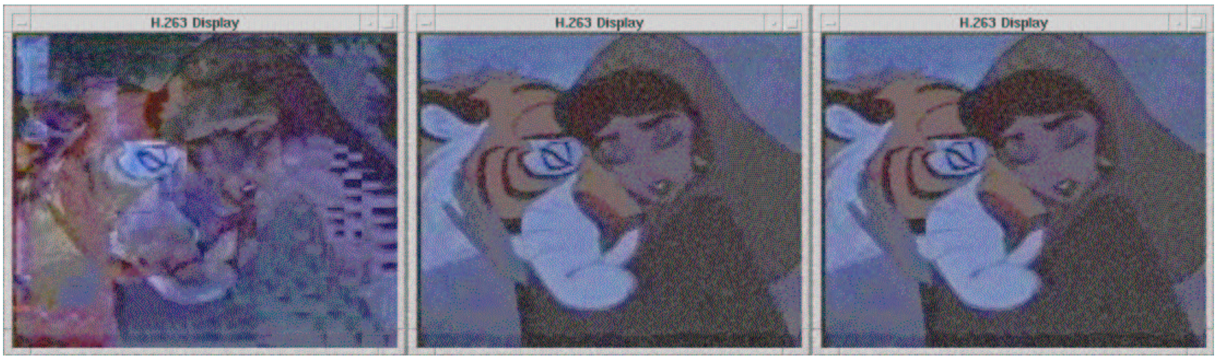


Figure 5.19: Comparison I of video sequence *Aladdin* after sequential (left), SMPT (middle), and the error–free (right) transmission.

Without going into the SMPT details we give a short explanation of the simulation setup. In the considered scenario there are nine ongoing video streams in one single cell, the link layer buffer is limited with $L_{Queue} = 40$ LPDUs, and the delay bound is $d_{delay} = 100$ msec. For this scenario we choose a 64 kbit/s video stream transmitted over the wireless link by each WT towards the base station. For this specific scenario we forced one WT to use an assigned video stream while all other WTs have chosen their trace files randomly. We observe that for both pictures the SMPT approach gives significantly better quality than the sequential transmission approach. In fact the SMPT picture is almost as good as the picture that would be obtained with an error–free transmission; it is only noticeable that in the SMPT picture the contours are somewhat *washed out* and not as sharp as in the error–free picture. The picture obtained with sequential transmission, on the other hand, is severely degraded in quality; it has several obvious artifacts (also, the left two–thirds of the picture in Figure 5.20 lag behind the error–free video picture). The degradations in the pictures are caused by video frames that missed their deadline and therefore could not be decoded. With each missed video frame the decoder misses

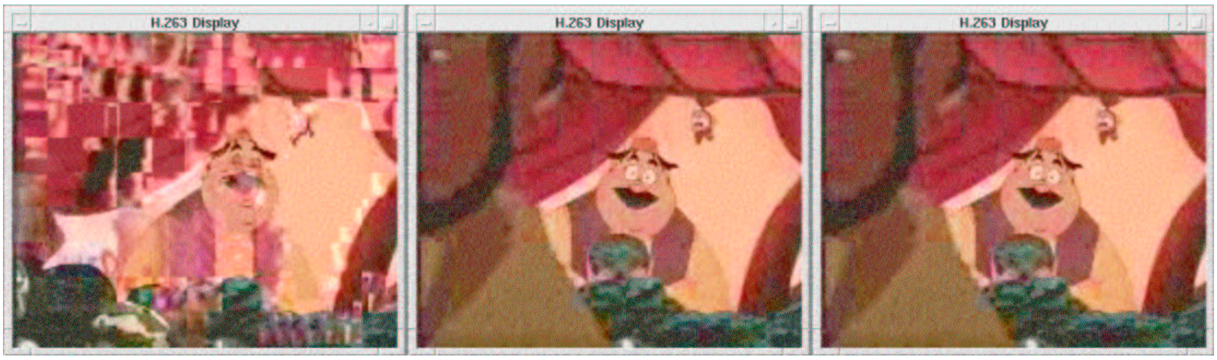


Figure 5.20: Comparison II of video sequence *Aladdin* after sequential (left), SMPT (middle), and the error–free (right) transmission.

the update of a number of macro–blocks which then results in artifacts and washed out contours in future frames that are predictive encoded with respect to the missing frame.

We note that these pictures are only indented to give a rough impression of how effective the simple SMPT technique is. We did not consider any refinements, such as error concealment techniques [158] which can decode partial video frames and thus improve the quality further at the expense of added complexity.

5.6 Conclusion

In this chapter we have presented and studied frame size traces of MPEG–4 and H.263 encoded videos. The parameters for the video encoding were chosen to allow video transmission over wireless bandwidth–limited channels. We have chosen the QCIF digital frame formats to achieve smaller bit rates. We have encoded over 25 videos of 15, 20, 30, or 60 minutes length. For each video we have generated MPEG–4 and H.263 encodings with several different quality levels. All in all, we have generated and studied over 150 hours of video traces. The results were presented in [46, 134, 135] and a publicly available library [135].

We have conducted a detailed statistical analysis of the generated traces. We have studied moments and autocorrelations as well as the long range dependence characteristics. We have estimated the Hurst parameter of the traces with the R/S statistic. For the analysis of the H.263 encodings, which have variable frame periods, we have introduced the notion of a *rate trace*. The rate trace facilitates the analysis of the H.263 frame sizes in conjunction with their associated frame periods. We have found that the traces are typically highly variable in their frame sizes and bit rates; especially the traces of low quality encodings. It is not possible to predict the traffic characteristic of a video because of its group (movie, cartoons, sport events, etc). In general sport events have always larger bit rates than the other groups, but no general classification can be presented. Therefore we will use a large set of video sequences of each group within the following simulation. Also, many of the traces show clear indications of long range dependence properties, which permit us to use them for our simulation environment of a wireless network. The presented traces can be easily used by other researchers for their development and evaluation of protocols and mechanisms for packet–switched networks, wireless networks, and optical networks. We conducted even a simulation interface for the PTOLEMY [159] and NS [160] simulation tool in [135].

Furthermore we have shown which kind of video sequences can be scaled. We explain the problems with the scaling process and gave some examples. We have shown that rate adapted video traces can not be scaled without loosing the traffic characteristic. We claim that non rate adapted video can be scaled maintaining the traffic characteristic. This statement is based on one example shown in this chapter and the comparison of several video traces presented in [135].

6 Model Definitions

This chapter provides a detailed description of the scenario, the source model, and the wireless link model of the analytical and simulation based performance evaluation. Furthermore we describe the tools, which were used for the simulation based performance evaluation.

6.1 General Scenario Description and Assumptions

We consider a wireless network where multiple **W**ireless **T**erminals (*WTs*) operate within one wireless cell as shown in Figure 6.1. All WTs communicate with one central **B**ase **S**tation (*BS*), whose coverage defines the cell boundaries. We assume a slotted MC–CDMA based wireless communication system with **T**ime **D**ivision **D**uplex (*TDD*) and a frame length of τ_{TDD}. The TDD structure provides alternating downlink (BS to WT) and uplink (WT to BS) transmission slots.

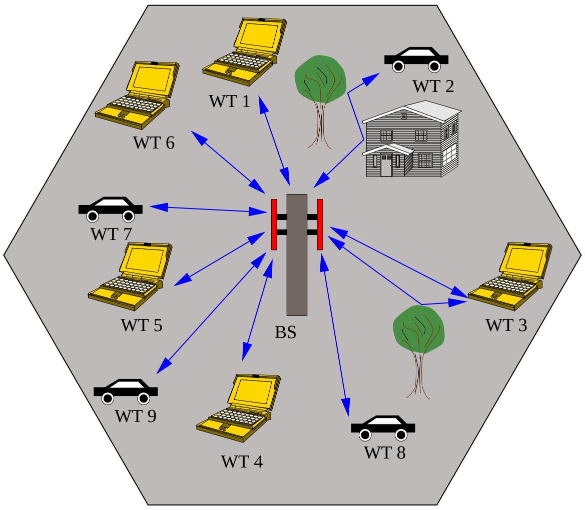


Figure 6.1: Wireless terminals communicating with one base station.

By the means of MC–CDMA WTs have the possibility to transmit and receive on R multiple channels. R denotes the maximum number of parallel channels per WT, where R is limited by the WT's hardware. All active channels of one WT are superimposed and modulated afterwards. For SMPT, we assume that **P**seudo–**N**oise (*PN*) code sequences using the *random sequence* approach, code sequences obtained by *random subspace* approach, or code sequences obtained by *controlled overlapping* approach are used within the uplink. The number of possible CDMA codes within one cell can be assumed to be much higher than the number of active used channels. After a WT is registered at the BS, the BS provides the WT with a unique set of PN–codes (see Section 4.5). This set of codes can be used by the WT at it's own discretion. Perfect power control within the WTs is assumed. In case a WT transmits packets on multiple channels at the receiver side either all or no packet is received correctly. This assumption is based on the consideration that all channels experience the same wireless path. Furthermore, in combination with the assumption of the **R**eed **S**olomon (*RS*) coding schemes, it would make sense to scramble the packets before sending and descramble them after receiving the packets.

As depicted in Figure 6.2 the application of a WT generates an information stream. This information stream is segmented into transport units (such as UDP or TCP segments and referred to as *segments* in the following) within the transport layer. The length of the segments is denoted by L_{S}. A segment is passed to the data link layer via the network layer, e.g. the **I**nternet **P**rotocol (*IP*). At the data link layer a segment is divided into N packets, $N \geq 1$. To each packet a header is added. The header is used to order data link packets, to assign them to the appropriate segments, and for error detection. One data link packet and a header are called a **L**ink–layer **P**acket **D**ata **U**nit (*LPDU*). The length of a LPDU is denoted by L_{LPDU} and fits into one transmission slot on the wireless link. All LPDUs are stored in a transmission queue with a fixed buffer capacity, L_{Q}, within the sender–side data link layer and are to be sent to the base station with different ARQ based transmission methods over the wireless link. We distinguish different sender–side strategies:

No knowledge at sender–side This means the sender will transmit all LPDUs stored in the queue indifferent how many retransmissions were already performed. We assume a limited sender–side queue. All segments that can not be stored in the queue are counted as loss. At the receiver–side a sub–set of successfully transmitted segments might not meet the delay constraints.

Knowledge at sender–side In this case the sender already knows the maximum **T**ransmission **W**indow (*TW*) , specified by the given jitter and delay constraints. In this case each received segment has to be counted as successfully transmitted. As given in Section 4.3.3 the abort criteria will help to maximize the capacity of the cell.

As mentioned already in Section 4.3.1 the sender–side transport entity passes one segment to the IP–based network layer at time T_0. If an error–free sequential transmission is assumed, the segment will arrive at the receiver–side network layer at time $T_1(L_{\text{S}})$, which depends on the segment length L_{S}. Thus, the minimal delay is given by $D_{\min}(L_{\text{S}})$. However, because of the error–prone wireless link, retransmissions of corrupted packets are required, which increase the segment's delay. Suppose that the receiver–side transport entity accepts only segments that arrive with a delay not higher than a maximum permissible delay D_{max}. If a segment arrives at the receiver–side at time T_2, such that $T_2 - T_0 \leq D_{max}$, the segment was transmitted successfully. For successfully transmitted segments the delay–jitter $J{=}T_2 - T_1(L_{\text{S}})$ is measured. Note, that the delay–jitter of

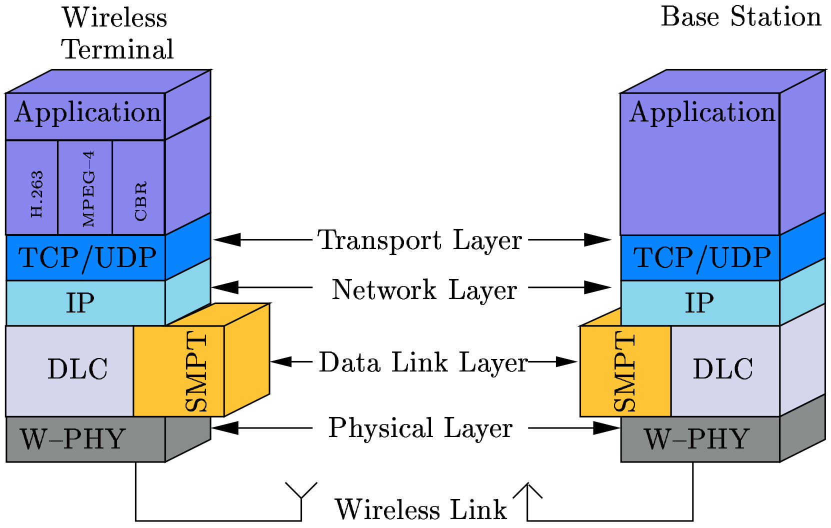


Figure 6.2: Protocol stack for the wireless terminal and the base station.

successful segments is never larger than τ_{jitter}. In addition to the delay–jitter, the delay and segment loss probability resulting in the goodput value, are measured. If all LPDUs of a segment are transmitted successfully, the receiving data link layer reassembles the LPDUs to the segment. The successfully received segment is passed immediately to the application via the network and transport layers.

6.2 Source Model

The traffic source component is an important entity in network modeling. The source behavior usually has a great impact on the overall performance evaluation of the model being investigated. The authors in [161] claim that different sources can easily lead to different performance measures. Therefore we have carefully chosen our traffic generators representing the requirements of todays wireless communication networks. On top of the transport protocol several applications can be used. We will focus on FTP and HTTP applications for the TCP layer and video application on top of UDP. We use the NS TCP model, which allows us to use traffic sources, which are comparable to results of other researchers. On the other hand we use our own MPEG–4, H.261, H.263+ and H.263 measurements [46, 134] to build our own traffic model.

6.2.1 Constant Bit Rate Traffic Generator

A very simple traffic model is the **C**onstant **B**it **R**ate (*CBR*) traffic generator. We distinguish from the following two cases: 1.) CBR traffic with constant packet length and 2.) CBR traffic with variable packet length. The former approach CBR_{fix} generates packets with constant length L_{packet} at equally spaced time–points. The time between two equally spaced time points is given by τ_{fix}. The transmission rate R_{fix} equals for this case

$$R_{fixPacket} = \frac{L_{packet}}{\tau_{fix}}. \tag{6.1}$$

Figure 6.3 depicts a generator for constant packet length. Once the generator was started by a Dirac impulse it produces the first packet and the next packets with time spacing of τ_{fix}. The second approach CBR_{var} generates packet i with variable length $L_{packet,i}$ at time–point t_i. For this case the transmission rate R_{var} is given by

$$R_{varPacket} = \frac{L_{packet,i}}{\tau_{var,i} - \tau_{var,i-1}}. \tag{6.2}$$

In Figure 6.4 a possible CBR generator is given for variable packet length. It has the same structure as in Figure 6.3, but the delay depends now on the randomly chosen packet length. Note, for both approaches the rates $R_{varPacket}$ and $R_{fixPacket}$ are the same.

6.2.2 Variable Bit Rate Traffic Generator

For a variable bit rate traffic generator we use the video traces introduced in Chapter 5. An example of such a trace file is given in Table 5.4. Each WT $j = 1 \ldots J$ has its own traffic generator. The generator selects randomly one of the H.263 or MPEG4 traces and generates random starting phases $\theta(j)$, into the selected traces. The $\theta(j)$ is independent and uniformly distributed over the lengths of the maximum starting phases value θ_{max}. Doing so it has to be assured that the WTs generate different traffic patterns. For the MPEG–4 generator, a frame with the length given in the trace file is generated every 40 ms. The H.263 generator has to consider in addition to the frame size also the time

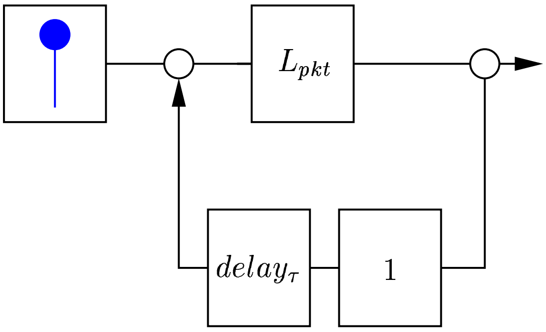


Figure 6.3: CBR traffic generator with constant packet length.

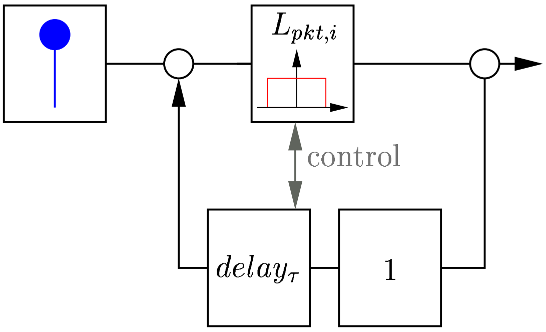


Figure 6.4: CBR traffic generator with variable packet length.

stamps in the trace files (multiple of 40 ms). After a randomly chosen time τ_{vsl} a new trace file is chosen. The duration of a video sequences τ_{vsl} is independent and uniformly distributed over the lengths of the maximum video sequence length[1].

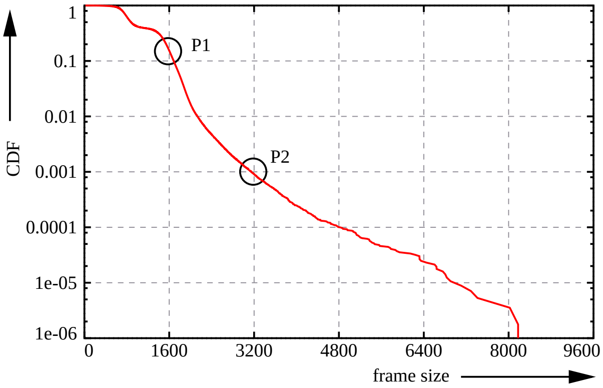


Figure 6.5: Complementary distribution function of the frame sizes for 25 different H.263 encoded films rate controlled with 64 kbit/s.

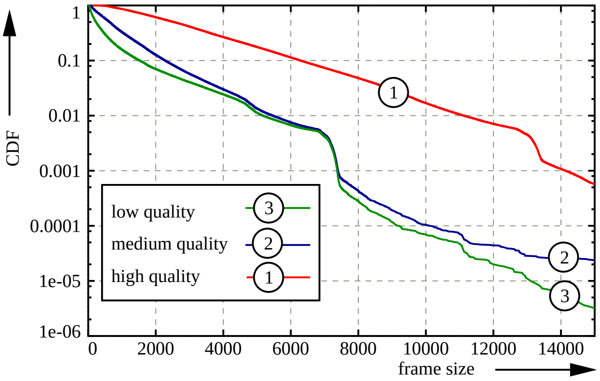


Figure 6.6: Complementary distribution function of the frame sizes for 25 different MPEG–4 encoded films with three quality levels (low/medium/high).

In Figure 6.5 and 6.6 the complementary distribution function (CDF) of the frame size traces of all 25 encoded films for H.263 and MPEG–4 is given, respectively. In Section 4.3.3 we have explained that the DLC sender will abort the transmission if a given maximum delay or jitter is exceeded. For a given DLC slot length and payload (e.g. 10 ms and 80 byte) these complementary distribution functions allow to calculate the minimum [2] loss rate. For the H.263 complementary distribution function we can see that a maximum delay of 200 ms leads to a minimum loss rate of 10% (P1 in Figure 6.5). This is because

[1] The video sequence length depends on the chosen video, which can be 15,20,30, or 60 minutes.

[2] Even for an error–free wireless link the loss rate can not be below than this.

of some frames that are too large. If the maximum delay is doubled up to 400 ms the minimum loss rate decreases to 0.1% (P2 in Figure 6.5).

6.3 Wireless Link Model

Wireless communication systems are limited in the performance by the characterization of the wireless radio channel. Modeling the radio channel is one of the most difficult parts of wireless radio system design. Depending on the cell environment (buildings, mountains, and moving obstacles), the subscriber's mobility, and the allocated spectrum the model will differ dramatically. The following section address the impacts on the radio channel. In Section 2.3.3 we had illustrated the relationship between packet error probability in CDMA systems and the number of active WTs. Equation 2.30 gives the packet error probability as a function of the number of active WTs, length of the packet, and the number of correctable bits. In the following we explain how we model the random nature of the wireless channel. We distinguish two different cases. The first case considers uncorrelated errors, while the second case reflects correlated errors.

6.3.1 Uncorrelated Packet Errors

For uncorrelated errors we simply derive the packet error probability p_{error} in dependency of the number of active WTs (see Equation 2.30) at the beginning of each uplink slot. In case a uniform random number $random_{uniform}$ is smaller or equal than the packet error probability p_{error} we consider this packet to be lost.

$$random_{uniform} \leq p_{error} \tag{6.3}$$

6.3.2 Correlated Packet Errors

Correlated packet errors capture the *bursty* nature of the wireless link. There are two important reasons for the wireless channel having bursty characterization: Rayleigh fading and shadowing. We show how to model these two effects and how to gather realistic parameters. Both approaches are based on a two state Markov chain. In [162, 28, 163, 164, 165] it was shown that the two state model is a useful and accurate wireless channel characterization for the design of higher layer protocols. It may be obtained from a more complex channel model, which may incorporate adaptive error control techniques, using weak lumpability or stochastic bounding techniques like those described in [166, 167].

Rayleigh Fading

In simulating the wireless link with Rayleigh fading we follow the well–known Gilbert–Elliot model. In [168] Gilbert used a two state Markov chain to capture the bursty nature of the wireless channel. Elliot generalized the Gilbert model slightly in [169]. In Figure 6.7 the Gilbert–Elliot model is depicted. Each wireless link is modeled as an independent discrete–time Markov chain with two states: *good* and *bad*. The transition probabilities of the Markov Chains are p_{GG}, p_{BB}, p_{BG}, and p_{GB}. The steady state probabilities of being in the *good* or *bad* phase are P_{Good} and P_{Bad} respectively. We assume that the value of the **B**it **E**rror **P**robability (*BEP*) in the *good* channel state, which has an average sojourn time of τ_{good}, depends only on the number of active CDMA channels. In the *bad* channel state, which has an average sojourn time of τ_{bad}, we assume that no communication is possible.

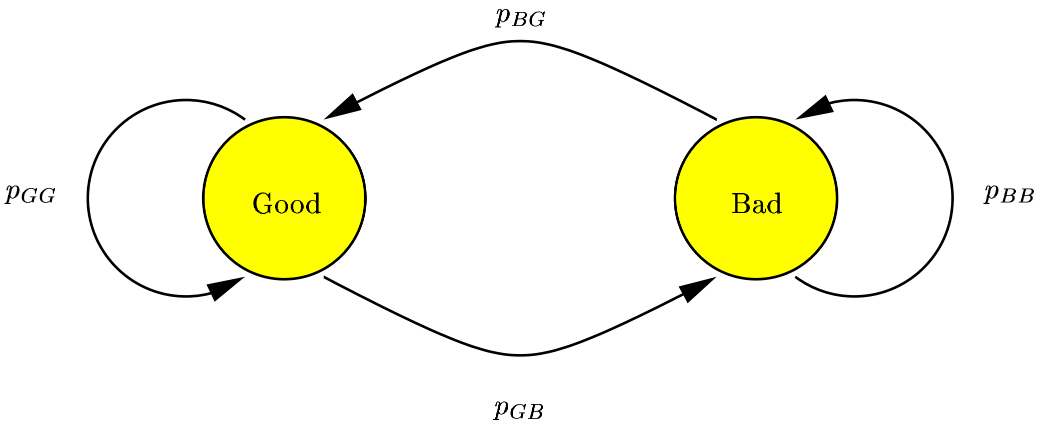


Figure 6.7: Discrete–time Gilbert–Elliot Model with *good* and *bad* channel state for flat fading.

To switch between the two states a random number is chosen at the beginning of each time slot. Depending on the transition probability the channel state will be changed or not. Thus the sojourn time staying in a specific channel state is geometrically distributed.

The following is a description of the transition probabilities and steady state probabilities for the Gilbert–Elliot model as given in [170] and [171]. Rayleigh fading causes long time periods where the wireless channel suffers from significant degradation of the received signal and as a result increases the **B**it **E**rror **P**robability (*BEP*) dramatically. The maximum Doppler frequency $f_{Doppler}$ is given by

$$f_{Doppler} = \frac{speed_{WT}}{\lambda_{carrier}} = \frac{speed_{WT}}{c} \cdot f_{carrier} \tag{6.4}$$

with $speed_{WT}$ the WT's speed, $\lambda_{carrier}$ the carrier wavelength, $f_{carrier}$ the carrier frequency, and c the speed of light $(3 \cdot 10^8 m/s)$. As we mentioned above we distinguish between two channel states; if the received SNIR is larger than a specific threshold Γ, the channel is considered as in the *good* channel state. In [170] we find the following definition $\rho = \Gamma/\overline{\Gamma}$ (where Γ is the Rayleigh fading envelope and $\overline{\Gamma}$ the local root mean square level). In [170] it is claimed that under conservative assumptions ρ can be set to -20 dB. The average sojourn time τ_{bad} is given by

$$\tau_{\text{bad}} = \frac{(e^{\rho^2} - 1)}{\sqrt{2\pi} \cdot f_{Doppler} \cdot \rho}. \tag{6.5}$$

The number of channel changes from *good* → *bad* per second is given by

$$N = \sqrt{2\pi} \cdot f_{Doppler} \cdot \rho \cdot e^{-\rho^2}. \tag{6.6}$$

Under the assumption of steady state conditions, the probabilities that a given wireless channel is either in the *good* or in the *bad* phase are given by

$$\vec{P} = \begin{pmatrix} P_{Good} \\ P_{Bad} \end{pmatrix} = \begin{pmatrix} e^{-\rho^2} \\ 1 - e^{-\rho^2} \end{pmatrix} = \begin{pmatrix} p_{GG} & p_{BG} \\ p_{GB} & p_{BB} \end{pmatrix} \cdot \vec{P}. \tag{6.7}$$

In [172] the transition probabilities are approximated by Equation 6.8 where R_{symbol} is the transmission rate in symbols per second.

$$\begin{aligned} p_{GB} &= \frac{N}{R_{symbol} \cdot P_{Good}} &= \frac{e^{\rho^2}-1}{R_{symbol} \cdot \tau_{\text{bad}}} \\ p_{GG} &= 1 - p_{GB} \\ p_{BG} &= \frac{N}{R_{symbol} \cdot P_{Bad}} &= \frac{1}{R_{symbol} \cdot \tau_{\text{bad}}} \\ p_{BB} &= 1 - p_{BG}. \end{aligned} \tag{6.8}$$

At this point one possible parameterization of the Gilbert–Elliot model is given. As an example we consider the UMTS system with $f_{carrier} = 1.885$ GHz, a pedestrian moving with $speed_{WT}$ of 0.2 m/s (nearly without mobility) and 1.39 m/s (fast moving person with 5 km/h), and $\rho = 10$ dB: For this specific scenario we achieve an average sojourn time τ_{bad} of 105 ms and 15 ms respectively. Table 6.1 gives the values for two different channel conditions of the Gilbert–Elliot Model. The values are based on the work of Chien and Letteri presented in [170]. The only values chosen by us were: the carrier frequency, the symbol rate (both UMTS like), and the speed of the WT.

Table 6.1: Different parameters for the wireless link for two states of 10 dB and 20 dB.

Ratio of the Rayleigh fading envelope to the local root mean square level	10dB		20dB	
speed of light c $\left[\frac{m}{s}\right]$	29792458			
carrier frequency $\lambda_{carrier}$ $[GHz]$	1.885			
symbol rate R_{symbol} $\left[\frac{1}{s}\right]$	100			
ρ	0.31622		0.1	
WT speed $speed_{WT}$ $\left[\frac{m}{s}\right]$	0.2	1.4	0.2	1.4
Doppler frequency $f_{Doppler}$ $\left[\frac{1}{s}\right]$	1.2573	8.8027	1.2573	8.8027
average bad channel sojourn time τ_{bad} $[ms]$	105.50	15.07	31.88	4.55
Probability of good channel state P_{Good}	90.48%		99%	
Probability of bad channel state P_{Bad}	9.52%		1%	
transmission probability good → bad	0.009968	0.069776	0.003152	0.022065
transmission probability good → good	0.990031	0.930223	0.996847	0.977934
transmission probability bad → good	0.094779	0.663457	0.313644	1.0 [†]
transmission probability bad → bad	0.905220	0.336542	0.686355	0.0 [†]

[†] The average bad channel sojourn time is smaller than symbol length and therefore the calculation will not work here.

The values for the bad channel duration time depend on the speed of the WT. Therefore, in Figure 6.8 the bad channel sojourn time versus the speed of WT $speed_{WT}$ for two states of 10 dB and 20 dB is given.

After having set the parameters for the Gilbert–Elliot model, we have to set the **P**acket **E**rror **P**robability (*PEP*) for two different states. We assume that no communication is possible within the bad state [173, 174]. Therefore the PEP is set to 0.5. For the good state the packet error probability depends heavily on the number of active channels

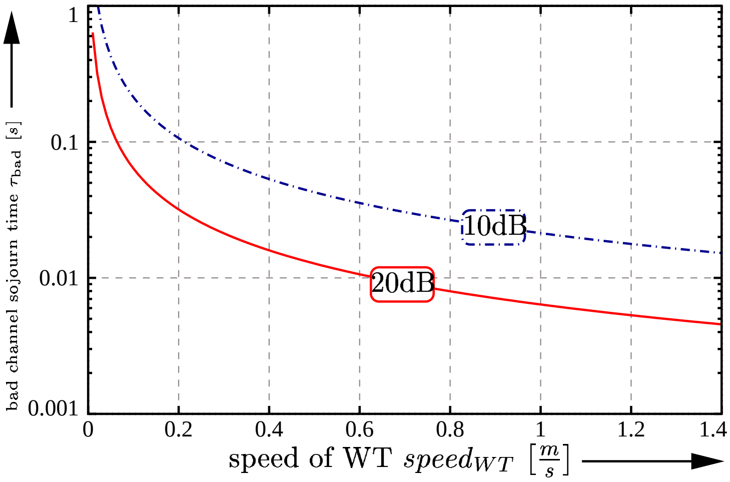


Figure 6.8: Bad channel sojourn time versus the speed of one wireless terminal $speed_{WT}$ for two states of 10dB and 20dB.

in the cell. As an example we assume the improved Gaussian approximation for **B**it **E**rror **P**robability (*BEP*) calculation (see Section 2.3.3.1) to calculate the packet error probability for BCH(1023,648,41) and BCH(1023,728,30) versus the number of active channels in the CDMA cell. The results are given in Figure 6.9 for a spreading gain of 16.

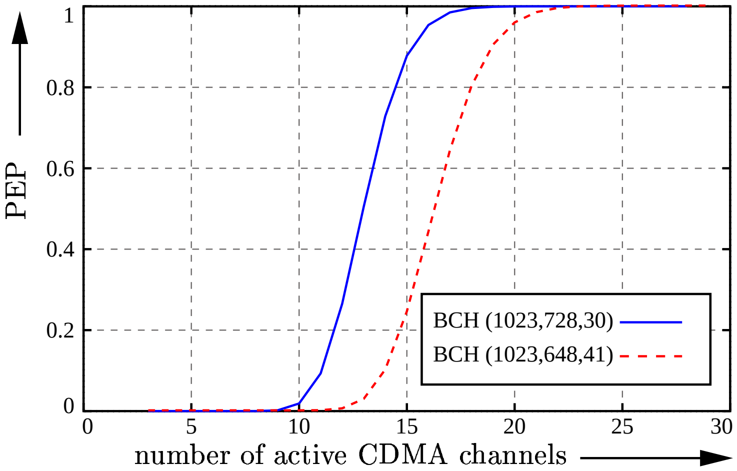


Figure 6.9: Packet error probability for BCH(1023,648,41) and BCH(1023,728,30) versus the number of active channels in the CDMA cell.

Once the parameters for the two state Markov chain are chosen we can concentrate on the modeling process. Each wireless link is modeled as an independent discrete–time Markov chain. Further, we assume that all parallel code channels of a wireless link are either in the good state or the bad state. As mentioned in Chapter 4.3 allocating channels will influence the effective bit rate (see Equation 4.1) by $\Delta_{Codes,k}$. To model the wireless channel we used a time discrete multi–layered two state Markov chain. As depicted in

Figure 6.10 we start the simulation at one selected operation point $P_{1,1}$ with a total number of b background channels. If the observing WT uses i additional channels the operation point will move to $P_{1,1+i}$ under the assumption that the channels states seen by each WT are independent of each other. The BEP of operation point P is used for the *good* channel state. This let increase the bit error probability on the wireless link modeling the impact of $\Delta_{Codes,k}$.

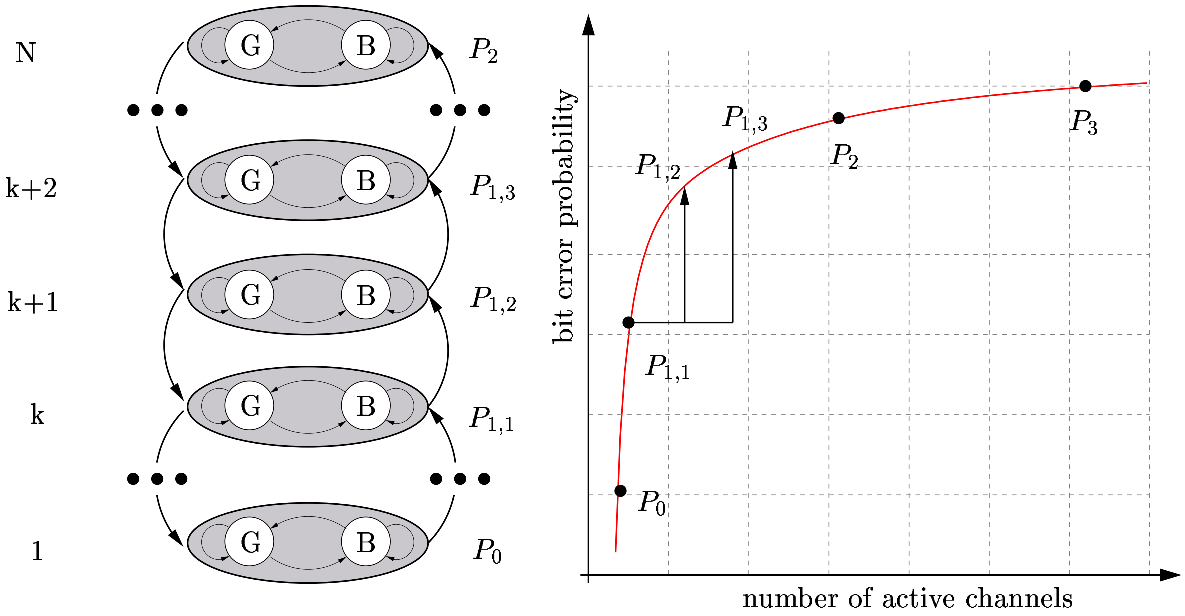


Figure 6.10: Multi–layered two state Markov chain for one selected operation point with N additional channels.

6.4 Simulation Tools

In the following we give a short description of the simulation tools we used to achieve the results presented in Section 8. The **N**etwork **S**imulator (*NS*) [160] is a **D**iscrete **E**vent (*DE*) simulator focusing at networking research. NS provides substantial support for simulation of TCP. Therefore the TCP model is generally accepted in the telecommunication community. We also used NS for all TCP simulations.

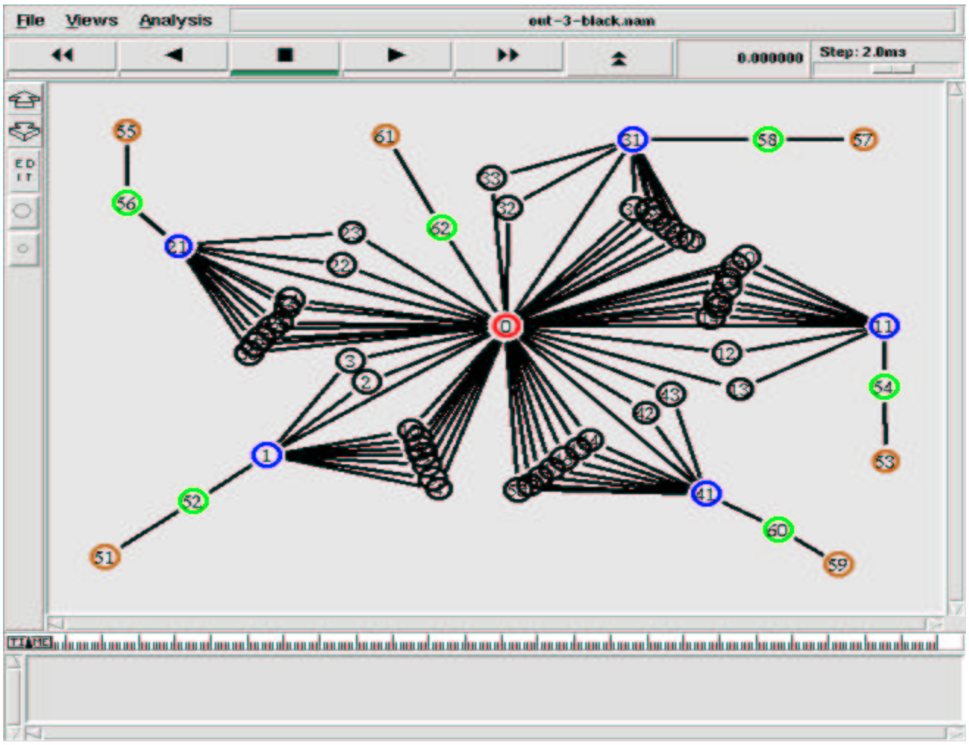


Figure 6.11: NS monitor screenshot of multi user scenario.

In Figure 6.11 it can be seen in the NS monitor tool how five WTs (e.g. node number $51, 53, 55, 57, 59$) are transmitting to one fixed terminal (node 61) via the base station over multiple channels using TCP. For our UDP simulations we used the DE domain in PTOLEMY [159]. The DE domain provides a general environment for time–oriented simulations. In contrast to NS no transport or higher protocol layers are implemented. PTOLEMY is more suitable to simulate the link level protocols for multi–user systems. PTOLEMY was used for UDP simulation. The stochastic nature of the processes occurring in telecommunication networks means that one needs to produce the final results with a known (small) statistical error. Additionally, in all simulation studies aimed at assessing long–run performance of networks, the output data is highly correlated, so they have to be analyzed using special statistical techniques. Accurate methods to achieve statistical analysis of correlated simulation output data has attracted a considerable scientific interest and effort. All simulations were executed using AKAROA–2. AKAROA–2 can be either linked to PTOLEMY [159] or NS [175]. This means that the accuracy of all estimates was automatically assessed sequentially during simulation, and the simulation was stopped when statistical errors of the results reached an assumed accuracy. That was measured by the relative statistical error, defined as the ratio of the half–width of the confidence interval and the point estimate, at a given confidence level (see Appendix B). Conducting steady–state simulation, we have used a standard procedure of AKAROA–2 for sequential determination of the length of the initial transient stage, based on a sequential version of Schruben's test for testing stationarity of time series; see [176] and [177]. On the other hand, output data in steady–state was analyzed by applying the method of **O**verlapping **B**atch **M**eans (*OBM*) , proposed by Meketon and Schmeiser [178]. To speed up computationally our intensive stochastic simulation we applied one special scenario of stochastic simulation, known as **M**ultiple **R**eplications **I**n **P**arallel (*MRIP*). MRIP controls multiple processors operating as individual simulation engines. Each individual processor generates its output data and submits it to a global analyzer for on–line data analysis. The global analyzer is responsible for stopping the distributed simulation when the statistical error of results reaches the required level. The speedup for simulations in combination with PTOLEMY and AKAROA was shown in [179].

6.5 Visualization and Validation Tools

To verify the functionality of the different SMPT mechanisms a new visualization tool named `SMPTAnalyser` was implemented [3]. The tool is generally able to illustrate discrete events over time. Simulation tools, e.g. PTOLEMY, are writing some debug information in a log file with the following syntax:

TIME:ENTITY:STATE:INFO

Thus, every time instant an entity changes its state it writes an information into the log file. Furthermore it is possible to add an information value. In the following Table 6.2 a short example of the output file is given for three entities.

Table 6.2: Excerpt of a `SMPTAnalyser` trace file.

TIME	ENTITY	STATE	INF
106.001	Cno1	Send	45
106.001	Cno2	Send	46
106.001	Cno3	Send	47
106.501	Cno1	Free	45
106.501	Cno2	Free	46
106.501	Cno3	Free	47
107.001	Cno1	Send	45
107.001	Cno2	Wait	46
107.001	Cno3	Wait	47
107.501	Cno1	Free	45
107.501	Cno2	Free	46
107.501	Cno3	Free	47
108.001	Cno1	Send	48
108.001	Cno2	Send	46
108.001	Cno3	Send	47

The `SMPTAnalyser` tool allows the user to choose any of the log files using the `File` Menu. The tool extracts automatically the number of entities. The user can look at all entities or even select some set of entities that he wants to view using the `View` Menu. In Figure 6.12 a screen shot of the `SMPTAnalyser` for one wireless terminal with three parallel channels is depicted. The entities are called `Cno1`, `Cno2`, and `Cno3`. Only the uplink actions are displayed. The idle times are marked with white color, the transmission is blue, and the wait[4] times are colored orange.

Even if we display here only one WT, this tool allows us to observe multiple SMPT WTs over a small and a wide time scale.

[3] The work was done during a student project at TKN in winter 1998/99.

[4] The wait state is for situations, where a wireless terminal was already sending on additional channels and have to wait until the probing phase is turned off.

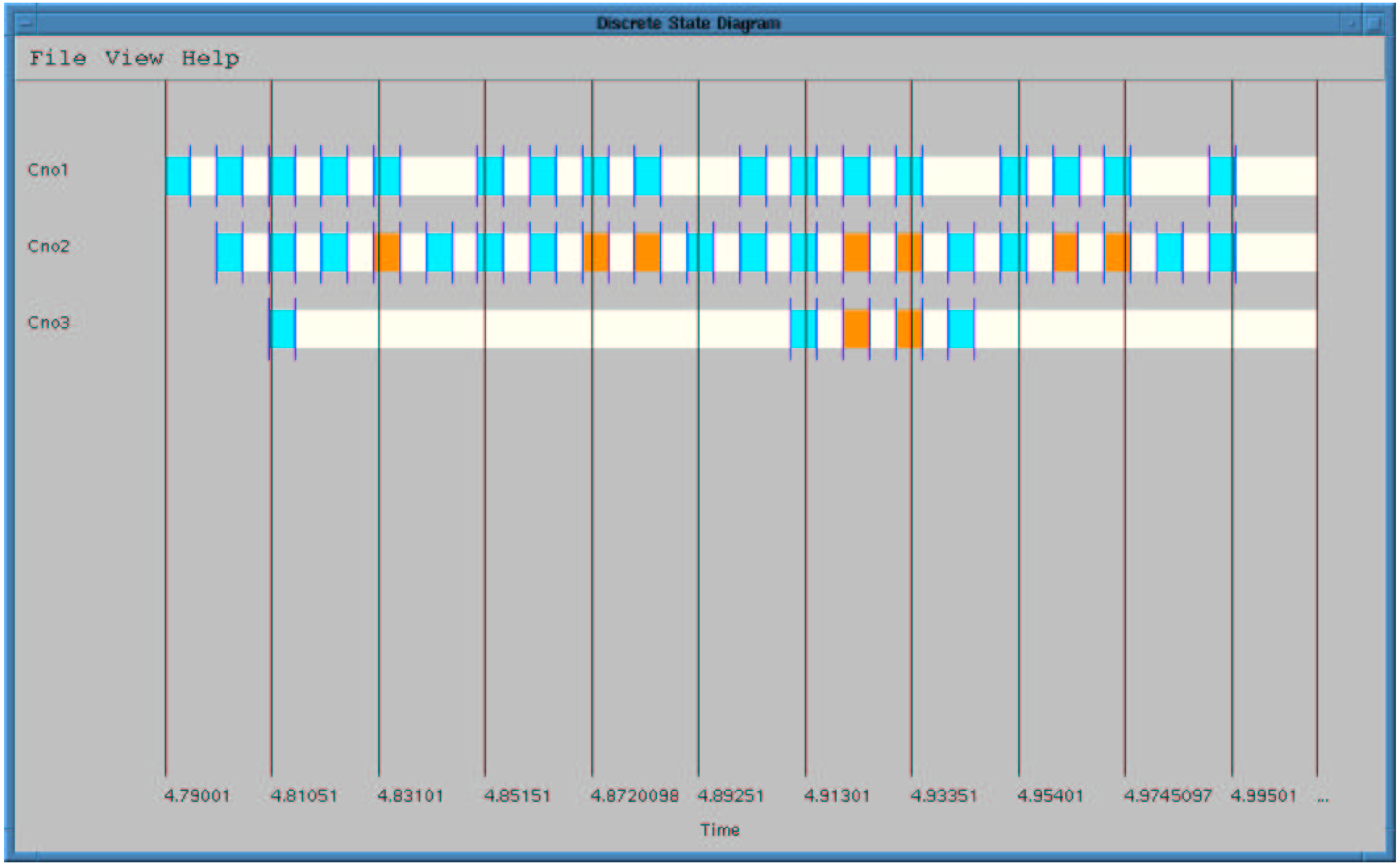


Figure 6.12: Screen shot of the `SMPTAnalyser` for one wireless terminal with three parallel channels.

7 Analytical based Performance Evaluation

In this chapter a mathematical derivation of the theoretical capacity (number of **W**ireless **T**erminals (*WTs*) with a given activity factor under given **Q**uality **o**f **S**ervice (*QoS*)) that can be supported in one single **C**ode **D**ivision **M**ultiple **A**ccess (CDMA) cell employing the **S**imultaneous MAC **P**acket **T**ransmission (*SMPT*) protocol is illustrated. The results presented in this chapter are based on our own results given in [180]. Since mobile communication systems based on the **S**ingle–**C**ode CDMA (*SC–CDMA*) approach have been dominated by voice services (homogeneous QoS), the calculation of the capacity was easy. It is claimed in [181] that each voice–oriented WT had an *activity* factor of 3/8 and that losses in a certain range could be tolerated. The activity factor is defined as the ratio of the time a medium is used to the time a certain medium is occupied (e.g. for telephone calls it is the ration the customer speaks to the call duration time). For future mobile communication systems the calculation of the capacity and the appropriate *activity* factors becomes more difficult, if we consider packet oriented data and real–time services. For these services *activity* factors can be hardly derived.

We are interested in the capacity of SMPT based mobile communication systems. Therefore we assume a single cell CDMA system with N WTs for the following considerations. Each WT is able to transmit and to receive on R **M**ulti–**C**ode CDMA (*MC–CDMA*) channels. The activity factor a describes the probability that wireless terminal i is active (i=1 . . . N). The number of active CDMA channels per cell is given by j. We assume that an overall number of J_{max} CDMA channels can be used within a single CDMA cell without exceeding a specific **S**ignal to **N**oise **I**nterference **R**atio (*SNIR*) value. A situation where more than J_{max} channels are used is referred to as *outage*. For telecommunication systems, the outage probability is usually used as the criterion to assess the user capacity [53, 181]. We assume that in case of an outage no communication by any WT is possible. The outage probability has to be limited and has to be smaller than ϵ.

$$Prob(j > J_{max}) \leq \epsilon. \tag{7.1}$$

For the SMPT approach we assume that each of the channels of one specific WT exhibit the same channel properties, i.e. if parallel channels are used either all or no channels are prone to errors. The channel model considered in this analysis is the one proposed by Gilbert–Elliot. We assume that no communication is possible within the bad channel state and that in the good channel state no error occurs as long as the number of used channels in the cell is less than J_{max}. If the number of channels is greater than J_{max} even in the good channel state no communication is possible. We made this assumption to keep the analytical model easy. The WT acts without the knowledge of the total number of used channels j within the cell. Perfect power control is assumed for all WTs.

7.1 Results

The system properties and the various mathematical formulas derived in [180] were used for the numerical computation of the capacity in the system. The computations were performed for three SMPT schemes (namely Basic SMPT, Slow Healing and Fast Healing), for various activity factors, for various values of R, which is the maximum number of channels supported by each user and for different values of ϵ which could correspond to tolerance values against overload situations. The maximal number of WTs in the cell without an outage is set to $J_{max} = 10$. The parameters of the wireless channel used in this work are as tabulated in Table 6.1.

In Figure 7.3 the capacity of the sequential and three SMPT schemes is presented for different values of ϵ=$\{0.0001, 0.001, 0.01\}$ and R. For the *Slow Healing* approach the capacity is given in Table 7.1. It can be seen that a higher number of parallel channels per wireless terminal results in a smaller cell capacity. For smaller activity factors the degradation in cell capacity becomes smaller. For 100% load ten WTs can be supported with the sequential transmission mode and only four WTs with the SMPT approach (ϵ=0.01). A reduced activity factor of 0.1 enables 50 WTs for the sequential and 37 for the SMPT transmission mode. While for the full load scenario the degradation goes down to 40%, the degradation for the low load scenario shrinks down to 74%. Table 7.1 reflects the situation that SC–CDMA systems allow more WTs in the cell than MC–CDMA. The reason why we want to use MC–CDMA is the better quality of the wireless link. Even if we loose capacity at the link layer perhaps in combination with higher protocol layers these drawbacks vanish. The higher quality level for MC–CDMA will be presented in Section 7.1.4. The results for the basic or the *Fast Healing* SMPT approach are basically the same. The full results are given in [180].

For a fair comparison the capacity is not the only parameter we have to look at. Additionally the loss probability and the delay of the packets have to be considered. All SMPT mechanisms have zero losses and the delay decreases with an increasing amount of parallel channels per wireless terminal. For the sequential transmission mode two modes are possible: 1.) In case of errors this packet is lost and the next stored packet will be transmitted. In this case we have a loss rate but no delay for the successful transmitted packets. 2.) All erroneous packets are retransmitted. Thus, the loss rate is zero (assuming an unlimited buffer), but the delay of the packets will increase. For retransmissions and a limited buffer a higher delay and losses occur.

7.1.1 Effect of ϵ

ϵ is the tolerance level of the application. It is the acceptable probability that the interference exceeds a maximum value. Few applications over transmission protocols such as UDP can tolerate certain high probability of error. Hence, these applications could choose higher values for ϵ such as $\epsilon = 0.01$. On the other hand, certain applications can tolerate very less probability of error and hence these applications require almost always, the interference to be below the threshold. These applications could choose $\epsilon = 0.001$. In general a larger value of ϵ leads to a higher capacity.

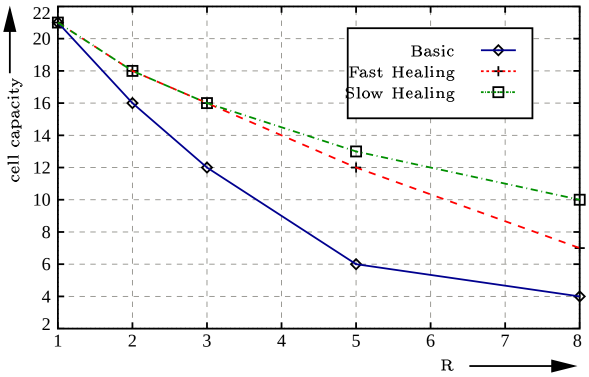


Figure 7.1: Cell capacity versus parallel number of channels per wireless terminal R for different SMPT schemes (ϵ=0.001, activity = 20%).

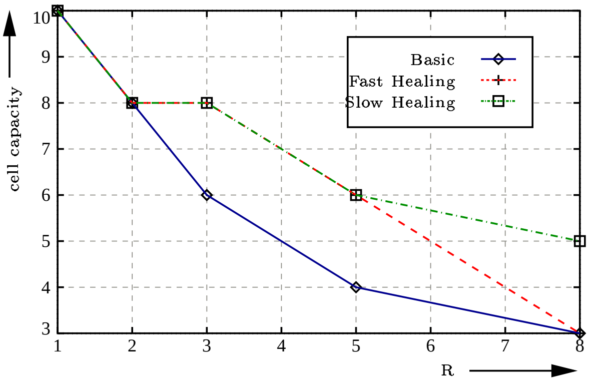


Figure 7.2: Cell capacity versus parallel number of channels per wireless terminal R for different SMPT schemes (ϵ=0.01, activity = 80%).

Table 7.1: Capacity of the system incorporating *Slow Healing* SMPT mechanism.

activity	$\epsilon = 0.01$					$\epsilon = 0.001$					$\epsilon = 0.0001$				
	r=1	r=2	r=3	r=5	r=8	r=1	r=2	r=3	r=5	r=8	r=1	r=2	r=3	r=5	r=8
0.1	50	45	43	40	37	38	33	30	25	19	30	25	21	15	10
0.2	26	23	22	21	19	21	18	16	13	10	17	14	12	8	5
0.3	18	16	16	14	13	15	13	12	9	7	13	10	9	6	4
0.4	15	13	12	11	10	12	10	9	7	5	11	9	7	4	3
0.5	12	11	10	9	8	11	9	8	6	4	10	8	6	4	3
0.6	11	10	9	8	7	10	8	7	6	4	10	7	6	3	3
0.7	10	9	8	7	6	10	8	6	5	3	10	7	6	3	2
0.8	10	8	8	6	5	10	7	6	5	3	10	7	6	3	2
0.9	10	8	7	6	5	10	7	6	4	3	10	6	6	2	2
1.0	10	8	6	6	4	10	7	6	4	3	10	6	5	2	2

7.1.2 Effect of R

R corresponds to the maximum number of simultaneous channels that a user could use to clear his backlog packets in the SMPT schemes. The advantage of using higher R lies in the fact that the delay or the delay jitter, that would have been caused because of the channel errors, are reduced. The higher the number of channels the better it is for delay intolerant applications. However the price paid for the usage of a higher number of channels is that the number of users supported in the cell is reduced. This can be easily seen in Figure 7.1 and 7.2. In general a larger value of R leads to a smaller capacity, but to better QoS support.

7.1.3 Effect of the Different SMPT Schemes

Figures 7.1 and 7.2 compare the effect of the different SMPT schemes on the system capacity for different activity factors of 20% and 80%. A common behavior in all these curves is that with an increased number of parallel chancels per wireless terminal, R, the capacity decreases. From the figures it can be easily observed that from the system capacity point of view, *Slow Healing* is better than *Fast Healing* which in turn is better than the basic scheme. The reason that SH performs better than the other schemes is that it adapts on the cell capacity more than the other schemes.

7.1.4 Impact on Buffer Overflow

In practical systems, the buffers are of finite length. Mainly there are two reasons for finite length buffers: 1.) A higher buffer length will introduce more complexity and more costs for the realization of a WTs. 2.) For real–time communication a limited buffer is one possibility to realize a bounded delay. Buffer overflow can occur if the buffer is too small. On the other side erroneous packets have to be retransmitted while no packet is used from the buffer. The probability of buffer overflow is one performance metric in our analytical model considering that erroneous packets has to be retransmitted. The lower the probability of buffer overflow, the better is the system.

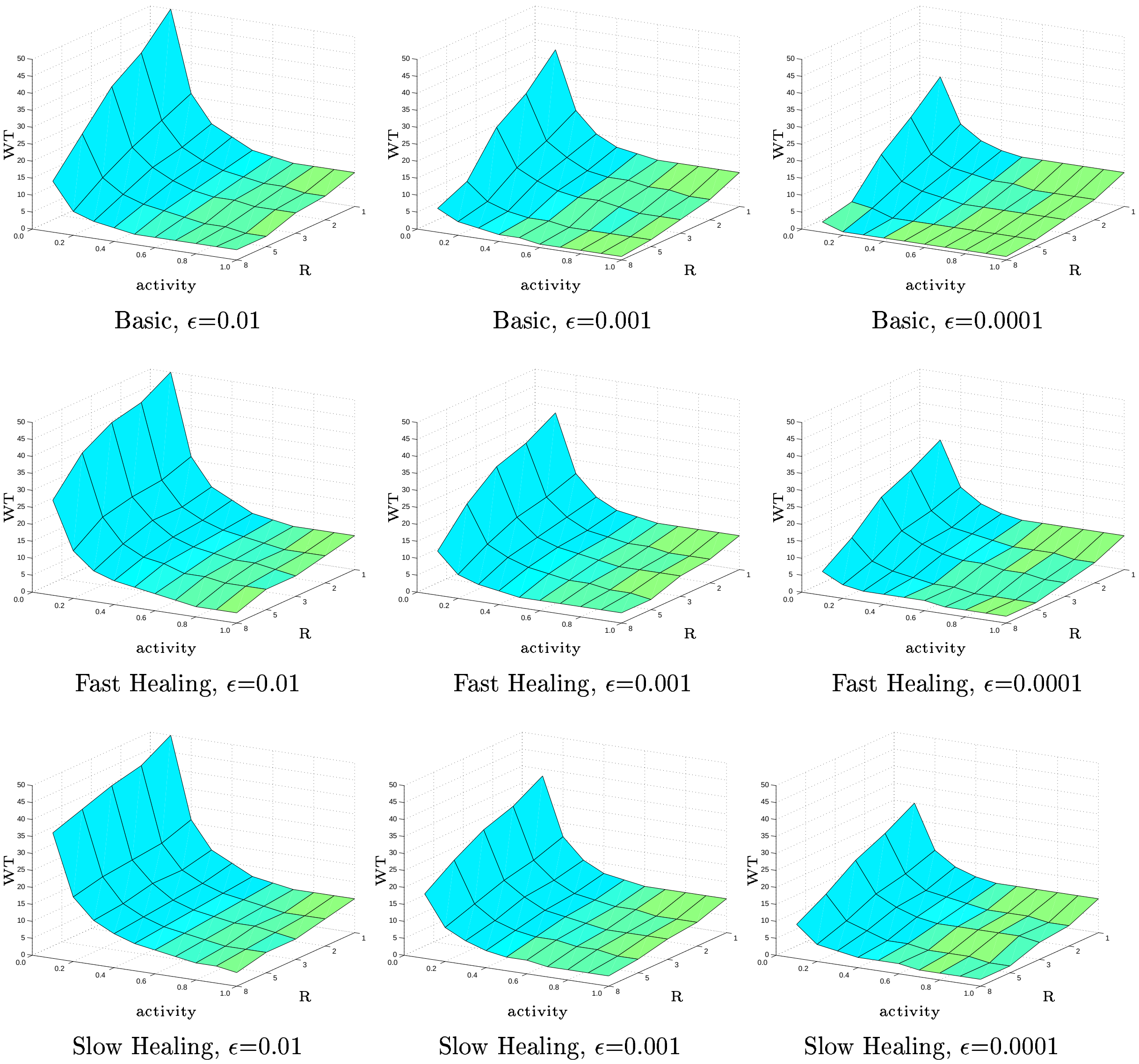
The metric used to quantify the buffer overflow β is evaluated for different SMPT schemes and for different values of R, which is the maximum number of simultaneous channels used in the system. The results are tabulated in Table 7.1.4. It can be noticed that in the case of the *Slow Healing* SMPT mechanism, there is not much reduction in β when the value of R is increased beyond three. However, it can be seen that by using SMPT (already with R=2) technique, the value of β reduces by a huge margin.

Table 7.2: Metric quantifying the buffer overflow tabulated for different SMPT schemes and different values of R.

B_{max}	1000	100
Sequential $R = 1$	9.35e-03	1.13e-01
SMPT SH $R = 2$	1.96e-39	1.26e-04
SMPT SH $R = 3$	1.40e-40	1.09e-04
SMPT SH $R = 5$	4.08e-41	1.03e-04
SMPT SH $R = 8$	3.18e-41	1.02e-04

The buffer overflow probability for buffer length storing 100 LPDUs is approximately

10% for the sequential and 0.1% for the SMPT transmission mode. If we transfer buffer overflow probability directly to loss probability SMPT yields smaller loss probabilities. Jitter values for higher protocol segments are smaller for SMPT than for the sequential case.



Basic, ϵ=0.01

Basic, ϵ=0.001

Basic, ϵ=0.0001

Fast Healing, ϵ=0.01

Fast Healing, ϵ=0.001

Fast Healing, ϵ=0.0001

Slow Healing, ϵ=0.01

Slow Healing, ϵ=0.001

Slow Healing, ϵ=0.0001

Figure 7.3: Capacity of different SMPT approaches versus activity factor a and number of supportable channels per WT R (R=1 sequential transmission approach).

7.2 Conclusion

The results presented in this section draw the following conclusions: SC–CDMA has a larger capacity than MC–CDMA for all values of the activity factor and number of parallel channels offering lower QoS in terms of loss rate or delay in comparison to SMPT.

Furthermore *Slow Healing* SMPT has the best performance among the SMPT approaches regarding the capacity. Taking the buffer overflow and the buffer occupancy as the QoS metric SMPT as a member of the MC–CDMA family performs better than SC–CDMA.

Note, in the simulation we follow strictly the SMPT approach presented in Chapter 4 that parallel channels will not be used if any error occurred meanwhile building the ramp on the wireless link. In the analytical case ramps will be build up even if an error occurs on the wireless link. This helps us to keep the analytical model simple, but introduces a kind of inconsistence between simulation and analytical consideration. Note, this inaccuracy leads to a significant smaller capacity for all SMPT schemes.

8 Simulation Based Performance Evaluation

This chapter is structured as illustrated in Figure 8.1. By simulations we contribute a performance evaluation for **S**imultaneous **M**AC **P**acket **T**ransmission (*SMPT*). The work is divided into two major parts. The first part is based on the **T**ransport **C**ontrol **P**rotocol (*TCP*). For this part two different environments are taken under consideration. In the first case we investigate the behavior of the TCP throughput and the TCP congestion window for one sender receiver pair. In the next case results are obtained for the **F**ile **T**ransfer **P**rotocol (*FTP*) and video applications for multiple sender receiver pairs. For both types of applications the cell capacity (number of customers that can be supported) and some QoS parameters such as TCP throughput, buffer underrun, and number of active channels are measured. The second main part is based on the **U**ser **D**atagram **P**rotocol (*UDP*). Two different types of applications are taken under consideration for multiple sender receiver pairs. One application is the **C**onstant **B**it **R**ate (*CBR*) source where the impact of SMPT on the power control entity, inter and intra–cell interference, and the cell capacity is investigated. For the video applications, distinguishing H.263 and MPEG–4 encoded video, QoS parameters such as jitter and losses are more important. Even for video applications the cell capacity is measured.

For the performance evaluation we chose the following metrics: 1.) The *goodput* value is defined as all LPDU packets that belongs to a completely transmitted video frame (or equivalently UDP segment). Higher goodput values reflect a situation where the system performs better. 2.) At the receiver the *losses* of video frames are counted. Only complete video frames (all LPDU packets are received successfully) are useful. Thus, the losses are the difference between all generated video frames and all complete received video frames. The lower the loss the better the performance of the system. 3.) For all completely transmitted video frames we measure the *jitter* value. Lower jitter values are desirable. 4.) If a complete video frame was received at the receiver–side within a given **T**ransmission **W**indow (*TW*) this video frame is referred to as a successful transmitted video frame. The higher the ratio of successfully transmitted video frames, the better the system performs. 5.) For a network provider the *capacity* or number of WTs that can be supported while guaranteeing a pre–specified QoS requirement is extremely relevant. The network provider always wants a large number of WTs in a cell. 6.) The mean and the variance in code usage have a large impact on the intra–cell and inter–cell interference [18], respectively. Smaller values lead to a better system performance.

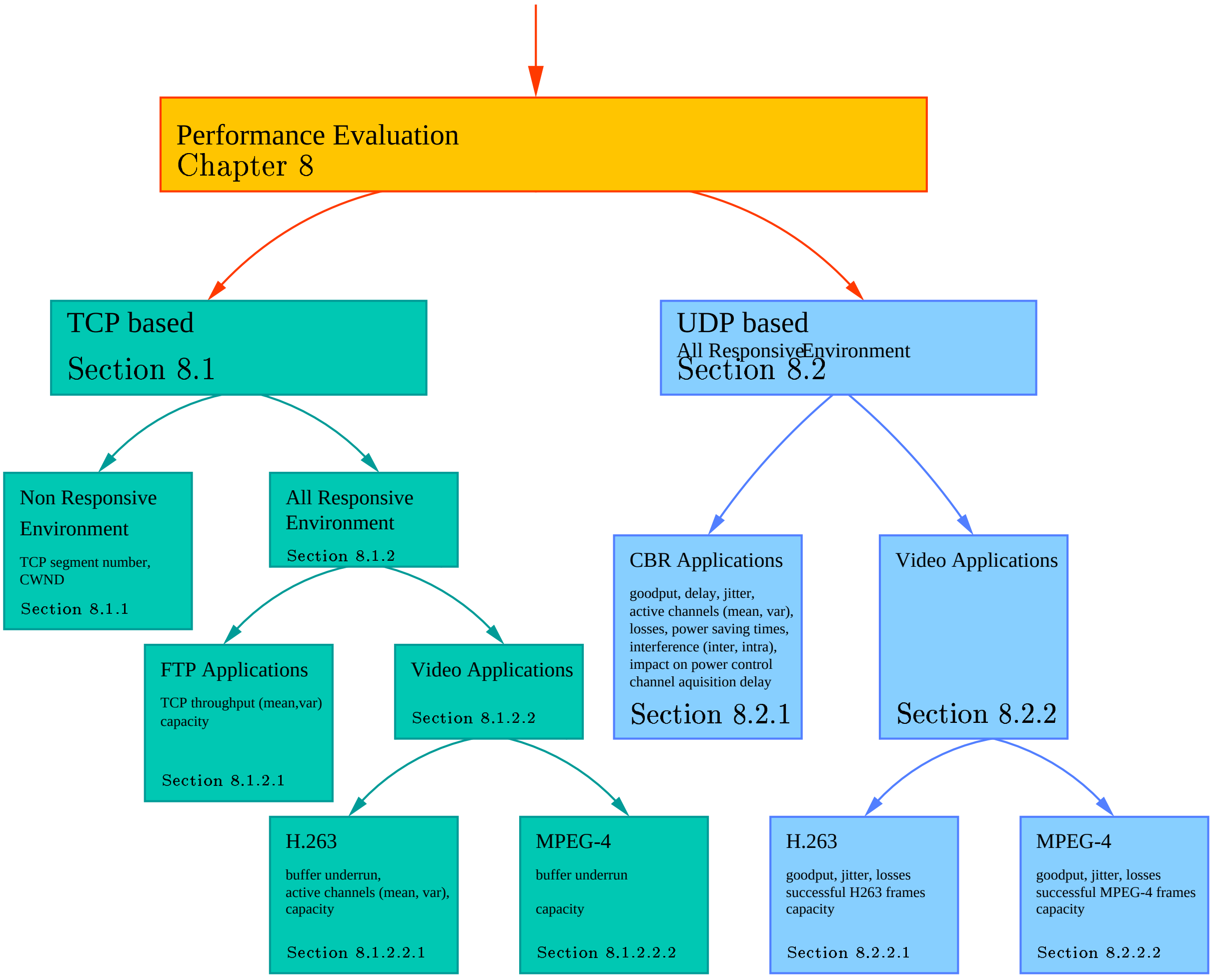

Performance Evaluation
Chapter 8
TCP based
Section 8.1
UDP based
All Responsive Environment
Section 8.2
Non Responsive Environment
TCP segment number, CWND
Section 8.1.1
All Responsive Environment
Section 8.1.2
CBR Applications
goodput, delay, jitter, active channels (mean, var), losses, power saving times, interference (inter, intra), impact on power control channel aquisition delay
Section 8.2.1
Video Applications
Section 8.2.2
FTP Applications
TCP throughput (mean,var) capacity
Section 8.1.2.1
Video Applications
Section 8.1.2.2
H.263
buffer underrun, active channels (mean, var), capacity
Section 8.1.2.2.1
MPEG-4
buffer underrun
capacity
Section 8.1.2.2.2
H.263
goodput, jitter, losses successful H263 frames capacity
Section 8.2.2.1
MPEG-4
goodput, jitter, losses successful MPEG-4 frames capacity
Section 8.2.2.2

Figure 8.1: Structure and metric of the performance evaluation.

8.1 Support of TCP Traffic over SMPT

In this section we provide a performance evaluation of TCP over SMPT. The performance evaluation is done by using two different scenarios. In the first scenario we consider the *Non–Responsive* scenario, where we investigate only one sender receiver pair with a constant interference level. The interference of additional channels is not taken under consideration. In the second scenario we consider multiple sender receiver pairs. This time the interference of additional channels is taken under consideration. Henceforth we call this the *All–Responsive* scenario. For the *All–Responsive* scenario the performance gain of SMPT for data services (FTP) and video services (MPEG–4 and H.263) is shown by means of simulations. Because we concentrate on the uplink transmission we study the streaming of video of a cellular **C**ode **D**ivision **M**ultiple **A**ccess (CDMA)–based wireless system. It is widely recognized that the TCP typically gives poor performance in wireless environments. It is also well known that TCP has many desirable properties (such as network stability, fairness). One of these desirable properties, *TCP–friendliness*, has emerged as an important property because of the dramatical growth in video streaming services. In this section we advocate the use of TCP as the transport layer protocol when streaming video in a **M**ulti–**C**ode CDMA (*MC–CDMA*) wireless system. Our approach is to stabilize the TCP–throughput for video transmissions over the wireless links by employing the SMPT approach. Our extensive simulations indicate that video streaming over TCP gives good performance when SMPT is employed.

8.1.1 Non–Responsive Interference Environment

In this section we consider the TCP performance over a wireless link for SC–CDMA and MC–CDMA systems for the *Non–Responsive* scenario. We consider one WT that is transmitting data to the base station using a FTP application (note that in order to show the impact on TCP, FTP is the appropriate traffic source, which has always something to send). On the wireless link no bursty errors are assumed. The error is modeled with constant error probability. Thus no **M**ultiple **A**ccess **I**nterference (*MAI*) interference is under investigation for this first simple scenario. For the simulations we used principally the parameters given in Table 8.1 without taking the Gilbert–Elliot Model into account. The only difference is the number of wireless terminals (set to one for this experiment) and that we assume a static bit error probability in the *good* channel state.

The most important parameter for FTP applications is the TCP throughput. In Figure 8.2 we plot the sequence number over the time [1] for the SMPT and the sequential transmission approaches for three different **P**acket **E**rror **P**robability (*PEP*). While the sequential approach gives smaller TCP throughput as the PEP increases, the SMPT approach is stable over a wide range of the PEP (10^{-6} up to 10^{-2}). There are two reasons for the increase in TCP throughput when using SMPT. Primarily, the SMPT approach offers more bandwidth than the sequential case. The second reason is that *spurious* retransmission of TCP segments can be avoided. These spurious retransmission takes place every time the **C**ongestion **W**indow (*CW*) shrinks down. We observe further that for the SMPT approach the sequence number increases steadily. For the sequential case we notice some collapse of the sequence number. This indicates that the SMPT approach has a stabilizing effect on the wireless link. SMPT stabilizes the throughput by overcoming the fast–time scale variations (typically on the order of tens of milliseconds) of the wireless channel. This is important if we later focus on video encoders, which can only react to the available channel bandwidth on a longer time-scale (typically on the order of hundreds of milliseconds or seconds).

Figure 8.3 shows the CW size over time in case of a SMPT and sequential transmission approaches for three different packet error probabilities. It can be seen that the SMPT approach in contrast to the sequential transmission approach never shrinks its CW. With a low PEP the congestion window is still high for the sequential transmission. But an increased PEP let shrunk the CW more often. The CW behavior of SMPT is the reason for stabilized throughput on the wireless link. The TCP segments can always be transmitted within the pre–calculated **R**etransmission **T**ime **O**ut (*RTO*).

Figure 8.4 shows the TCP throughput versus the maximal number of CDMA channels

[1] The TCP throughput is defined as the time derivative of the sequence numbers.

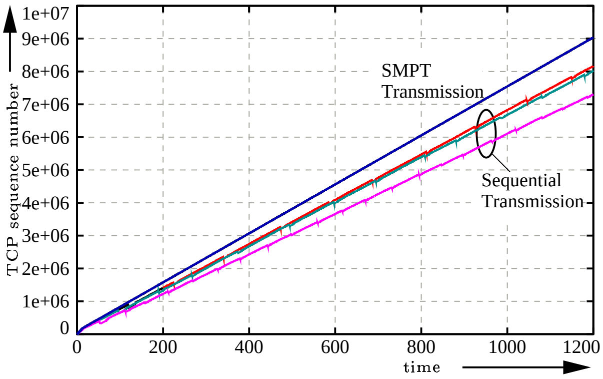


Figure 8.2: TCP sequence number versus time for different error probabilities.

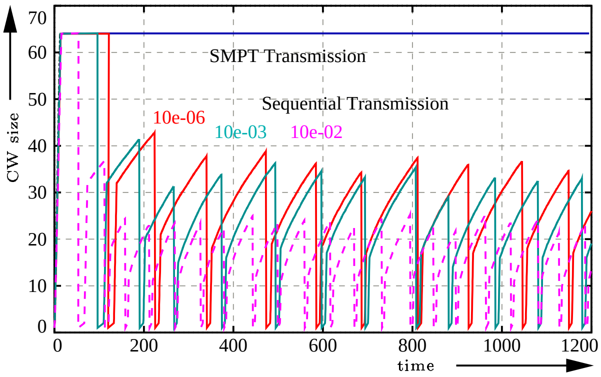


Figure 8.3: TCP congestion window size versus time for different packet error probabilities.

R_{max} per WT. We observe that for the *Non–Responsive* environment a number of parallel channels $R_{max} \geq 3$ per WT will not lead to any improvement. For a wide range of PEP the SMPT approach offers nearly the same TCP throughput values. Only the sequential case ($R_{max} = 1$) results in variable TCP throughput values in dependency on the PEP. If we assume a variable bit error probability on the wireless link, the advantage of SMPT versus the sequential transmission is a stabilized TCP throughput.

In Figure 8.5 the influence of the bad channel sojourn time for the TCP throughput is given. With a higher number of parallel channels the TCP throughput can be stabilized longer and leads to a higher TCP throughput. Longer bad channel sojourn times decrease the TCP throughput for all transmission approaches. The higher the bad sojourn times are the smaller is the SMPT gain. This behavior can be explained if we remember that SMPT tries to stabilize the QoS parameters within one higher protocol segment. If the bad channel sojourn time is larger than such a segment, the SMPT approach suffers nearly in the same way as the sequential transmission mode.

We conclude that for the *Non–responsive* environment SMPT yields good results for the TCP throughput. Independent of the bit error probability the TCP throughput gets stabilized. In real systems there will be a reaction on the channel allocation. Therefore we will focus on a more realistic scenario in the next section and investigate if there is still a SMPT gain.

8.1.2 All–Responsive Interference Environment

In this section we investigate the *All–Responsive* interference environment for two different types of traffic sources. *All–Responsive* means that the interference within the CDMA cell is produced by J WTs transmitting towards the base station. Each additional CDMA channel will lead to an increased interference in the cell. Thus, once again the errors are modeled by the Gilbert–Elliot model, but in this section the errors in the good state depends on the interference level in the cell. As described in Section 2.3.3.1 mechanisms are found that channels of the same WT will not interfere with each other. We investigate the FTP application and then focus on video streaming applications. We use the H.263

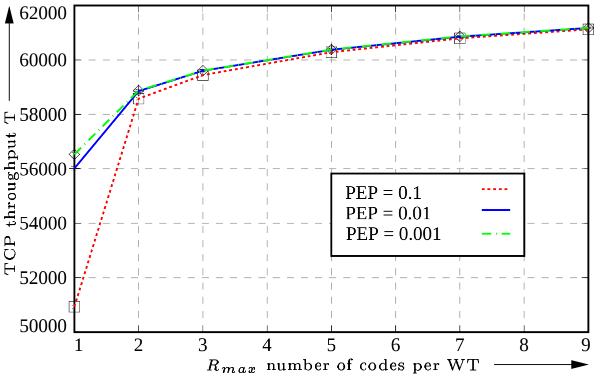


Figure 8.4: TCP throughput versus the maximal number of CDMA channel R_{max} per WT for different packet error probabilities and 100 ms bad channel sojourn time.

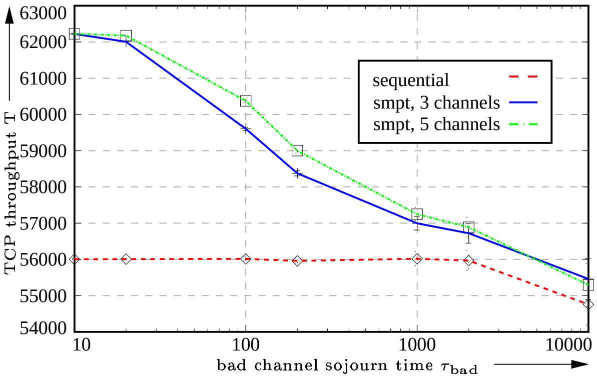


Figure 8.5: TCP throughput versus the bad channel sojourn time τ_{bad} for $p = 10^{-2}$ for different multiple channels R.

and the MPEG–4 video encoding schemes and show the difference of rate controlled and variable bit rate video streams. We will investigate the capacity of CDMA systems for single and multi code CDMA.

8.1.2.1 FTP based Simulation

For the first FTP scenario the bit rate of one wireless CDMA channel is set to 64.8 kbps. Parameters for the simulation are given in Table 8.1. In Figure 8.6 the TCP throughput versus the number of WTs is given. We compare the sequential case with the *Slow Healing* SMPT for different values of $R_{max} = \{2, 3, 8\}$. It can be seen that for small number of WTs the SMPT approach results in a higher TCP throughput than the sequential transmission technique. The sequential transmission has throughput values of 56.4 kbps up to 13 WTs. For SMPT with R_{max}=2 the throughput increases up to 59.8 kbps. For higher values of R_{max} the throughput increases further up to 60.9 kbps. But for SMPT this high throughput can be offered only for up to ten WTs. The set of WTs where the throughput is stable is defined as *operational* phase.

Figure 8.7 shows the standard deviation of TCP throughput σ_{TCP} versus the number of WTs. We again compare the sequential case with the SMPT case for different values of $R_{max} = \{2, 3, 8\}$. In addition to these curves the standard deviation of TCP throughput σ_{TCP} for an error–free channel is depicted. The value of σ_{TCP} for an error–free channel is not zero, because of the way we measure the TCP throughput[2]. For the *operational* phase the SMPT approaches offer half the standard deviation values than the sequential case. If three or more parallel channels are available σ_{TCP} reaches the minimum.

For this scenario we conclude that a total number of parallel channels per WT R_{max}

[2] The TCP throughput is measured in 10 sec intervals. Because of this fixed time intervals and the feature of TCP to acknowledge a group of TCP segments it can happen that the TCP throughput varies even for the error-free scenario.

Table 8.1: Simulation parameters for TCP–based FTP transmission

Employment	Parameter	Value
Scenario	number of Wireless Terminals J	1 – 26
Application	Type	FTP
Transport Layer	Segment size L_{TCP} [bytes]	1400
	TCP header [bytes]	20
Network Layer	L_{buffer} [segments]	10000
	IP header [bytes]	20
Data Link Layer	Packet size L_{MAC} [bytes]	80
	SMPT policy	Slow Healing
Physical Layer	Slot length $\tau_{\text{frame}} = \frac{L_{\text{MAC}}}{C}$ [ms]	10
	number of Available Channels R	1, 3
	Bit Rate C [kbps]	64.8
	Spreading Gain	16, 32
	Frame size [bits]	1023
FEC code	Payload [bits]	648
	Redundant [bits]	375
	Correctable errors	30
Wireless Link	Bad state duration τ_{bad} [ms]	100
	Good state duration τ_{good} [ms]	1000
	Bad state p_{err}	1.0
	Good state p_{err}	Improved Gaussian Approximation
	Background channels b	2,4
Simulation	Confidence Level CV	99%
	Measure interval τ_{tic} [s]	10

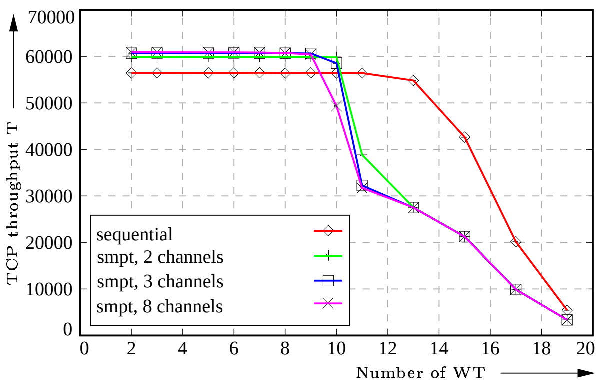


Figure 8.6: TCP throughput versus the number of wireless terminals.

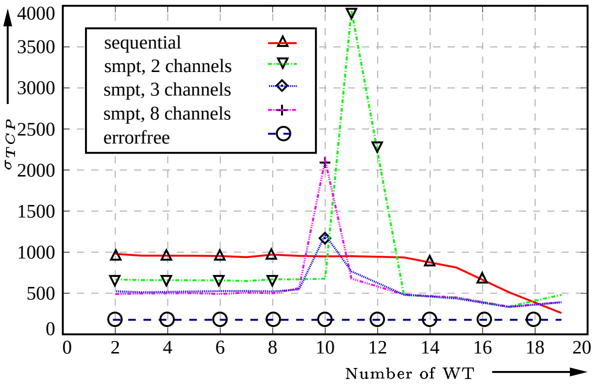


Figure 8.7: Standard deviation of TCP throughput versus the number of wireless terminals.

equal to three offers both, a high number of supportable WTs (up to 10) and a small value of σ_{TCP} (500bit/sec). The sequential case can support more WTs (up to 13), but has smaller TCP throughput values with a higher standard deviation σ_{TCP} (1000bit/sec). As we claim in Section 2.2, a small variance in throughput would be desirable for video applications.

With the second FTP scenario we want to investigate the overall capacity of the CDMA system in dependency of the spreading gain. As system parameters we assume a bit rate of 72.2 kbps and two different spreading gains $G_{Spreading}$=16 and $G_{Spreading}$=32. In Figure 8.8 the TCP throughput versus the number of WTs within the CDMA for a spreading gain $G_{Spreading}$=16 is given. As in the first FTP scenario, SMPT is offering higher throughput values for a smaller operational phase than the sequential case. SMPT achieves the highest system throughput with J=8 WTs resulting in an overall throughput of 520 kbps. The sequential transmission approach reaches the highest system throughput with J=11 WTs resulting in an overall throughput of 638 kbps. If we now increase the spreading gains $G_{Spreading}$=32, we see in Figure 8.9 that the operational phases for both approaches are nearly the same and that SMPT has an higher throughput. Thus, an higher overall throughput can be achieved by using SMPT.

The reason that SMPT performs better with a higher spreading gain is the **P**acket **E**rror **P**robability (*PEP*) for these two different scenarios ($G_{Spreading}$=16 and $G_{Spreading}$=32) given in Figure 8.10. In both cases a BCH(1023,728,30) [83] code was used. For $G_{Spreading}$=16 the PEP changes from 10^{-3} to 10^{-1} in the range of three channels, while for $G_{Spreading}$=32 this range increases up to five channels. Therefore, we claim that the slower the PEP increases the better SMPT performs. To illustrate this behavior we can take a PEP bang–bang[3] function. In this case the sequential case performs at its best, because the system stays in the *good* range of the PEP curve until the number of channels enter the *bad* range of the PEP curve. However, in the case of SMPT even though the number of WTs is much smaller than the capacity of the sequential transmission, there exists many instances when the SMPT approach enter the *bad* range of the PEP curve. Thus,

[3] ϵ (t)=1; t $\geq$ 1 else ϵ (t)=0

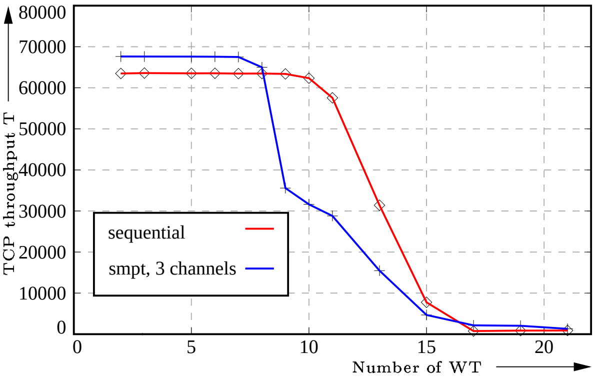


Figure 8.8: TCP throughput versus the number of WTs within the CDMA for a spreading gain $G_{Spreading}$=16.

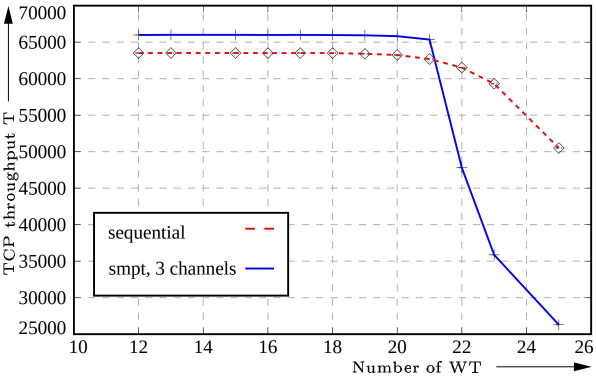


Figure 8.9: TCP throughput versus the number of WTs within the CDMA for a spreading gain $G_{Spreading}$=32.

a higher spreading gain, allowing a higher multiplexing effect, results in better SMPT performance.

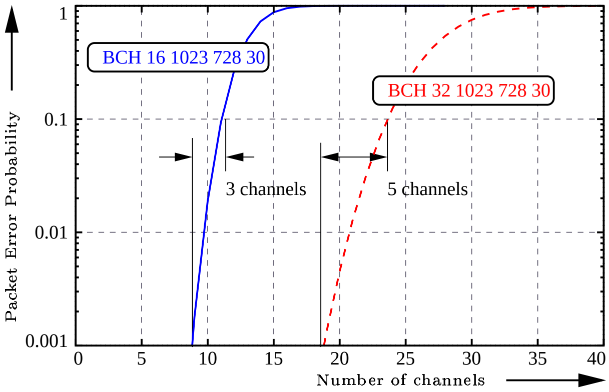


Figure 8.10: Packet Error Probability for a BCH code (1023,728,30) with two different spreading gains $G_{Spreading}$=16, 32.

We have shown the effect on the FTP throughput using the *Slow Healing* SMPT approach. But for FTP session there are no real QoS requirements by the user. Therefore, we will investigate the performance of video applications over TCP.

8.1.2.2 Video Source

Before we start to investigate the SMPT performance for video applications, we give a short motivation why we find video streaming over TCP promising. Market research finds that mobile commerce for 3G wireless systems and beyond will be dominated by basic human communication such as messaging, voice, and video communication [182]. Because of its typically large bandwidth requirements, video communication (as opposed to the lower rate voice and the elastic e–mail traffic) is expected to emerge as the dominant type of traffic in 3G/4G wireless systems [1]. Video services are typically divided into real–time services (e.g., video conferencing) and streaming (e.g., web–based streaming of a news clip or the video feed from a surveillance camera). Our focus in this section is on video streaming where the client may tolerate a small start–up delay before the playout commences. Video streaming schemes typically rely on UDP as the transport protocol [183, 184]. As streamed video applications become more popular, the Internet will become dominated by UDP streams. UDP streams, however, can lead to instability in the Internet. This is because UDP streams are not responsive to network congestion, as opposed to TCP streams [185, 186]. With the explosive growth of multimedia applications, UDP–based multimedia streams have the potential to cause two major problems in the Internet: 1.) congestion collapse and 2.) unfair allocation of bandwidth among competing traffic flows [187]. Therefore, TCP friendly video streaming schemes are desirable. Jacobs and Eleftheriadis [188] introduced a TCP friendly approach for streamed video. In contrast, in this section we advocate to use directly TCP as the transport protocol for video streaming. As is well known, using TCP as the transport protocol for video streaming in wireless environments leads to the well documented performance problems of TCP over wireless links. Our approach is to employ MC–CDMA in combination with the SMPT scheme [189, 21, 23] to stabilize the data link throughput by reducing losses and delay variations. Besides ensuring the stability of the Internet and achieving fair bandwidth allocation, TCP has a number of important advantages. First, TCP is reliable and ensures the lossless transport of the video stream. This is important for video streams that do not tolerate errors or error propagation, such as surveillance video. Another advantage of TCP is that it ensures the in–order delivery of the video frames which could also be achieved with **R**eal **T**ime **P**rotocol (*RTP*) using UDP). We also note that there are a number of drawbacks to using TCP as the transport protocol for video streaming. First, TCP does not support multi–cast. Secondly, TCP's slow start mechanism and its ARQ based recovery from packet losses may interfere with the timely delivery of the video frames. Essentially, TCP trades off increased delay for lossless transport service. We demonstrate in this section that by using SMPT at the link layer we can mitigate the interference of TCP's slow start and ARQ mechanisms with the timely delivery of the video frames. Our focus in this section is on the streaming of rate controlled encoded video in the uplink direction in a cellular wireless system, i.e., from wireless clients to a central base station. We focus on a reliable video streaming service that does not skip frames but instead suspends the playout at the receiver when the video consumption (temporarily) exceeds the video delivery. We provide extensive simulation results that demonstrate that our approach of combining SMPT at the data link layer and TCP at the transport layer supports video streaming in an efficient manner.

The problem of efficient video streaming over wireless links has attracted a great deal of attention recently. Several works, see for instance [190, 191, 192] attempt to improve the video quality by employing adaptive video coding schemes. The basic assumptions shared by all mentioned works are that the traffic source is based on H.263, real time services are applied, and the wireless link can be modeled with a two state Markov-Chain. To bound the time delay within an acceptable range for real-time video services the allowed maximum number of retransmission attempts are limited. In [190] it is further assumed that CDMA is applied as the air interface technology. Other works employ hybrid error correction, see for instance [193], to make the video transmission more robust. While the approaches pursued in this literature have made significant progress towards improving the efficiency of video streaming over wireless links, the issue of TCP friendliness has received very little attention. In fact, the proposed approaches rely largely on UDP as the transport protocol and typically have no mechanism to ensure TCP friendliness. In this section we propose a video streaming scheme that uses TCP as the transport protocol and is therefore by default TCP friendly. We also demonstrate that our scheme has favorable performance characteristics.

In [189, 23] a SMPT scheme for stabilizing the data link layer throughput over the wireless channel was introduced and studied for elastic data traffic. We now give a brief discussion of the impact of wireless link error on the TCP performance for elastic traffic. In this section we focus on the streaming of video over the wireless links in a single cell of a cellular wireless system. We consider the uplink streaming of rate controlled H.263 and non rate controlled MPEG–4 encoded video from J WTs to the base station. (We note that the streaming in the uplink direction is a particular challenge, as the WTs act in an independent, uncoordinated fashion; unlike the case of downlink streaming where the base station can coordinate the transmissions.) The base station acts as a receiver[4]. At the receiver side (i.e., base station) we assume a play–out buffer. In the simulations each WT randomly selects one out of 25 video sequences, which are obtained from [134] including sport, movie, and news video sequences. Also, each WT selects an independent random starting phase into the selected trace to ensure the statistical independence of the transported video streams. The WT commences the video streaming by filling the receiver–side play–out buffer to a pre–specified *offset* value τ_{off} (in units of time). The receiver side application starts to play out the video once the play–out buffer reaches the offset value. Under normal circumstances, for every frame period (which is typically an integer multiple of 40 msec for H.263 encoded video [134]) the receiver removes a frame from the play–out buffer, decodes it, and displays it. If at any of these epochs there is no complete video frame in the play–out buffer, the receiver experiences playback starvation, which we refer to a *buffer underrun.* When a buffer underrun occurs the receiver temporarily suspends the play–out of the video. The receiver waits until the play–out buffer is filled to the offset value, and then resumes the play out of the video. The average *interruption time* as well as the *buffer underrun rate* Ω depend on the *offset* value. While the interruption time can be approximately[5] calculated with

$$InterruptionTime = \frac{offset \cdot MeanFrameLength}{PhyLinkRate \cdot FrameSpacing}, \tag{8.1}$$

the buffer underrun rate Ω has to be evaluated by simulations. Note that no video frames are skipped when a buffer underrun occurs. This makes this reliable video streaming scheme well suited for applications that can tolerate short pauses in the video playback,

[4] As long as the wireless hop is the critical path we assume that the video streams are consumed within the base station. We note that the bandwidth of the back bones are normally that high, that all videos can be delivered to other terminals within the wired network in a timely fashion.

[5] Because of TCP it can not always be guaranteed that the provided bandwidth is used such as the situation after detecting collisions. Furthermore, the *PhyLinkRate* depends on the actual interference level.

but do not tolerate any loss of video frames, such as the video feed from a wireless surveillance camera. The duration of the buffer underrun (i.e., the suspension of the video playback depends on the bit rate of the video and the throughput of the TCP transport protocol at that particular instant). In our performance evaluation we study primarily the buffer underrun rate Ω (in buffer underruns per second) and the average time between buffer underruns T_Ω (in seconds). For higher values of T_Ω and lower values of Ω the systems performs better. We also investigate the inter–cell and intra–cell interference. All parameters used for this simulations are given in Table 8.2.

8.1.2.2.1 H.263 with Rate Control In this section we discuss the system behavior for 64 kbps rate controlled H.263 encoded video. To accommodate the overhead of the upper protocol layers and some retransmissions we chose a bit rate of 72.8 kbps for the physical layer. (Note that the over–provisioning of the wireless channels allows even the sequential transmission scheme to perform retransmissions.) In Figure 8.11 and 8.12 we plot the buffer underrun rate as a function of the number of WTs for the SMPT approach (R_{max} = 3) and the sequential transmission approaches with single (72.8 kbps) and double (145.6 kbps) bit rate. The offset value τ_{off} is set to 0.5 sec (Figure 8.11) and 1.0 sec (Figure 8.12). For the first set of simulations the spreading gain is set to 16. Even though the offset value has a significant impact on the buffer underrun rate, the two figures reflect the same overall behavior.

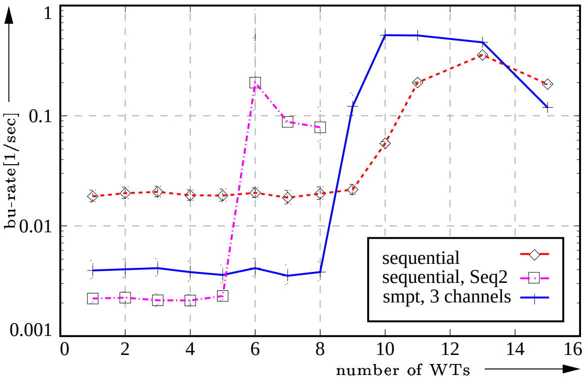


Figure 8.11: Offset 0.5 sec.

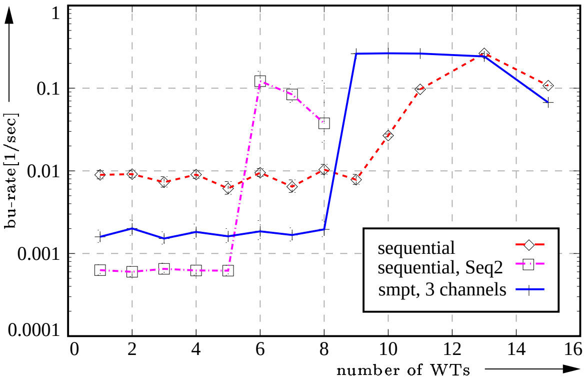


Figure 8.12: Offset 1.0 sec.

Buffer underrun rate versus number of WTs for the SMPT approach (R_{max}=3) and the sequential transmission approaches with single and double bit rate ($G_{Spreading}$ = 16).

First, we discuss the differences between the sequential transmission and the SMPT approach. The buffer underrun rate for the sequential case is nearly one order of magnitude larger than for SMPT for a certain range of the number of WTs (from 1 to 9). Within this range the buffer underrun rate is almost constant. We refer to this range as *operational phase*. For more WTs, i.e., to the right of the operational phase the buffer underrun rate increases dramatically. Note, that the operational phase of the sequential case contains one more WT (i.e., 9 WTs) than the operational phase of the SMPT approach (which is 1 to 8 WTs), but the user has to accept lower quality for the sequential case. The dramatic increase of the buffer underrun rate is the result of an increased usage

Table 8.2: Simulation parameters for TCP–based video streaming

Employment	Parameter	Value
Scenario	number of Wireless Terminals J	1 – 26
Application	Type	Video
	Encoder	H.263 rate control
	Bit Rate [kbps]	64
	$\overline{Peak/Mean}$ frame size	5.48
Transport Layer	Segment size L_{TCP} [bytes]	1400
	TCP header [bytes]	20
Network Layer	L_{buffer} [segments]	10000
	IP header [bytes]	20
Data Link Layer	Packet size L_{MAC} [bytes]	91
	SMPT policy	Slow Healing
Physical Layer	Slot length $\tau_{\text{frame}} = \frac{L_{\text{MAC}}}{C}$ [ms]	10
	number of Available Channels R	1, 3
	Bit Rate C [kbps]	72.8
	Spreading Gain	16, 32
	Frame size [bits]	1023
FEC code	Payload [bits]	728
	Redundant [bits]	295
	Correctable errors	30
Wireless Link	Bad state duration τ_{bad} [ms]	100
	Good state duration τ_{good} [ms]	1000
	Bad state p_{err}	1.0
	Good state p_{err}	Improved Gaussian Approximation
	Background channels b	2,4
Simulation	Confidence Level CV	99%
	Measure interval τ_{tic} [s]	10

of CDMA channels of all WTs, which results in higher BEPs. Increasing the number of WTs further leads to a small decrease in the buffer underrun rate. This is caused by TCP mechanisms, which try to adapt to the channel behavior. Henceforth, we concentrate on the operational phase, where both approaches give acceptable results. For the given scenario with 0.5 sec (1.0 sec) offset we observe for the operational phase that the average time between buffer underruns is $T_{\Omega,seq} = 50$ sec ($T_{\Omega,seq} = 100$ sec) for the sequential and $T_{\Omega,SMPT} = 250$ sec ($T_{\Omega,SMPT} = 500$ sec) for the SMPT approach. The reason for lower buffer underrun rates is illustrated in Figure 8.13 where the buffer content versus time is depicted for the sequential and the SMPT transmission mode. The buffer size is measured at one dedicated WT. The figure reflects the stabilizing effect of SMPT. Within the investigated time interval no buffer underrun takes place for SMPT. On the other hand, the buffer content of the sequential transmission mode is highly variable and two buffer underruns occur. We note that for illustration the offset value, which corresponds to the buffer capacity at the receiver is set to 3 sec in this sample path plot. (The excursions up to 3.5 sec are due to the granularity of the TCP segments.) For all the following figures the offset value is set to either 0.5 sec or 1 sec.

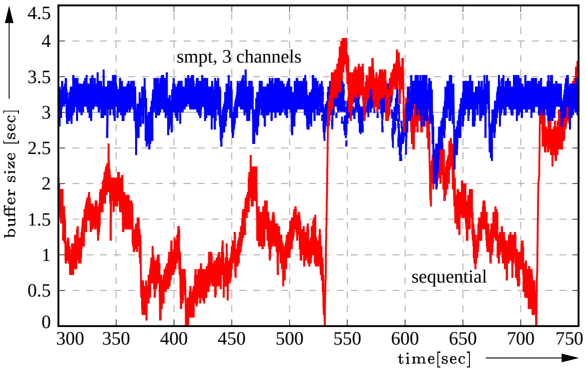


Figure 8.13: Buffer size for sequential and SMPT transmission approach over 450 ms.

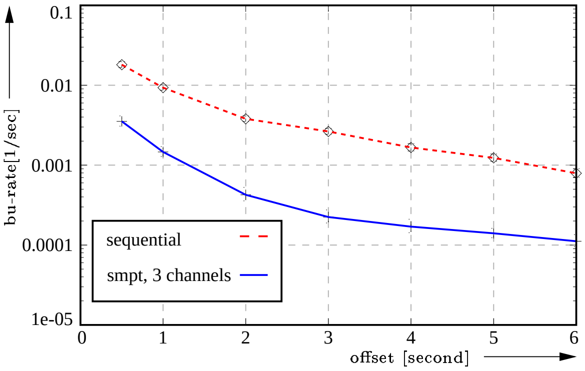


Figure 8.14: Buffer underrun rate for the sequential and the SMPT transmission approach versus offset values of 0.5 sec up to 6 sec.

To show that the SMPT gain is not only due to the higher bit rates (bundling of CDMA channels), we compare the SMPT approach with a sequential transmission mode using double bit rate. The doubled bit rate is achieved by halve the spreading gain G. Note, a comparison of SMPT and the sequential transmission mode with the same bandwidth as the SMPT approach is not easy to perform. By doubling the bit rate and halving the spreading gain allows us to keep the energy per bit ratio constant. With the higher bit rate, we can send even two packets within one time slot. For each packet we use the same coding approach as before. We observe from Figure 8.11 and Figure 8.12 that for this transmission mode with doubled bit rate, the average time between buffer underruns is $T_{\Omega,seq^2} = 500$ sec; thus the improvement in the buffer underrun rate is slightly higher than for SMPT. However, the operational phase (capacity of the cell) is much smaller than for the SMPT approach (five video streams for doubled bit rate versus eight video streams with SMPT). To decrease the buffer underrun rate further we could increase the offset value τ_{off}. The impact of the offset value τ_{off} is given in Figure 8.14. We note, that higher offset values require a a larger buffer and introduce a larger time shift between play–out time and reality. We observe that for the entire range of studied offset values τ_{off}, the buffer underrun rate of SMPT is roughly one order of magnitude smaller than that of the sequential transmission mode. Next, we investigate the capacity (i.e., the maximum number of supported WTs in the operational phase) and the impact of the spreading gain. With a spreading gain of 32 (instead of 16, as used before) the buffer–underrun rate versus the number of WTs is given for two different offset values in Figure 8.15 and 8.16. In Figure 8.15 the offset value is set to 0.5 sec and in Figure 8.16 to 1.0 sec. In this higher spreading gain scenario, SMPT achieves both a lower buffer–underrun rate and a higher capacity (21 video streams with SMPT versus 19 with sequential mode, noting that the sequential mode has a buffer underrun rate that is one order of magnitude larger). The reason for this behavior is a larger multiplexing effect on the wireless link. With a spreading gain of $G = 16$ the BEP changes from 10^{-3} to 10^{-1} in the range of three active channels, while for $G = 32$ the range increases to five WTs. With this larger range, a higher multiplexing gain can be achieved (see Figure 8.10).

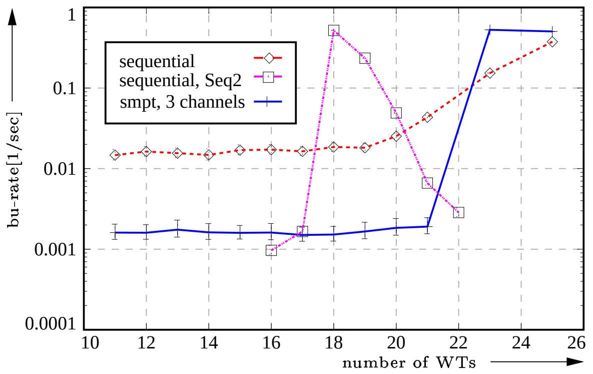


Figure 8.15: Offset 0.5 sec.

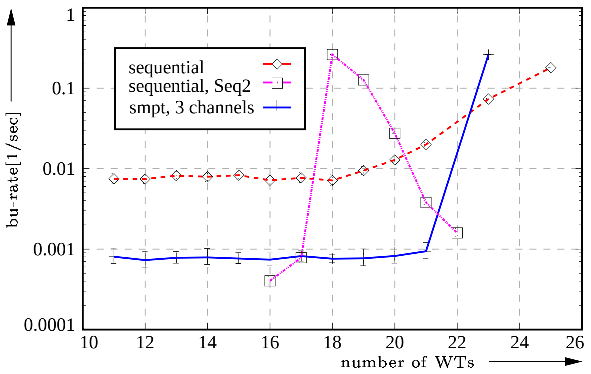


Figure 8.16: Offset 1.0 sec.

Buffer underrun rate versus number of WTs for the SMPT approach (R_{max}=3) and the sequential transmission approaches with single and double bit rate ($G_{Spreading} = 32$).

An effect that was not taken into consideration in our simulations so far is the inter–cell and intra–cell interference. The inter–cell interference is a measure for the energy that the cell *exports* to the neighboring cells. Therefore, the mean number of used channels within a CDMA cell versus the number of WTs for the SMPT approach ($R_{max} = 3$) and the sequential transmission approaches with single and double bit rate for an offset of 0.5 sec are given in Figure 8.17. We observe that the sequential transmission mode with single bit rate and SMPT produce the same amount of interference within the operational phase. Only the sequential transmission mode with double bit rate gives smaller inter–cell interference within its operational phase. This may be explained by the fact that transmitting with a higher data rate is helpful if all data can be send before the next bad channel state starts.

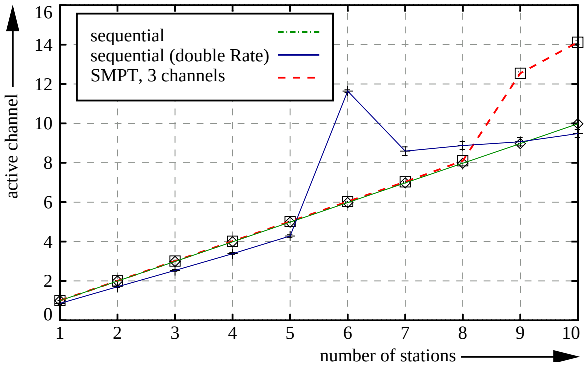


Figure 8.17: Mean number of used channels within a CDMA cell versus number of WTs for the SMPT approach (R_{max}=3) and the sequential transmission approaches with single and double bit rate for an offset of 0.5 sec.

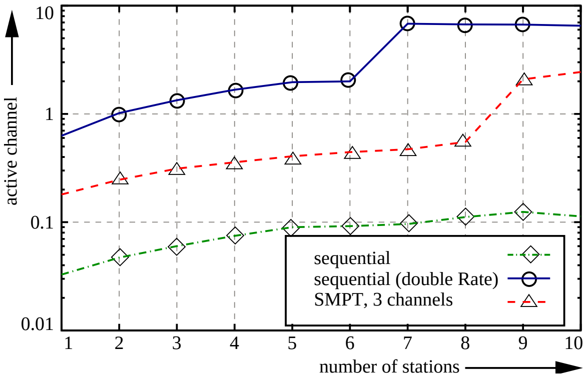


Figure 8.18: Standard deviation of used channels within a CDMA cell versus number of WTs for the SMPT approach (R_{max}=3) and the sequential transmission approaches with single and double bit rate for an offset of 0.5 sec.

The intra–cell interference, on the other hand, reflects the variability of the number of used channels. The higher the variability the better has to be the power control entity. Therefore, we depict the standard deviation of the number of used channels in a CDMA cell versus the number of WTs for the SMPT approach ($R_{max} = 3$) and the sequential transmission approaches with single and double bit rate for an offset of 0.5 sec in Figure 8.18. We observe that the values of the standard deviation for SMPT are one order of magnitude larger than for the sequential transmission mode with single bit rate and one magnitude smaller than for the sequential transmission approach with double bit rate.

8.1.2.2.2 MPEG–4 without Rate Control In this section we want to discuss the performance of streamed MPEG–4 video over wireless uplinks. In contrast to the H.263 video used in Section 8.1.2.2.1 the MPEG–4 video will be encoded without any target rate. Like it is shown in Section 5.2.2 it is not unusual that the peak bit rate can be a multiple of the mean bit rate. For the low quality level we observe e.g. a peak to mean bit rate ratio about 20 for the *Silence of Lambs* movie. We use the MPEG–4 video traces with low quality level introduced in Section 5.2.2. To have comparable results to H.263 we scaled each video trace file to a mean bit rate of 64 kbps (according to Section 5.4 scaling of these video sequences is possible). Thus the MPEG–4 video trace has the same mean bit rate but is still highly variable. For the MPEG–4 performance evaluation we use the same scenario and metric as it was already introduced in Section 8.1.2.2.1.

In Figure 8.19 the buffer underrun rate [$1/sec$] versus the number of WTs within one CDMA cell is given for the sequential and the *Slow Healing* SMPT transmission mode. Note that for the SMPT approach we assumed $R_{max} = 3$. The offset was set to three seconds and the available bit–rate on the wireless link is 72.8 kbps. We observe the

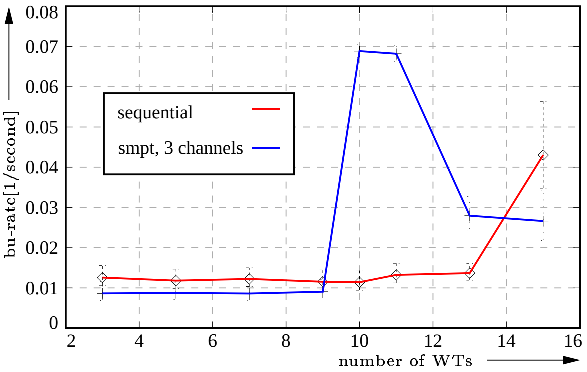


Figure 8.19: Buffer underrun rate versus the number of WTs using MPEG–4 video streams (offset = 3 seconds, 72.8kbps uplink).

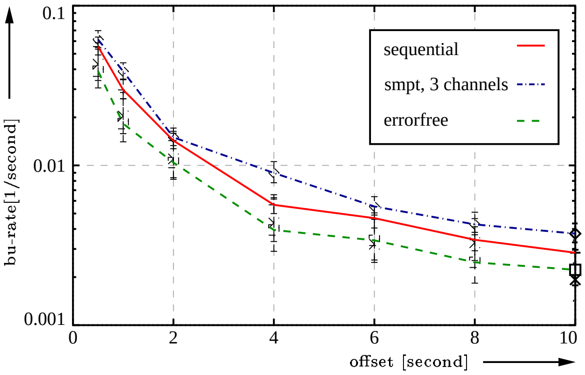


Figure 8.20: Buffer underrun rate versus the offset for nine WTs (72.8kbps uplink).

same behavior for both the sequential and the SMPT approach as that in case of H.263 performance evaluation. The SMPT approach has a smaller buffer underrun rate than the sequential transmission mode for less than ten WTs in the cell. For a higher number of user the systems become unstable for SMPT. But the gain of SMPT is much smaller than we observed for the rate controlled H.263 traffic. For an offset of three seconds the SMPT approaches will have a buffer underrun every 113 sec. The sequential transmission mode has an empty buffer every 83 sec. Both values seem to be unacceptable for streaming applications. If we compare these result with H.263 where we used the same offset value, the buffer underrun occurs each 601 sec for the sequential and each 4461 sec for the SMPT transmission mode.

The buffer under rate can be enlarged by simply introducing higher values for the offset. In Figure 8.20 the buffer underrun rate versus the offset for nine WTs is depicted. We observe that even with an offset value of ten seconds the time between two buffer underruns is still large. For such an offset value the buffer underrun rate is $4 \cdot 10^{-2}$ sec^{-1} and $3 \cdot 10^{-2}$ sec^{-1} for the sequential and the SMPT approach, respectively. With an offset of only six seconds the buffer underrun rate for H.263 was below 10^{-3} sec^{-1} for both approaches. Even for different values for the offset the gain of SMPT is small. The reason for this is that H.263 is rate controlled, while MPEG–4 is not. The variability of the video stream lead to situation were the bit rate of the video stream is very high and the buffer on receiver–side will not get filled fast enough. Even for an error–free channel the buffer underrun rate is high as shown in Figure 8.20.

To overcome the described problem of high buffer underrun rates we have to reduce the bit rate in case the data rate is to high for the current channel conditions. Two possibilities are taken into considerations in the following work (see [48] for details):

Simulcast A stream is encoded in different quality levels simultaneously. Depending on the wireless channel the encoding rate can be switched.

Temporal Scalability An MPEG–4 GOP consists of three types of frames (see Section 5.1.1), namely I,P, and B frames. In our source model we use one I frame, three P frames and eight B frames within one GOP. The I frames are the most important ones and therefore have to be transmitted always successfully. But frames of the other type can be skipped if necessary.

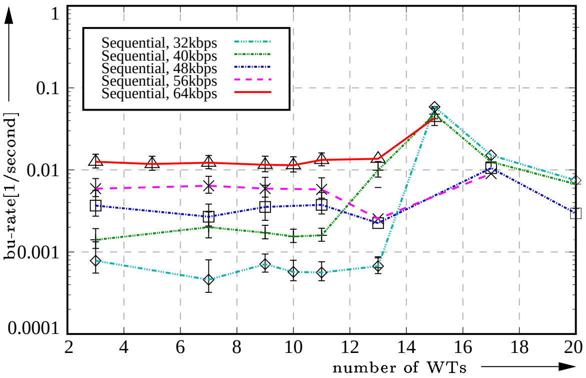


Figure 8.21: Buffer underrun rate for the sequential transmission approach versus the number of WTs using different target bit rates.

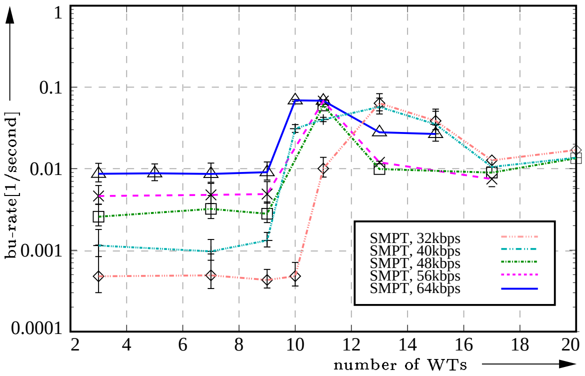


Figure 8.22: Buffer underrun rate for the SMPT transmission approach versus the number of WTs using different target bit rates.

First we consider the simulcast approach. Therefore we use the same video sources with different mean bit rates of 32,40,48,56, and 64 kbps. Note, the video streams are still variable. In Figure 8.21 the buffer underrun rate for the sequential transmission approach versus the number of WTs is given. In Figure 8.22 the buffer underrun rate for the SMPT approach is depicted. We see that a degradation of the mean bit rate lead to considerable buffer underrun rates for both approaches. The gain of SMPT in contrast to the sequential transmission mode becomes even smaller with smaller mean bit rates. But a diminution of the mean bit rate results also in a worse video quality. The operational phase is much larger for the sequential than for the SMPT approach. The sequential case has J=13 WTs within its operation phase for all bit rates. For the SMPT approach we note an increase for the operational phase using 32 kbps. For this bit rate the number of supportable WTs increase from J=9 up to J=10. As mentioned before we want to switch the encoding rate in dependency of the channel state. Therefore we have to consider a certain amount of signaling. Furthermore we face one new problem using TCP as the transport protocol. Even if the knowledge of the wireless channel is perfect, the changes in the encoding rate will affect the bit rate only with a time delay. This time delay is created by the TCP buffer. As an example consider the situation were the wireless channel is considered to be *good* and we use an certain encoding rate. If the conditions on the link change dramatically, e.g. only the half of the bit rate can be achieved, we change at once the encoding rate. But still for a while higher encoding rates are transmitted over the wireless link, because some frames, encoded at a high bit rate, were already buffered by the TCP layer and can not be changed or removed afterwards.

Using the temporal scalability approach frames are discarded in dependency of the sender–side or receiver–side statistic. Henceforth, these approaches are called *Sender Statistic Based Approach (SSBA)* and *Receiver Statistic Based Approach (RSBA)*, receptively. The remaining question is at which protocol layer we need to discard the frames in such situations. Because of TCP's reliable transmission the frames has to be discarded above the transport layer. Therefore we introduce a sub–layer to skip video frames between the video application and the transport layer (see Figure 8.23).

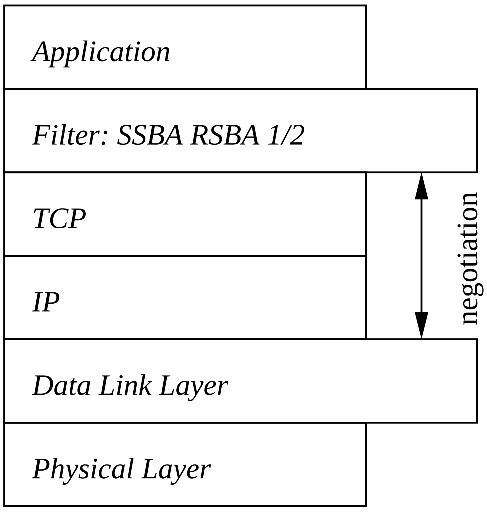


Figure 8.23: Protocol stack with sub–layer for discarding video frames.

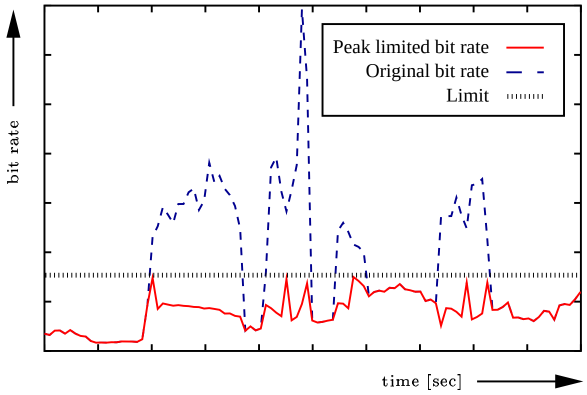


Figure 8.24: Comparison of the original video stream with a stream produced by the *Peak–Limiter.*

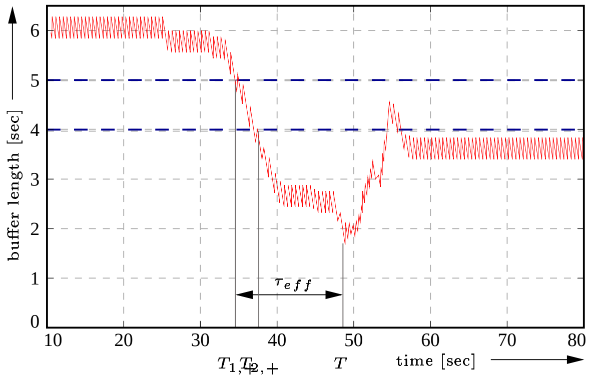


Figure 8.25: Receiver side buffer for the RSBA–2.

The SSBA limit the peak rate of the video sequences by skipping B or B and P frames within one GOP. Therefore we call this approach also *Peak–Limiter.* Each GOP is passed to TCP via the filter entity. If the mean bit rate for each GOP exceeds a given bit rate B_{peak} all B frames are skipped within this GOP. If the mean bit rate still exceeds B_{peak}

after skipping the B frames the P frames are also skipped. This approach acts at the sender–side and therefore no signaling is needed. Note, the amount of lost frames will be the same for the sequential and the SMPT approach. In Figure 8.24 the bit rate of the original video stream and the bit rate of the peak–limited video stream is given. The peak–limited stream is sometimes smaller than the target bit rate B_{peak}. This is because we throw away either all B frames or all B and P frames. In Figure 8.26 the buffer underrun rate versus the number of WTs within one CDMA cell is given for the sequential and the SMPT approach. We see that the buffer underrun rate improves at the expense of video quality. For both approaches the frame loss rate of B frames is about 5.1% and the loss rate of P frames is 1.8%. For SSBA the sequential transmission mode leads to better results.

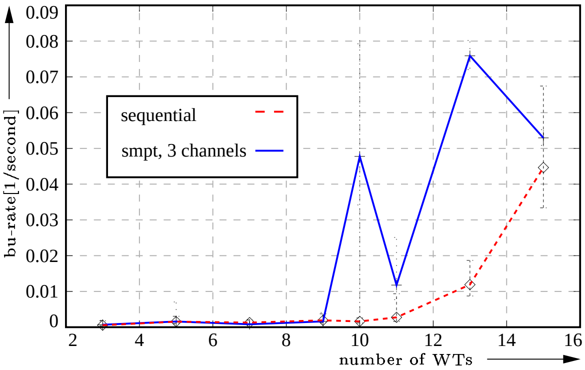


Figure 8.26: Buffer underrun rate versus the number of WTs for the sequential transmission approach and the SMPT approach using the SSBA policy.

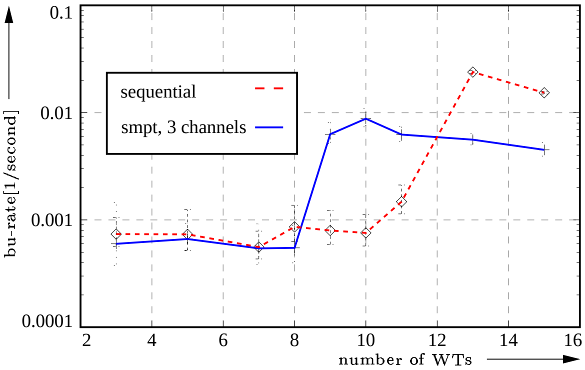


Figure 8.27: Buffer underrun rate versus the number of WTs for the sequential transmission approach and the SMPT using the RSBA–1 policy.

For all RSBA we assume that on sender–side the full knowledge about the play–out buffer at the receiver side is known. This could be achieved by a peer–to–peer communication between the video applications. On the other side this requires at lot of signaling, which has to be transported over the wireless link. This is even more critical if signaling has to be done within bad channel conditions. Therefore we suggest to calculate the play–out buffer within the sender–side data link control entity. Of course the calculated state of the the play–out buffer is not perfect. But on the other side no signaling is necessary over the wireless link.

In addition to the SSBA we introduce a sender side buffer. This sender side buffer allows us to fill–up the receiver side buffer if its gets empty. The RSBA skips frames of the sender side buffer in dependency of the receiver–side play–out buffer. If the play–out buffer occupancy goes below a certain threshold $T_{1,+}$ of play–out buffer time the sender side filter entity will not pass B frames to the TCP layer. These frames are lost. If the play–out buffer occupancy goes further below a threshold $T_{2,+}$ even the P frames are skipped together with the B frames. If the play–out buffer increases and a threshold $T_{2,-}$ is reached, P frames will be transmitted again. Once the buffer play–out time increases over a threshold $T_{1,-}$ the filter stops skipping B frames. To heal the receiver–side buffer we introduce a sender–side buffer. Before the play–out process at the receiver starts the sender and receiver–side buffer have to be filled. Therefore it is possible to fill up the receiver–side buffer at the expense of the sender side buffer.

In the following we distinguish two RSBA approaches: RSBA–1 has an unlimited sender side buffer. An unlimited sender side buffer can only be achieved with prefetched media. RSBA–2 has an limited buffer with six seconds of buffering time. For both approaches we assume a receiver side buffer of six seconds and set $T_{1,+}$=5.0 sec, $T_{2,+}$=4.0 sec, $T_{2,-}$=4.5 sec, and $T_{1,-}$=5.5 sec.

In Figure 8.27 buffer underrun rate versus the number of WTs for the sequential transmission approach and the SMPT using the RSBA–1 policy is given. We note a significant performance improvement for both approaches comparing with the results presented in Figure 8.20 can be achieved for the buffer underrun rate. On the other side the video quality suffers by skipping P and B frames. The loss rate for both, the sequential and the SMPT approach are given in Figure 8.28 and 8.29.

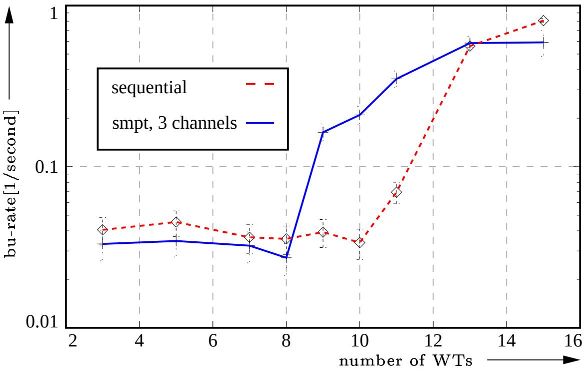


Figure 8.28: Frame loss rate of B–frames for the RSBA–1 approach.

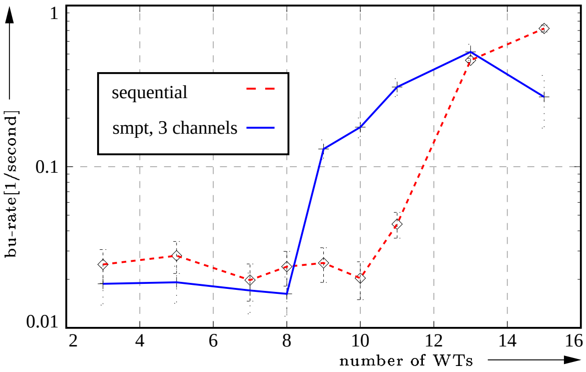


Figure 8.29: Frame loss rate of P–frames for the RSBA–1 approach.

In Figure 8.30 buffer underrun rate versus the number of WTs for the sequential transmission approach and the SMPT using the RSBA–2 policy is given. The results are a little bit worse than for the RSBA–1 case. The reason for this behavior is the limited sender side buffer. Once the sender side buffer is consumed, the systems reacts like the basic MPEG–4 transmission. The frame loss rates of B and P frames for the RSBA–2 approach, presented in Figure 8.31 and 8.32, increase steadily because of the increasing interference level.

8.1.3 Conclusion

We have studied the streaming of video using TCP as the transport protocol in cellular CDMA–based wireless systems. TCP has a number of desirable properties, such as network stability and fair bandwidth allocation, for the future Internet. Video streaming over TCP, however, is generally known to give poor performance, especially in wireless environments. We have proposed a scheme that uses TCP to stream video in a MC–CDMA system. Our scheme employs **S**imultaneous **MAC** **P**acket **T**ransmission (*SMPT*) to effectively stabilize the wireless links. This stabilizing effect significantly improves the

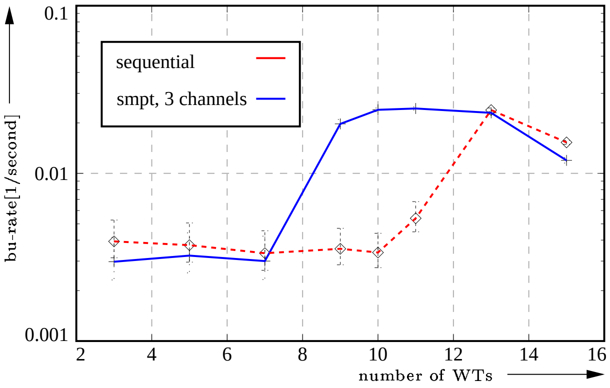


Figure 8.30: Buffer underrun rate versus the number of WTs for the sequential transmission approach and the SMPT using the RSBA–2 policy.

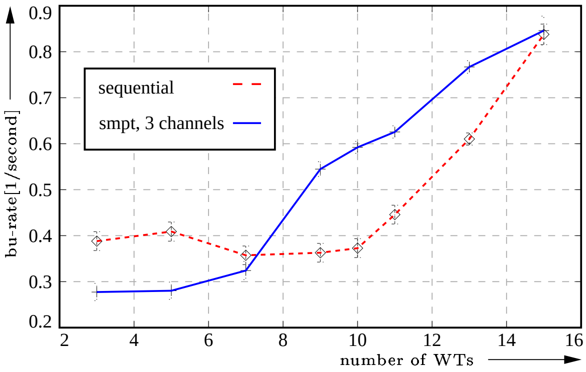


Figure 8.31: Frame loss rate of B–frames for the RSBA–2 approach.

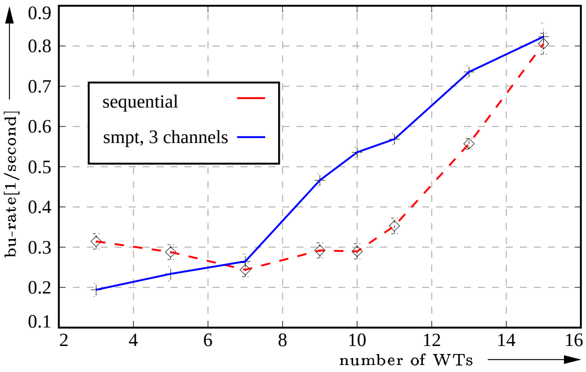


Figure 8.32: Frame loss rate of P–frames for the RSBA–2 approach.

performance of TCP over the wireless links. We found that our approach gives good results for rate controlled encoded video. For a given buffer underrun requirement (which corresponds to a pre–specified client playback starvation probability), our SMPT approach increases the number of supported streams in a cell in a typical CDMA scenarios. We also observed that SMPT performs better with higher spreading gains. We note that our streaming scheme preserves the isolation of the protocol layers. Specifically, SMPT does not require knowledge of any TCP parameters. Independently of the TCP operation, SMPT stabilizes the throughput over the wireless link and thus significantly reduces the probability that the TCP round trip time out is exceeded. On a final note, all numerical experiments in this section were conducted for rate controlled encoded video, which is typically less variable than video encoded without any rate control (i.e., open–loop encoded video). The peak–to–mean frame size ratio is typically less than 4 – 5 for rate controlled video, while the peak–to–mean ratio of the frame sizes may be up to 14 or even higher for open–loop encoded video. Our numerical experiments reported in this section indicate that the streaming of rate controlled encoded video over TCP gives good performance when the stabilizing SMPT scheme is employed. We have also conducted extensive numerical experiments for the streaming of open–loop encoded video. We have found that the streaming of open–loop encoded video over TCP in wireless systems gives generally poor performance, both without and even with SMPT. We attribute this result to the limited ability of TCP to accommodate the highly variable open-loop encoded video streams.

8.2 Support of UDP Traffic over SMPT

In this section we provide a performance evaluation of UDP over SMPT. Two different traffic sources are used to investigate the effect of the SMPT approaches. CBR sources are used because of their simplicity. Thus all effects measured are based on the SMPT algorithm and are not an effect of the traffic pattern. Video source, similar to the TCP performance evaluation, are used to investigate the effect of some SMPT approaches for real scenario.

8.2.1 CBR Source

To investigate the effect of the different SMPT approaches CBR sources are taken as traffic generators. With this simple traffic source the impact on neighboring cells, the impact on the power control entity and the effect of the channel acquisition time is given for the *All–Responsive* environment.

8.2.1.1 Comparison of Enhanced SMPT Techniques

This work is based on [194, 23] and parameters chosen are based on the **I**ntegrated **B**roadband **M**obile **S**ystem (*IBMS*) [195, 196] project. In this section we present the simulation results for the enhanced SMPT techniques and the sequential transmission for comparison. We concentrate on the QoS parameters goodput, i.e. bit rate at the transport layer consisting of successfully transmitted segments, segment loss probability, and delay–jitter. The transport level delay–jitter and delay bounds are set to $\tau_{\text{jitter}} = 240$ ms and $\tau_{\text{delay}} = 600$ ms. The segment length is uniformly distributed from one to 4000 bytes. Each WT is allowed to use up to R_{max}=8 channels in parallel. The basic rate of one CDMA channel is fixed at 64 kbit/s with **L**ink–layer **P**acket **D**ata **U**nit (*LPDU*) unit of length L_{LPDU}=192 bytes. The application of each WT generates a load with constant bit rate R_{C}; thus each WT uses exactly one channel when the communication is error–free. The queue in the MAC layer can store L_{Q}=4800 bytes. The simulations were performed with a confidence level of 95% for the goodput value. The rest of the simulation parameters are given in Table 8.3.

In Table 8.4, the goodput and segment loss probability is given for a specific cell penetration of 10 WTs. The *Slow Start*, *Slow Healing* and the *Fast Healing* mechanisms yield high goodput values of roughly 60 kbit/s, while *Fast Start* gives only 28 kbit/s, which is even lower than the goodput of the sequential transmission mechanism. The results for the segment loss probability are similar. We note that the different transmission strategies were optimized to achieve a high goodput. All delay–jitter values are bounded by τ_{jitter}. The *Slow Start* approach leads to the smallest values, both in terms of mean (10 ms) and variation (32 ms). The precise value for the mean number of used codes to support 10 WTs is 19.74 for the *Fast Start* and 10.59 for the *Slow Healing* mechanism. As noted above, a high number of used channels leads to performance degradation.

In Figure 8.45 the goodput versus the number of WTs for different transmission strategies is depicted. For a small number of WTs the goodput for all SMPT strategies is much higher than the sequential case with 47 kbit/s. The *Start Up* mechanisms yield

Table 8.3: Simulation parameters for the UDP scenario

Employment	Parameter	Value
Application	τ_{max}	600 ms
	τ_{δ}	240 ms
	χ	95%
Transport Layer	L_{segment}	1-4000 byte
Data Link Layer	L_{Queue}	4800 byte
	L_{DPDU}	192 byte
	R_{probing}	1 LPDU/slot
Physical Layer	τ_{slot}	24 ms
	R_{max}	1-8
	Spreading Gain	128
	Bit Rate	64 kbit/s
Wireless Link	τ_{bad}	60 ms
	τ_{good}	540 ms
	p_{good}	Gaussian Approximation PACS
	Background Noise	3 WTs
Simulation	Confidence Level	95%
	Relative Error	5%

63 kbit/s, while the *Self Healing* mechanisms result in 60 kbit/s. With an increasing number of wireless devices some strategies lead to results even worse than the sequential case. *Slow Start* and *Slow Healing* always have better performance in terms of goodput values. As mentioned above, using channels in parallel can lead to intra/inter–cell performance degradation. Therefore, *Fast Healing* with a maximum number of 5 and 8 channels is depicted. It can be demonstrated for this approach, that a lower maximum number of parallel channels leads to a performance gain. The reason for this is the lower mean number of used channels in the cell. Allowing up to 5 channels per WT with 10 WTs in the cell results in an overall number of 13.47, instead of 19.74 for a maximum number of 8 (see also Table 8.4). Even with a small number of parallel channels such as 2 or 3 channels, we observe very good results for all SMPT approaches. A higher number of codes leads only to performance improvements for a small load in the cell, which can be observed by an increased goodput from 60 to 63 kbit/s.

Table 8.4: Simulation results for 10 wireless terminals using the sequential transmission and the SMPT approach with eight parallel channels.

Parameter		Sequential	Simultaneous MAC Packet Transmission			
			Start		Healing	
			Fast	Slow	Fast	Slow
Goodput	[bit/s]	43849.36	28208.3	**61827.5**	**60213.3**	**59930.8**
Throughput	[bit/s]	52147.3	38157.0	62229.7	61639.7	60933.7
Jitter	E{}	3.76	2.26	**0.434**	0.917	1.13
	Var{}	6.163	7.308	**1.331**	**1.291**	**1.905**
Segment Loss Rate	[%]	24.03	44.25	**2.28**	**3.75**	**4.30**
Overall Codes	E{}	**8.98**	19.74	11.16	**10.81**	**10.59**
	Var{}	**1.154**	156.474	12.510	3.524	**1.691**
Idle Period	Length	8.49	6.75	5.75	1.98	2.63
	prob. [%]	11.11	32.38	38.35	3.30	3.82

8.2.1.2 Impact on Neighboring Cells – Inter Cell Interference

A high number of used channels will lead to a performance degradation. It has always to be assured that the system is protected against system instability. Figure 8.33 shows the mean number of used codes in one cell for all transmission approaches for an increasing number of WTs. Both values, the mean and the variance in code usage, have a high impact on the intra-cell and inter-cell interference [18], and therefore have to be kept small. If the overall number of codes in one cell will be used by leaps and bounds (high variance) the power control unit with the CDMA based mobile terminal can not re-calibrate itself. Therefore a moderate usage of codes has to be guaranteed. E.g. in IS95 only 1db per 1.25ms can be adjusted under usage of the closed loop approach.

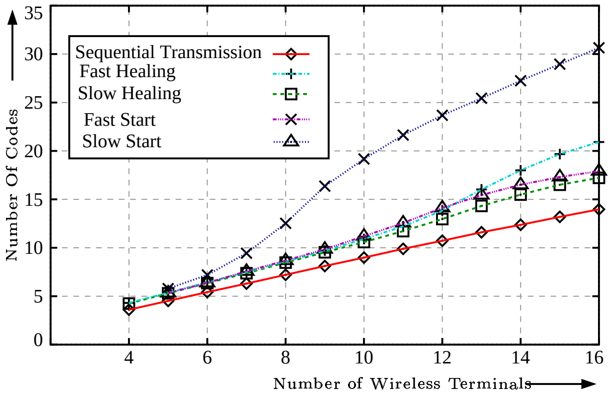


Figure 8.33: The overall number of codes used in one wireless cell for different transmission mechanisms.

8.2.1.3 Impact on Power Control Entity – Intra Cell Interference

The resulting impact on power control and power saving is discussed in the following. Therefore, in Figures 8.34 and 8.35 the conditional probability $p(\tau{+}1, \tau)$ of sending with an overall number of codes during slot τ+1 after slot τ for two different SMPT approaches is depicted. The conditional probability $p(\tau{+}1, \tau)$ reflects the dynamics of the number of used codes in the wireless cell. Fast time–scale-variations of the code usage require a high–performance power control that is able to deal with such large energy fluctuations. Figure 8.35 depicts the conditional probability $p(\tau{+}1, \tau)$ for the *Fast Start* mechanism. Figure 8.34 gives the conditional probability for the *Slow Healing* mechanism. We observe that the *Fast Start* mechanism has a significantly larger variance of the code usage than the *Slow Healing* mechanism. This conclusion is also supported by the values summarized in Table 8.4. The variance of the total number of codes used by the *Fast Start* is 156.47, while the *Slow Healing* mechanism only has a variance of 1.69. It always has to be assured that the system is protected against instability. Both the mean and the variance in code usage have a large impact on the intra–cell and inter–cell interference [18], and should therefore be kept small. If the overall number of codes in one cell has a large variance, the power control unit of the CDMA based mobile terminal can not re–calibrate itself. Therefore a moderate usage of codes has to be guaranteed.

At last, the energy aspect has to be interpreted. We see from Table 8.4, that the sequential transmissions, *Fast Start* and the *Slow Start* , achieve a large probability of having idle periods, during which no sending takes place. The idle periods occur, if no transport segment has to be transmitted at the moment and the sender has either already transmitted the last **L**ink–layer **P**acket **D**ata **U**nit ($LPDU$) of a transport segment or dismissed transmission because of the delay bounds. Moreover, the average idle periods are long. Note, however, that sequential transmission and *Fast Start* mechanisms achieve long idle times, because they fail to meet the pre–specified QoS parameters (see Section 6.1) and discard LPDUs resulting in a high segment loss probability. The *Slow Start* mechanism can achieve both: energy saving as well as a small segment loss probability.

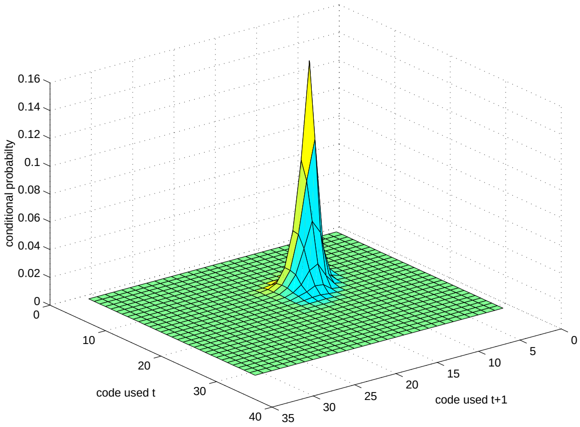


Figure 8.34: Conditional probability sending with overall number of codes for the *Slow Healing* approach.

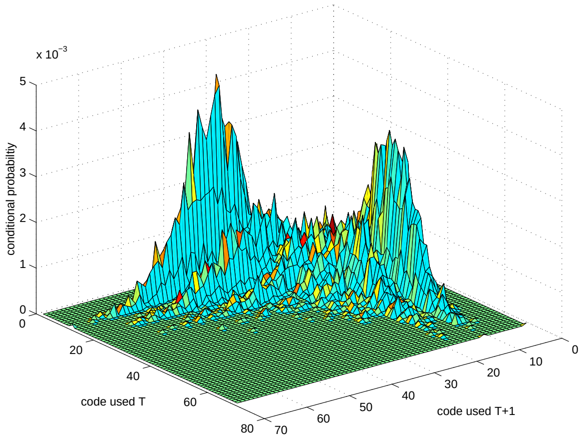


Figure 8.35: Conditional probability sending with overall number of codes for the *Fast Start* approach.

8.2.1.4 Impact of the Channel Acquisition Delay

As mentioned in Section 4.4.2 we want to investigate the impact of the channel acquisition delay. Therefore, in Figure 8.36 and 8.37 the impact of the channel acquisition delay for 10 and 12 WTs in one CDMA cell is given for the *Slow Healing* approach. For the simulation we assume a CBR source with variable packet length between 384byte and 3840byte. Furthermore segments that have already a jitter larger than 80 ms or a delay larger than 560ms are skipped by the sender entity. In both figures the **S**egment **L**oss **P**robability (*SLP*) [%] versus the channel acquisition delay [DLC frame] is depicted. In Figure 8.36 we see that both, the sequential and the SMPT approaches were investigated for ten WTs in the cell. While the sequential transmission mode is obviously not effected by the channel acquisition delay, the SLP increases of the SMPT transmission mode for higher acquisition delays. But even for acquisition delays of four DLC frames (Assuming UMTS frame slots of 10 ms this would lead to an acquisition delay of 40 ms.) the SLP is still below the sequential transmission mode. The sequential transmission approach leads to SLP of 4%. The SMPT approach has SLP of 0.8% for no delay and increases up to 2.0% for delay values of four frames slots[6].

For twelve WTs we notify that 1.) the segment loss probabilities increase for both, the sequential and the SMPT approach and 2.) higher channel acquisition delays helps SMPT to decrease the SLP. The SLP for the sequential transmission is 5.5%. The SMPT mechanism has SLP of 7% for no delay. For a delay of 1 DLC frame the loss probability decrease down to 1.5%. With higher delay values it increases again up to 3.0% for delay values of four frames slots.

The reason for this phenomena can be explained if we take a closer look on the overall used CDMA channels in the cell. For SMPT with zero acquisition delay the mean number

[6] Note, a channel acquisition delay value of 0.5 means, that the first channel can switched on with zero delay and the next channel with a delay of one.

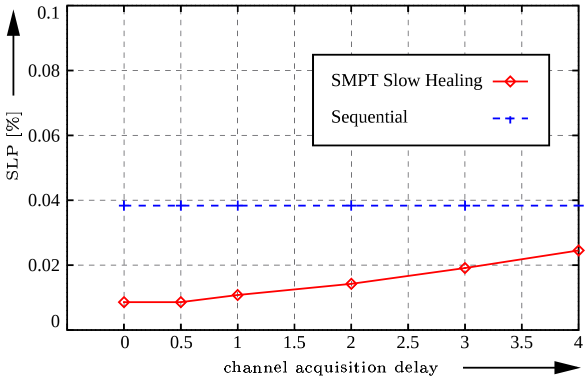


Figure 8.36: Impact of the channel acquisition delay on the SLP for 10 WTs.

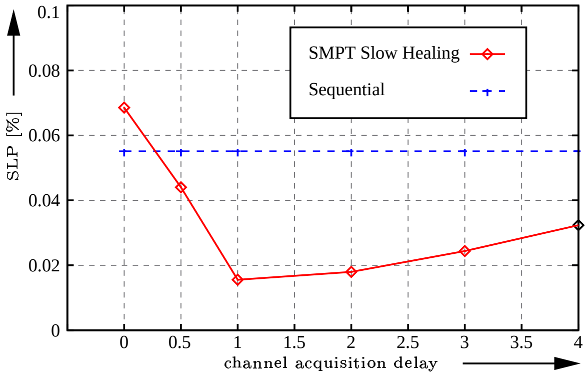


Figure 8.37: Impact of the channel acquisition delay on the SLP for 12 WTs.

of used codes is 12.52 (variance 3.93). For a delay value of 20 ms the number of used channels decreases to 12.14 (variance 0.47). This shows that a more moderate way of using additional channels could lead to an improvement if too many WTs are in the cell. Until now, we took the channel acquisition delay as a physical restriction for the SMPT mechanism. As seen by the results of Figure 8.37 a ramp with a smaller gradient in building the ramp is helpful in some situations such as overload. It has to be mentioned that an increased channel acquisition delay always leads to an increased jitter of the segments. This is given in Figure 8.38 for the sequential nd the SMPT approach for 10 and 12 WTs. For higher channel acquisition delays the jitter values of SMPT can be even larger than the jitter values of the sequential transmission mode.

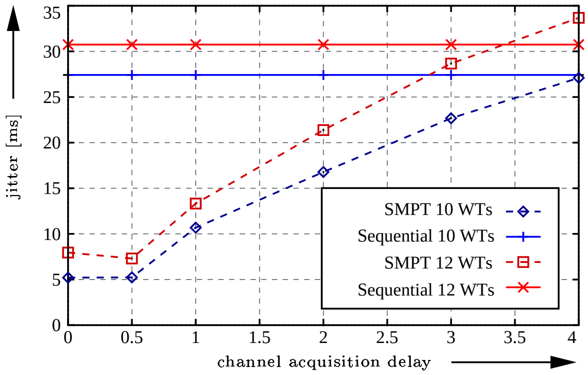


Figure 8.38: Impact of the channel acquisition delay on the jitter.

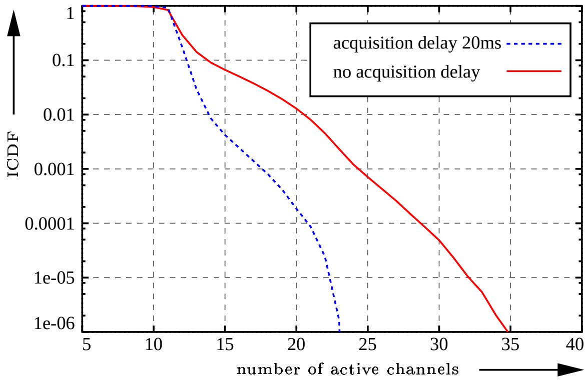


Figure 8.39: Inverse cumulative density function (ICDF) for the overall number of channels active in the cell.

8.2.1.5 Conclusion

Different SMPT approaches were introduced and evaluated. Mechanisms that use the wireless resources in a moderate way, such as *Slow Start* and *Slow Healing* achieve the best results. For a typical scenario of 10 WTs, both mechanisms achieved a segment loss probability of less than 5%; the segment loss probability of the sequential transmission is almost five times larger. While *Slow Start* achieves better power saving, the *Slow Healing* mechanism has a smaller variance of the number of used codes; the variance is almost as small as for the sequential transmission, which is very important for minimizing the impact on the power control entity. Moreover, we have demonstrated that using multiple channels can result in performance degradation, if the mechanism (e.g. *Fast Start*) uses the wireless resources excessively. We conclude that it is insufficient to merely equip the WTs with data rate adaptive procedures. Instead, mechanisms such as *Slow Healing* SMPT have to be implemented within the data link layer to (1) protect the system stability, and (2) enable heterogenous QoS support for multimedia applications. It can be observed that the improvement of the throughput and delay within SMPT is achieved with a relatively low number of additional codes. Furthermore we have shown the impact of the channel acquisition delay.

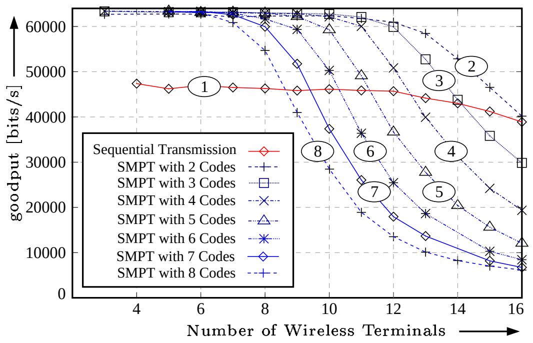


Figure 8.40: Goodput for the *Fast Start* approach for 1,2,3,4,5,6,7,8 codes

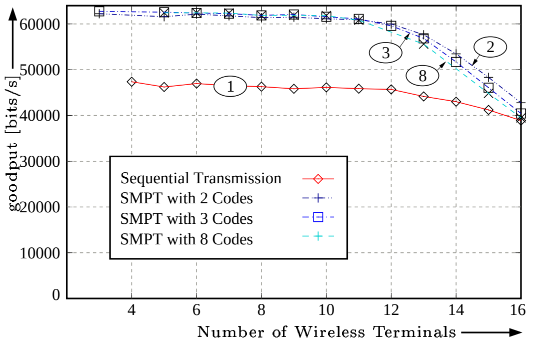


Figure 8.41: Goodput for the *Slow Start* approach for 1,2,3,8 codes

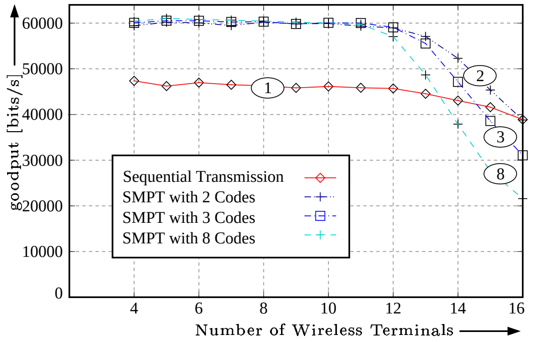


Figure 8.42: Goodput for the *Fast Healing* approach for 1,2,3,8 codes

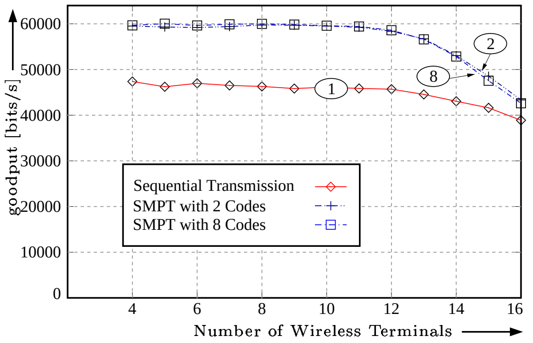


Figure 8.43: Goodput for the *Slow Healing* approach for 1,2,8 codes

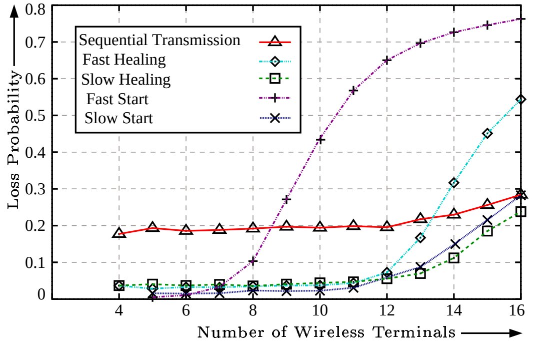


Figure 8.44: Loss probability (long–run fraction of segments lost) versus number of wireless terminals for different link layer transmission strategies with maximum number of eight channels per WT

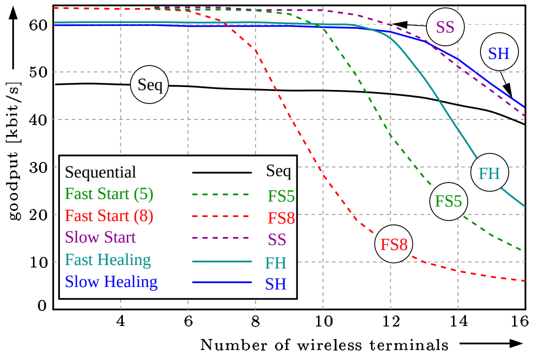


Figure 8.45: Goodput versus number of wireless terminals for different link layer transmission strategies with maximum number of eight channels per WT (additional even five for *Fast Start*).

8.2.2 Video Source

In the following we investigate the transmission of *Real–Time* Video over the wireless link. In the first scenario (presented in Section 8.2.2.1) each WT transmits rate controlled H.263 encoded video with a target bit rate of 64 kbps and in the second scenario (presented in Section 8.2.2.2) video encoded using MPEG–4 without rate control towards the BS. Note that the bit rate of the encoding without rate control is highly variable (see Section 5.2.2). Results for the rate controlled H.263 video are present in Section 8.2.2.1, while the results for MPEG–4 video are given in Section 8.2.2.2. For the simulation we use the trace driven traffic generator introduced in Section 6.2.2. The video sequence lengths are uniformly distributed over of the maximum video sequence length given by the trace files. The video offset phase θ_{max} was drawn from a uniform distribution over the interval [0, 50] seconds. As transport protocol the **U**ser **D**atagram **P**rotocol (*UDP*) is used. Each video frame is encapsulated in one UDP segment; and each UDP segment is passed to the DLC entity through the IP layer. At the DLC entity we assume that either a sequential transmission or one of the SMPT approaches is used. Thus, at the DLC entity a video frame together with an UDP and IP header is segmented into several **L**ink–layer **P**acket **D**ata **U**nit (*LPDU*) packets. A BCH(1023,640,41) **F**orward **E**rror **C**orrection (*FEC*) is used, which allows a throughput for one CDMA channel of 64 kbps. All parameters for the simulation are given in Table 8.5.

8.2.2.1 H.263 with Rate Controlled

In this section we investigate the transmission of video encoded with rate control. Compared to video encoded without rate control (see Section 8.1.2.2.2), video encoded with rate control is much easier to support than video encoded without rate control.

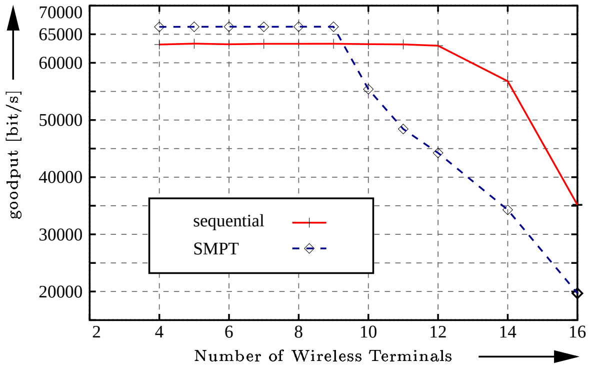


Figure 8.46: Goodput versus number of wireless terminals.

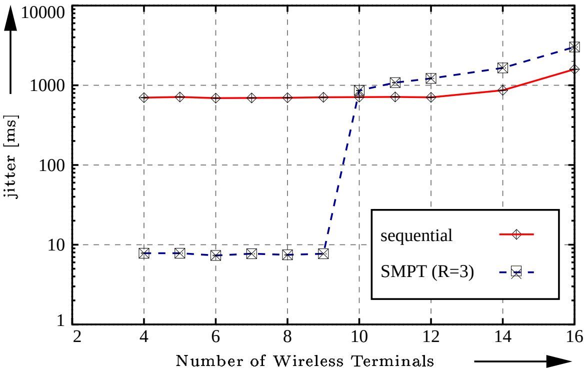


Figure 8.47: Jitter versus number of wireless terminals.

8.2.2.1.1 Impact of number of wireless terminals In Figure 8.46 the goodput versus the number of wireless terminals is given for the sequential transmission mode and SMPT (R = 3) with *Slow Healing*. For the same scenario the jitter value is presented in Figure 8.47. As seen in Section 8.1 for TCP based measurements, SMPT offers both high

Table 8.5: Simulation parameters for UDP–based real–time services.

Employment	Parameter	Value
Application	Application Type	H.263
	source rate	64kbps
	encoding type	rate controlled
	video sequence length	video sequence depending 15,20,30, or 60min uniform distributed
	Video Offset Phase θ_{max}	Uniform [0, 50] sec
	τ_{max}	variable
	τ_{δ}	∞
	χ	100%
Transport Layer	$L_{\text{segment,Transport}}$	no limitation
Network Layer	$L_{\text{segment,Network}}$	no limitation
Data Link Layer	L_{Queue}	12800 byte
	L_{DPDU}	128 byte
	ECC	47 byte
	BCH	1023,640,41
	Header	1 byte
	SMPT policy	Slow Healing
	R_{probing}	1 LPDU/slot
Physical Layer	τ_{slot}	10 ms
	k_{max}	2
	Spreading Gain	16,32
	Type of Spreading Sequences	PN, WBE
	Bit Rate	64 kbit/s
Wireless Link	τ_{bad}	30 ms
	τ_{good}	2970 ms
	Background channels b	2,4
Simulation	Confidence Level	99%
	Relative Error	1%

goodput values and low jitter values, for a certain range of WTs. In this case up to nine WTs can be supported with a goodput of 66 kbit/s and a jitter smaller than 10 ms. The sequential transmission mode offers 63 kbit/s for an even wider range of WTs (up top 12 WTs), but the jitter ranges up to 1 sec. This is unacceptable for real–time services. We defined already the *operational phase* as the range of WTs, where the service is stable over a wide range of WTs. For this simulations we conclude that the operational phase for the sequential case is 12 and for SMPT only 9. Thus, there exists a trade–off between the range of the operational phase and QoS provided in terms of goodput and jitter. For a smaller operational phase the QoS provided by SMPT is much better than the QoS provided by sequential transmission.

8.2.2.1.2 Impact of receiver–side delay constraint $\tau_{\max}$ In Figures 8.48 and 8.49 we plot the probability of successfully delivering a video frame as a function of the number of ongoing video streams for the sequential transmission mode and *Slow Healing* SMPT. We give the probability of successful video frame delivery for the delay bounds $\tau_{\mathrm{d}elay}$ = 50, 100, 150, 200, and 250 msec. We note that the link layer at the sending wireless terminal is not aware of these delay bounds. The link layer simply tries to transmit the LPDUs in its buffer; it is not aware of the fact that the LPDUs carry video frames with playout deadlines. Thus, our approach preserves the isolation of the layers of the networking protocol stack and allows for the deployment of our mechanisms in low–cost wireless terminals. At the receiver's transport layer only the video frames meeting the delay bound are passed up to the application. We observe from Figure 8.48 that only a very small fraction (less than one percent) of the video frames is transmitted successfully with the sequential transmission mode for the chosen delay bounds. As expected, larger delay constraints result in a larger probability of successful video frame delivery. However, even for a delay bound of $\tau_{\mathrm{d}elay} = 250$ msec, less than one percent of video frames are transmitted successfully. (We note that this very poor performance is due to the relatively large default size of the link layer buffer of $L_{\mathrm{Q}ueue} = 100$ LPDUs. The link layer is not aware of the video frame deadlines and transmits all the LPDUs in the buffer, even though they may carry video frames that have already missed their deadline. The larger the buffer the more delay the LPDUs may experience in the buffer. We study the impact of the buffer size in detail in Section 8.2.2.1.3.)

We observe from Figure 8.49 that with the *Slow Healing* SMPT mechanism the probability of sending a video frame successfully is very high. For nine and less wireless terminals (each sending one video stream) the success probability for a delay constraint of $\tau_{\mathrm{d}elay} = 250$ msec is 98%. Smaller delays constraints $\tau_{\mathrm{d}elay}$ decrease the success probability, but even for $\tau_{\mathrm{d}elay} = 150$ msec, the success probability is still around 60%. We conclude that for its operational phase up to nine ongoing video streams, the *Slow Healing* SMPT mechanism achieves high probabilities of successful video frame transmission. The sharp drop–off of the success probability for SMPT for ten wireless terminals is due to the increasing interference level in the wireless cell, which results in a steep increase in the bit error probability at that point.

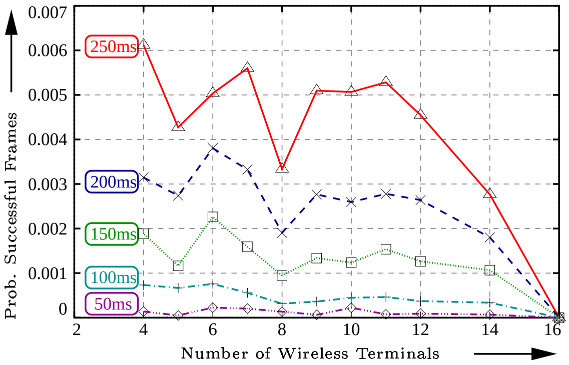


Figure 8.48: Successfully transmitted H.263 frames with receiver–side delay constraints of $50, 100, 150, 200, 250$ ms for the sequential transmission mode.

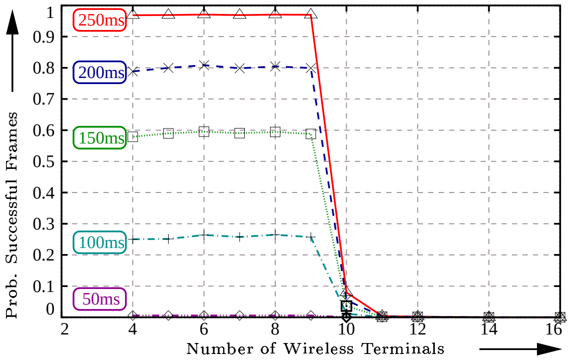


Figure 8.49: Successfully transmitted H.263 frames with receiver–side delay constraints of $50, 100, 150, 200, 250$ ms for the *Slow Healing* transmission mode.

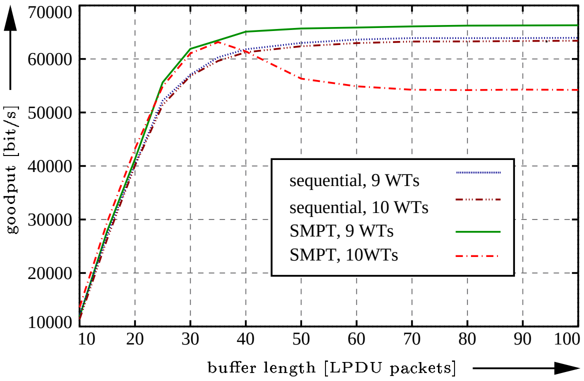


Figure 8.50: Goodput for rate controlled H.263 frames versus buffer length.

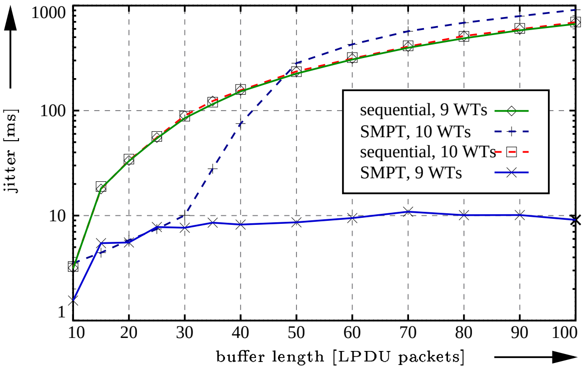


Figure 8.51: Jitter for rate controlled H.263 frames versus buffer length.

8.2.2.1.3 Impact of sender–side buffer size L_{Queue} For the simulation in Section 8.2.2.1.1 and in Section 8.2.2.1.2 we set the buffer at the link layer to 100 LPDU (12.8 kbyte) packets. This leads to large jitter values for the sequential transmission mode. In this section we conduct an investigation of the sender–side buffer size. It is well known that the jitter value can be limited by limiting the buffer length. On the downside, a smaller buffer results in a larger loss rate, and consequently in a smaller goodput. To quantify this trade–off we present the goodput and the jitter values as functions of the buffer size in Figures 8.50 and 8.51, respectively. We investigate the situation where nine and ten WTs are in the cell, using either sequential transmission or SMPT. The sequential transmission has two characteristics: 1.) The cell load (either nine or ten WTs) has no significant impact. 2.) As expected, both the jitter and the goodput decrease as the buffer size is decreased. Reducing the buffer size to 30 LPDU packets, for instance, results in a jitter value of 100 ms and a goodput of 56 kbit/s. For a buffer size of 30 LPDU packets there is no significant difference between the performance for nine or then WTs when SMPT is used. Note that the operational phase for SMPT includes nine WTs with a buffer size of 100 LPDU packets. For nine WTs the jitter is value always below 10 ms and the goodput reaches 66 kbit/s for large buffers. For ten WTs supported with SMPT the jitter is below 10 msec for small buffers (30 LPDU packets or less); for larger buffers the jitter increases dramatically. The important implication of this experiment is that SMPT achieves both large goodput and small jitter for a small buffer size (around 30 LPDU packets).

As noted above, real–time video transmission requires that the video frames are delivered within a tight delay bound. To achieve a tight delay bound the buffers should be small. In Figures 8.52, 8.53, 8.54, and 8.55 we plot the probability of sending a video frame successfully as a function of the frame size (in byte) and the resulting jitter for *Slow Healing* SMPT ($R_{max} = 3$) and the sequential mode for two different buffer values of 2560 byte and 5120 byte is given. The probabilities are given for a scenario with nine WTs in the cell. The values are measured at one specific WT. The bi–modal frame distribution of H.263 video encoder (as given in Chapter 5) is all figures in common. Furthermore, it is noticeable that SMPT yields better jitter values than the sequential transmission mode. In Figure 8.52 and 8.54 the jitter values for smaller frames are higher than for larger frames. The reason for this is that larger frames will fit only in an almost empty queue and therefore will not suffer by retransmission of preceding frames. Smaller frames fit always in the queue and therefore suffer by retransmissions of preceding frames. Thus the frame loss is higher for larger frames than for smaller frames if the sequential transmission mode is used. In Figure 8.53 and 8.55 the SMPT case is given. For SMPT even smaller frames have slightly higher values than the larger frames. The overall jitter values are smaller than for the sequential case, but nevertheless smaller frames achieve larger jitter values than larger frames do. But in case of SMPT transmission mode the larger jitter values for smaller frames is no result of the queuing process. The reason is that larger frames, after being affected by the error–prone wireless link, can be healed while the transmission process. Frames that are smaller than the error duration on the wireless link have to be healed after this bad channel.

8.2.2.1.4 Impact of sender–side knowledge For the following scenario we assume that all WTs take the video frames and try to transmit them towards the BS within a given maximum tolerable delay $\tau_{\max}$. We assume that $\tau_{\max}$ is well know at the sender–side. Once this delay $\tau_{\max}$ is exceeded at the sender the video frame is discarded and the DLC entity starts to transmit the next video frame. Furthermore we assume that **W**elch **B**ound **E**quality (*WBE*) sequences are used.

The results are given in Figure 8.56 for the sequential transmission and in Figure 8.57 for the *Slow Healing* SMPT approach. On the z–axis the video segment loss probability versus the maximum tolerable delay (x–axis) and the number of WTs in the cell (y–axis) is given. Together with the sequential and the SMPT approach the optimal video segment loss probability is given (always the lower surface). The results will never be better than this surface, because this is the video segment loss probability for an error–free communication with the same bandwidth. In spite of error–free communication video frames are lost because of their size (please refer to Figure 6.5).

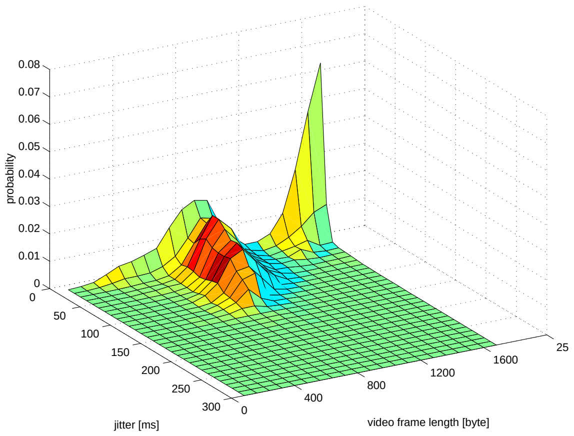


Figure 8.52: Probability of sending a H.263 video frame successfully versus the frame size and the resulting jitter for the sequential transmission mode for buffer size of 2560 byte.

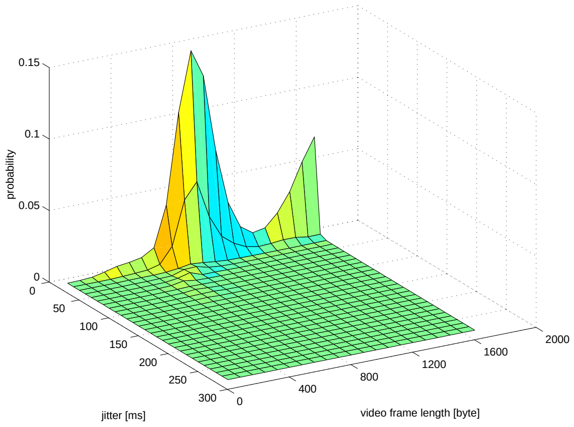


Figure 8.53: Probability of sending a H.263 video frame successfully versus the frame size and the resulting jitter for the *Slow Healing* SMPT transmission mode for buffer size of 2560 byte.

In Figure 8.56 the loss probability for the sequential transmission is given. The **S**egment **L**oss **P**robability (*SLP*) is always large, indifferent of the maximum allowed delay or number of WTs. The smallest SLP is achieved for a maximum tolerable delay τ_{max} of 400 ms. For this case the value of the SLP equals 20%. For lower values of τ_{max} the SLP value increases because larger frames are skipped. For larger values of τ_{max} the SLP value increases as well. For this scenario even the larger frames fit into the queue. This results in a higher load of the cell because more WTs have something to send. Therefore the bit error probability increases and frames will be skipped at the sender because by retransmissions they exceed the transmission window. By increasing the number of WTs even for $\tau_{\mathrm{max}} = 400$ ms the SLP increases.

In Figure 8.57 the loss probability for SMPT is given. The results can be divided into two regions. The first region covers the situation where more than 18 WTs are in the cell. Then the loss probability for SMPT equals that of the sequential case and is not acceptable for real–time communication. The second region covers the situation where 18 or less WTs are in the cell. For this situation the SLP decreases for higher maximum allowed delay values. In comparison to the optimal video segment loss probability the SMPT loss probability is only sightly larger.

Thus, for this scenario where the sender is aware of the maximum allowed delay for each frame, SMPT yields significantly better SLP values than the sequential case for a limited number of WTs. The SLP values for SMPT can be four magnitudes smaller than for the sequential transmission mode (for $\tau_{\mathrm{max}} = 800$ ms, WT ≤ 18).

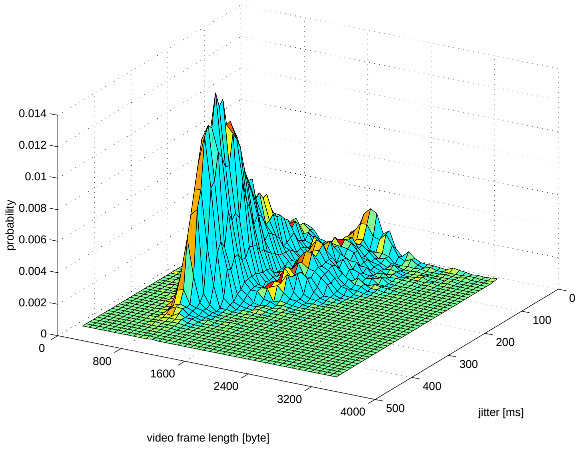


Figure 8.54: Probability of sending a H.263 video frame successfully versus the frame size and the resulting jitter for the sequential transmission mode for buffer size of 5120 byte.

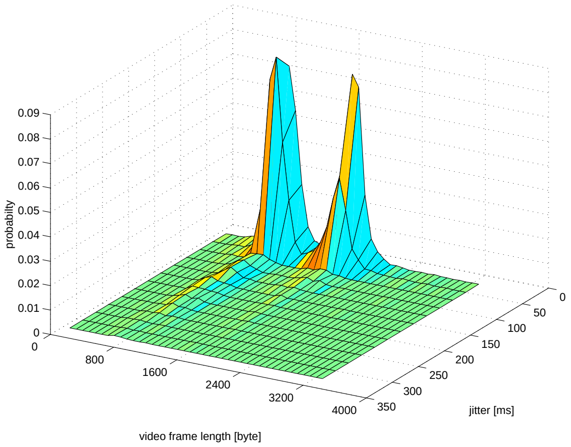


Figure 8.55: Probability of sending a H.263 video frame successfully versus the frame size and the resulting jitter for the *Slow Healing* SMPT transmission mode for buffer size of 5120 byte.

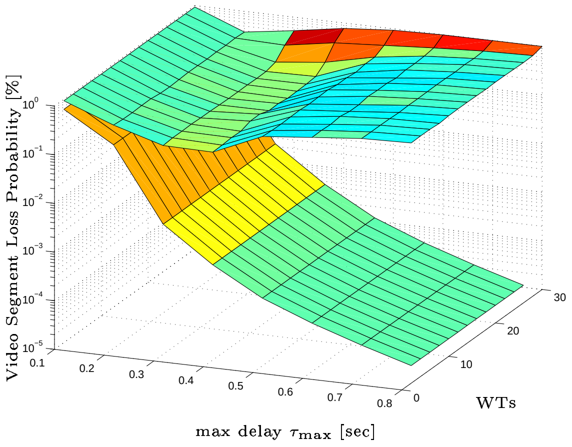


Figure 8.56: Video segment loss probability for rate controlled H.263 video source versus number of WTs and maximum allowed delay $\tau_{\max}$ for the sequential transmission approach and the optimal case.

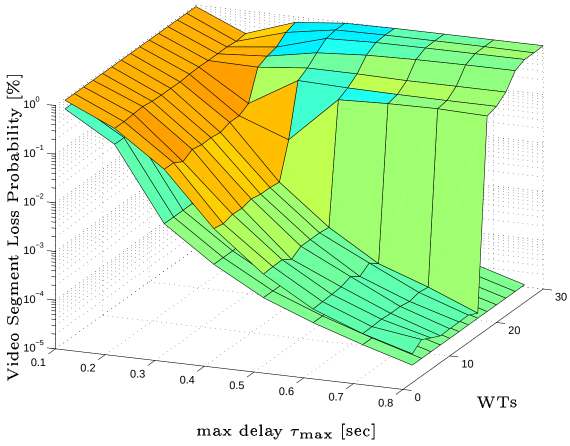


Figure 8.57: Video segment loss probability for rate controlled H.263 video source versus number of WTs and maximum allowed delay $\tau_{\max}$ for SMPT approach and the optimal case.

8.2.2.2 MPEG–4 Without Rate Control

In this section we study the uncoordinated real–time transmission of video encoded without rate control (i.e., in an open loop). For these simulations we used the frame size traces of 25 videos encoded in MPEG–4 with low quality (available from [134]). This open–loop encoding avoids the complexity introduced by rate control. (Also, it results in a constant video quality at the encoder output, whereas the video quality at the encoder output is slightly variable when rate control is employed). However, the traffic produced by open–loop encoding is highly variable. Not only are the individual video streams highly variable in the sizes of their video frames, but also the different video streams differ significantly in their frame size statistics, e.g., the video streams vary in their mean bit rate (see [134] for the exact statistical properties of the 25 video traces used). These variability pose a particular challenge for the network transport. As an example the mean frame length aggregated over 800 frames versus time is given in Figure 8.71 for eight different video sequences. The frame lengths differ dramatically over time among the video sequences. The red line in Figure 8.71 shows the frame length versus time and the green line gives the mean frame length. Therefore, it would not be sufficient to simply stabilize the wireless link for a fixed data rate as we have already shown in Section 8.1.2.2.2 for TCP measurements. Therefore we apply the *Start–Up* SMPT mechanisms for this scenario (see Section 4.4.1.1 for details). In Figure 8.58 the usage of multiple channels for link stabilization and support for variable bandwidth versus time is given.

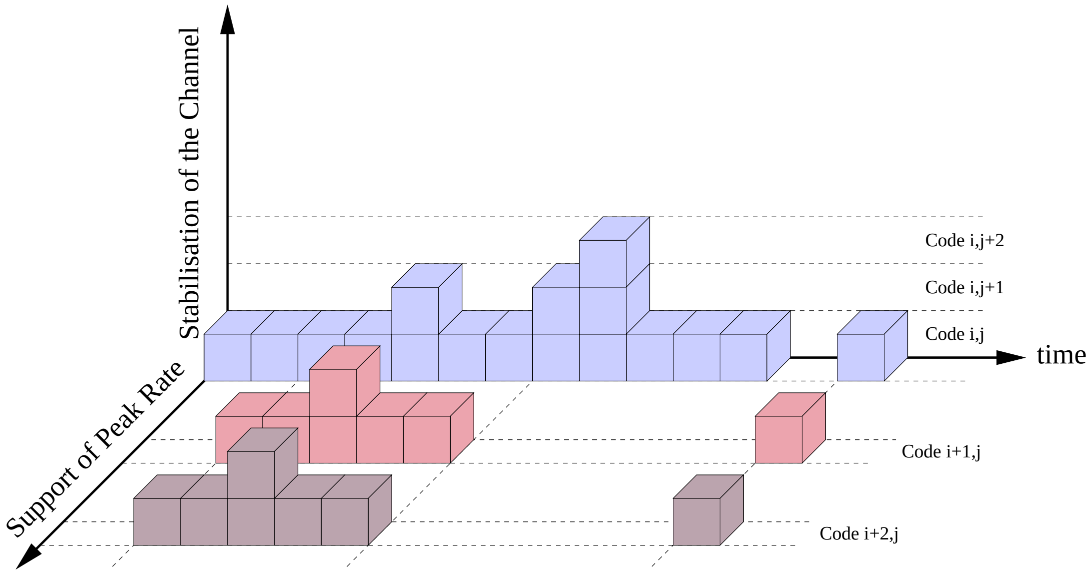


Figure 8.58: Support of variable bit–streams with SMPT using *StartUp*.

Two different *Start–Up* mechanisms were introduced; namely *Slow Start* and *Fast Start* SMPT. In contrast to the preceding performance evaluation we increase the spreading gain up to 32 such as for the MPEG–4/TCP measurements.

Throughout this section the spreading gain is set to a default value of 32 (whereas it was 16 throughout the preceding section). The reason for this larger spreading gain setting is that CDMA systems generally achieve better statistical multiplexing for larger spreading gains. In the previous section the goal of the *Slow Healing* SMPT transmission mechanism was to stabilize the throughput of the wireless link to the target bit rate of the rate–controlled video encodings; which have only small variations and hence require only a small amount of statistical multiplexing. For the highly variable open–loop encoded video considered in this section, a significant amount of statistical multiplexing is required for efficient transport.

8.2.2.2.1 Impact of number of wireless terminals We investigate the effect of number of wireless terminals on goodput, losses, and the jitter value. We consider three transmission approaches : 1.) sequential (with a bit rate of 64 kbps (i.e., a spreading gain of 32) and a bit rate of 128 kbps (i.e., a spreading gain of 16)), 2.) *Slow Start*, and 3.) *Fast Start*. For this scenario no sender side knowledge is assumed and therefore no constraints on the delay or the jitter are take into account. Losses occur only if the sender side buffer can not stored an incoming video frame.

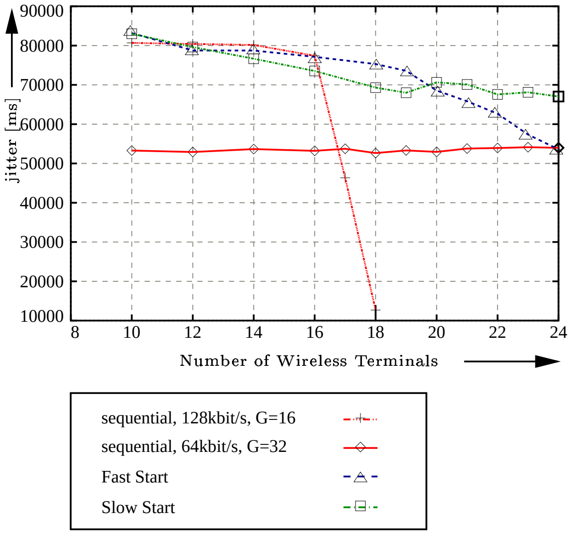


Figure 8.59: Goodput versus number of wireless terminals for MPEG–4 video sequences.

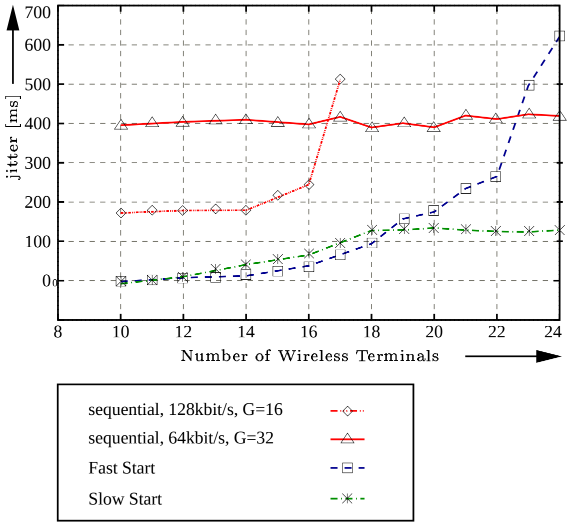


Figure 8.60: Jitter versus number of wireless terminals for MPEG–4 video sequences.

In Figure 8.59 we plot the goodput as a function of the number of wireless terminals for the different transmission approaches. We observe that the sequential transmission mode with 64 kbps has a stable goodput of 53 kbps over the entire considered range of the number of wireless terminals (each transmitting one video stream). The sequential transmission mode with 128 kbps, which effectively transmits always on two CDMA code channels and produces therefore significantly larger interference, achieves a higher goodput of approximately 80 kbps up to 16 ongoing video streams. Up to 20 ongoing video streams the *Fast Start* SMPT approach achieves the highest goodput. For more video streams the goodput drops slowly to the level of the sequential transmission mode with 64 kbps. The *Slow Start* SMPT approach gives a slightly smaller goodput than the *Fast Start* SMPT approach for less than 20 video streams, but for more video streams it achieves the largest goodput.

In Figure 8.60 we plot the average jitter J as a function of the number of wireless terminals. Sequential transmission with 64 kbps gives an average jitter J of 400 msec. This jitter is too large for real–time video transmission, considering that for video conferencing the jitter constraint is typically $\tau_{\mathrm{j}itter} = 150$ msec. By doubling the bit rate a smaller jitter (which is still above the threshold for video conferencing) is achieved for up to 16 ongoing video streams. Only the SMPT mechanisms achieve jitter values that are acceptable for real–time communication. For up to 18 wireless terminals in the cell the *Fast Start* SMPT mechanisms gives a slightly smaller jitter than the *Slow Start* SMPT mechanism. With a larger number of ongoing video streams in the cell, the *Fast Start* SMPT mechanism becomes unstable. This is because the *Fast Start* SMPT mechanism always uses all R CDMA code channels without taking the interference level in the cell (governed by the other terminals' activities) into consideration. The *Slow Start* SMPT mechanisms, on the other hand, gives an average jitter below 150 msec even for 24 ongoing video streams. The *Slow Start* SMPT mechanism gives stable performance as the cell load increases since it probes out the capacity in the cell by slowly increasing the number of used CDMA codes.

8.2.2.2.2 Impact of receiver–side delay constraint $\tau_{\max}$ In Figure 8.61, 8.62 and 8.63 the successfully transmitted MPEG–4 frames are given versus the number of WTs for the sequential and two SMPT transmission mode (*Slow Start* and *Fast Start*). As in Section 8.2.2.1.2 the investigation concentrates on successfully transmitted frames, which are frames that have delay values smaller than $50, 100, 150, 200, 250$ ms at the receiver. At the sender these delay constraints are not known. The sender always tries to send the frames as fast as possible. In Figure 8.61 successfully transmitted MPEG–4 frames are given for the sequential case. The probability to send video frames successfully is stable over a wide range of WTs. The value depends only on the delay constraint. For a delay constraint of 50 ms and 250 ms the probability of sending successfully is 15% and 40%, respectively. Higher success probabilities can be achieved by the SMPT approaches.

In Figure 8.62 successfully transmitted MPEG–4 frames are given for the SMPT *Fast Start* approach versus the number of WTs. For *Fast Start* the probability of sending a segment successfully is very high if the number of WTs is below 20 WTs and the delay constraint is higher than 150 ms. For less than 20 WTs the success probability for a maximum delay of 250 ms is 90%. Smaller maximum delays decrease the success probability. For more than 18 WTs the success probability decrease significantly with each additional WT. This means that for this scenario and the *Fast Start* SMPT transmission mode the maximum number of WTs is set to 18 WTs.

In Figure 8.63 successfully transmitted MPEG–4 frames are given for the SMPT *Slow Start* approach versus the number of WTs. Even for this SMPT approach the success probability is quite high. In contrast to the sequential transmission mode the success probability is decreasing with each WT in the cell. But the success probability is always higher for the *Slow Start* than for the sequential case.

For comparison different transmission schemes and the resulting probability for successfully transmitted MPEG–4 frames with a maximum receiver–side delay constraint of

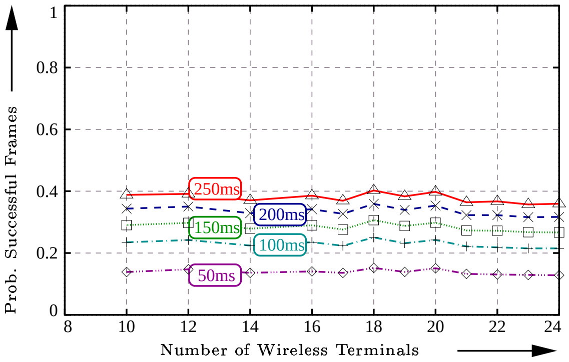


Figure 8.61: Successfully transmitted MPEG–4 frames with receiver–side delay constraints of 50, 100, 150, 200, 250 ms for the sequential transmission mode.

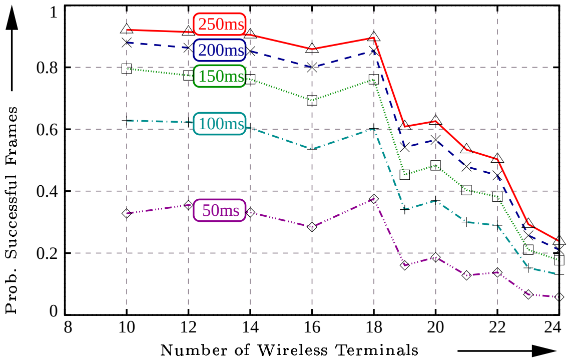


Figure 8.62: Successfully transmitted MPEG–4 frames with receiver–side delay constraints of 50, 100, 150, 200, 250 ms for the *Fast Start* transmission mode.

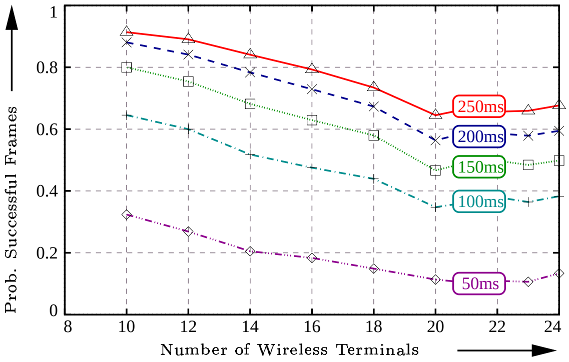


Figure 8.63: Successfully transmitted MPEG–4 frames with receiver–side delay constraints of 50, 100, 150, 200, 250 ms for the *Slow Start* transmission mode.

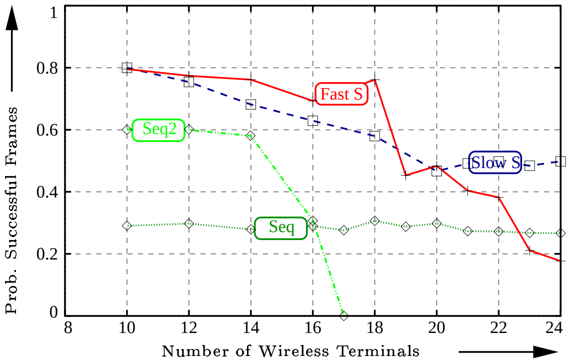


Figure 8.64: Comparison of different transmission schemes for successfully transmitted MPEG–4 frames with a maximum receiver–side delay constraint of 150 ms.

150 ms is given in Figure 8.64. Additionally to the SMPT approaches (`FastS` and `SlowS`) and the sequential transmission mode (`Seq`), a sequential transmission with doubled bit rate and half the spreading gain is given also (`Seq`^{2}). While the *Fast Start* approach yields the best success probability for a smaller set of WTs, the *Slow Start* approach has only slightly worse probabilities for the situation up to 18 WTs. In contrast to the *Fast Start* approach the *Slow Start* approach does not become unstable. Moreover it has a graceful degradation with each additional WT. Note, the *Slow Start* approach yields always better results than the sequential approach. It has to be mentioned that the sequential transmission mode with doubled bit rate achieves better results than the sequential for less than 16 WTs, but becomes unstable afterwards.

8.2.2.2.3 Impact of sender–side buffer length L_{Queue} As in Section 8.2.2.1.3 for rate controlled H.263 video, we investigate the effect of sender–side buffer length L_{Queue} for MPEG–4 video without any target bit rate. We investigate the impact of the buffer length on the jitter and the loss probability. Smaller buffers result in larger losses and smaller jitter values. The trade–off between jitter and loss probability has always to be considered. In Figure 8.66 the jitter value versus the buffer size for 18 WTs and different transmission schemes are given. The buffer length is given in LPDU packets. Each LPDU packet is 80 byte long. The transmission schemes are *Slow Start* SMPT, *Fast Start* SMPT, and the sequential approach. For the given load scenario the SMPT schemes yields better results than the sequential transmission. Among the SMPT mechanisms *Fast Start* results in lower jitter values than *Slow Start*. As mentioned earlier, we have to consider the trade–off between jitter and losses. Therefore the goodput versus the buffer length for 18 WTs and the same transmission schemes as before are given in Figure 8.65. Even for the losses *Fast Start* yields slightly better results than *Slow Start*, but achieves a much better loss probability than the sequential approach.

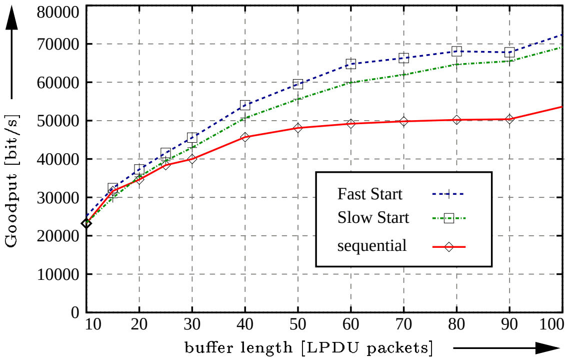


Figure 8.65: Goodput versus buffer length for 18 WTs and different transmission schemes.

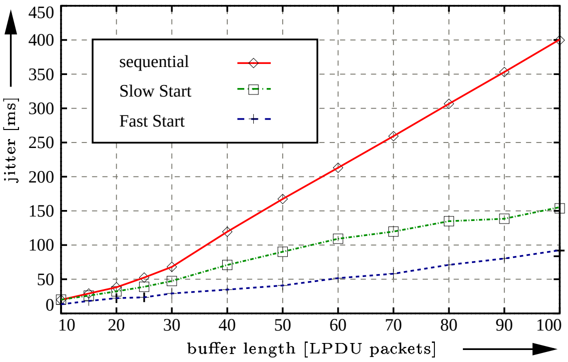


Figure 8.66: Jitter versus the buffer length for 18 WTs and different transmission schemes.

While for short buffer length the jitter and loss probabilities not differ significantly among the transmission approaches, for larger buffer length the differences become obvious. Note, larger buffer length result in high cost end–systems. For the sequential transmission scheme a buffer length of 2560 byte (20 LPDU packets) seems reasonable.

For this buffer length the jitter value is small and the loss rate is nearly stable. This means a larger buffer length will not lead to a significant improvement. For the SMPT approach buffer length between 6400 - 8960 byte (50 up to 70 LPDU packets) gives both a small jitter and a small loss probability.

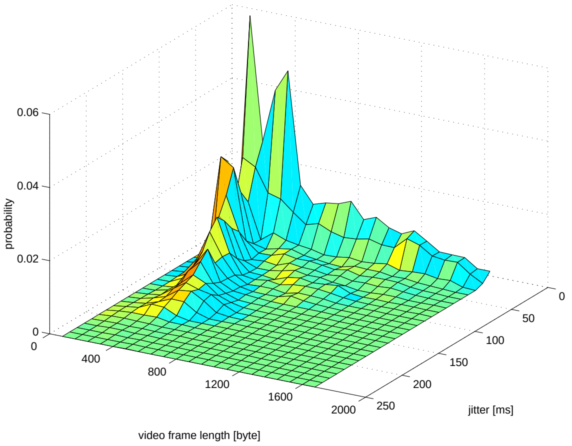


Figure 8.67: Probability of sending a MPEG–4 video frame successfully versus the frame length and the resulting jitter for the sequential transmission mode for buffer length of 2560 byte.

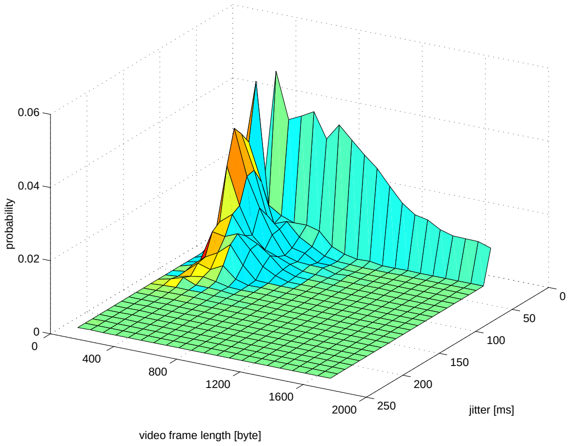


Figure 8.68: Probability of sending a MPEG–4 video frame successfully versus the frame length and the resulting jitter for the *Slow Start* SMPT transmission mode for buffer length of 2560 byte.

In Figures 8.67, 8.68, 8.69, and 8.70 we plot the probability masses of the jitter J (in msec) as a function of the video frame size (in byte). We consider the sequential transmission mode and *Slow Start* SMPT in this experiment. The link layer buffer is set to $L_{\mathrm{Q}ueue}$ = 20 LPDUs (2560 bytes) and $L_{\mathrm{Q}ueue}$ = 40 LPDUs (5120 bytes)), respectively. There are 18 wireless terminals sending one video stream each in the cell. Note that *Slow Start* SMPT may transmit a video frame faster than the sequential transmission would over an error–free channel. This results in negative jitter values J for the *Slow Start* SMPT mechanism, which we count as a jitter of zero. We observe from the figures that *Slow Start* SMPT achieves generally smaller jitter values than sequential transmission, with the differences being more pronounced for the larger link layer buffer. As we have observed for the rate–controlled video in Section 8.2.2.1.3, it is also the case for video without rate control that smaller video frames experience a larger jitter. This is again due to the fact that smaller video frames fit more easily into the link layer buffer and are more likely to be delayed by video frames ahead of them in the buffer. This effect is more pronounced for sequential transmission than for SMPT. For the large buffer, small frames experience jitter values of up to 500 msec with sequential transmission, whereas the jitter values are less than 200 msec with SMPT.

Note that most of the given results in this section are for a cell load of 18 WTs. As given in Figure 8.60 the *Fast Start* approach yields the best performance for 18 or less

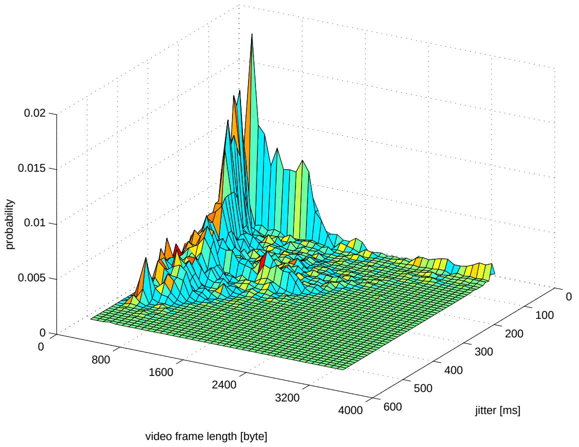


Figure 8.69: Probability of sending a MPEG–4 video frame successfully versus the frame length and the resulting jitter for the sequential transmission mode for buffer length of 5120 byte.

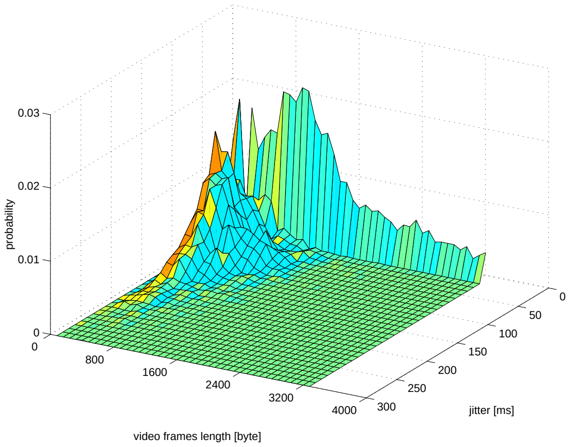


Figure 8.70: Probability of sending a MPEG–4 video frame successfully versus the frame length and the resulting jitter for the *Slow Start* SMPT transmission mode for buffer length of 5120 byte.

WTs. Increasing the number of WTs further leads to instability of this approach. Thus, only a small number of WTs can be allowed with *Fast Start* for the cell. However, the *Slow Start* approach achieves nearly the same good results as the *Fast Start* but is stable for a wide range of WTs. Even for 24 WTs the goodput value, the losses and the jitter value is better than for the sequential case. Therefore we suggest to use the *Slow Start* SMPT approach to support video streams encoded without rate control.

8.2.2.3 Conclusion

We have studied the real–time transmission of video by distributed wireless clients in a wireless cell. We have demonstrated that simple to deploy SMPT mechanisms enable the efficient transmission of video with tight real–time constraints on the order of 150 msec, thus enabling real–time applications, such as video conferencing, games, and tele–medicine. For video encoded with rate control we found that the *Slow Healing* SMPT mechanism achieves high goodput and small video frame loss probability while supporting a large number of simultaneous video streams in the cell (i.e., giving high cell capacity). For the more bursty video traffic resulting from encoding without rate control we found that the *Slow Start* and *Fast Start* SMPT mechanisms provide efficient transmission scheduling to achieve a high cell capacity and good video quality. The *Slow Start* SMPT mechanism is particularly resilient and degrades gracefully as the load on the cell increases. We studied the impact of the link layer buffer, the only hardware component required by the SMPT mechanisms. We gave guidelines for the dimensioning of this buffer.

While we considered the transmissions from distributed wireless terminals to a central base station in a cellular wireless network in our simulations, we emphasize that the presented SMPT mechanisms can be deployed readily in Ad–hoc wireless networks. The SMPT mechanisms do not require any coordination of the transmissions among the distributed clients, and can thus be used for point–to–point transmissions in a cluster of a wireless Ad–hoc network.

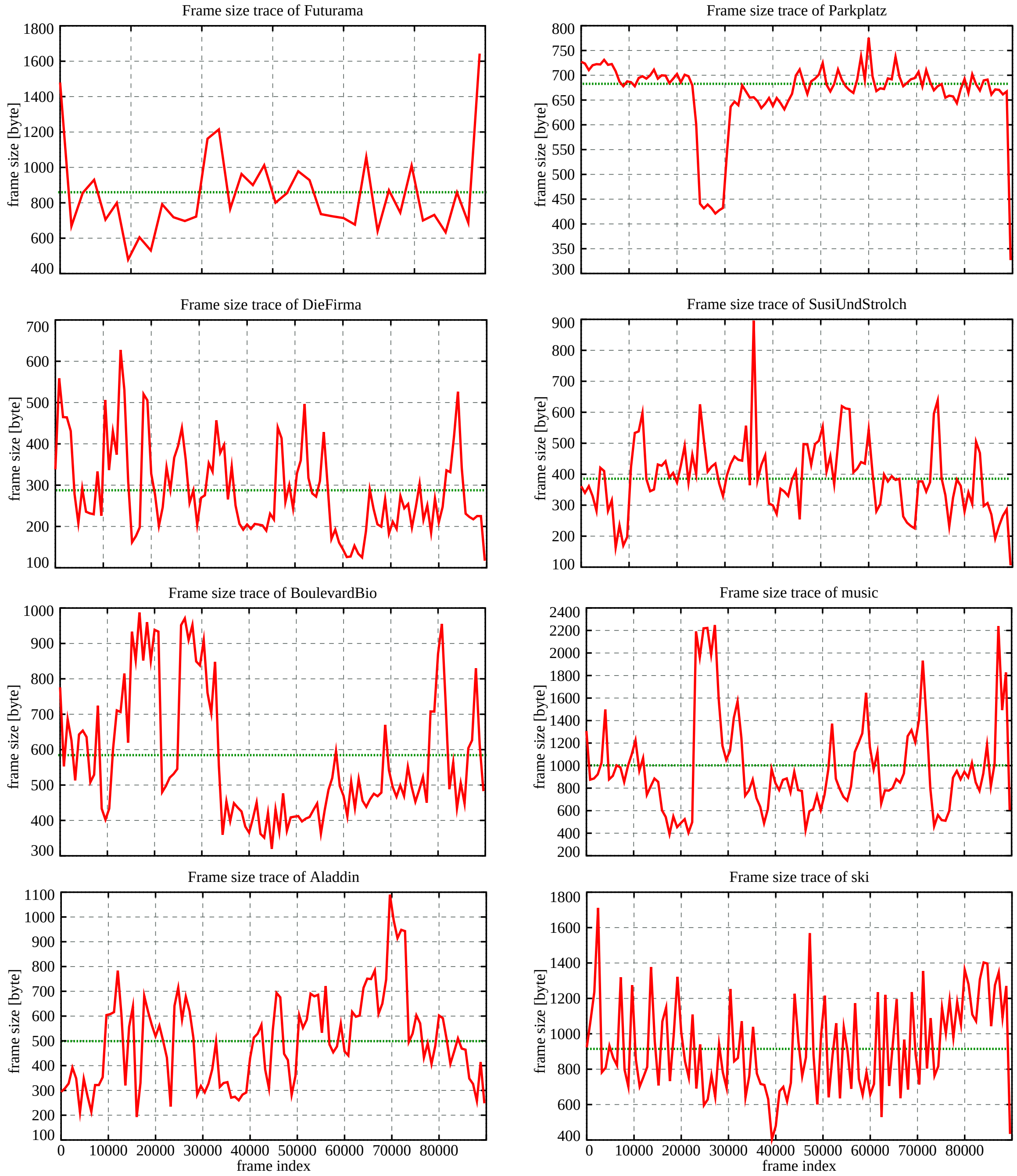


Figure 8.71: Mean frame length aggregated over 800 frames versus time for eight different video sequences.

9 Conclusion and Outlook

This dissertation is concluded by summarizing the major results and pointing out the directions for future research for the **S**imultaneous **MAC** **P**acket **T**ransmission (*SMPT*) approach.

9.1 Conclusion

We have introduced an approach for QoS support in wireless systems. This approach is called **S**imultaneous **MAC** **P**acket **T**ransmission (*SMPT*) . Four different SMPT approaches based on the **M**ulti–**C**ode CDMA (*MC–CDMA*) were introduced for the QoS support of uplink communication in wireless networks. SMPT enabled wireless terminals transmit and receive on multiple CDMA channels. Two approaches, namely *Slow Healing* and *Fast Healing*, try to stabilize the wireless link by using parallel channels for retransmissions of corrupted packets. These approaches are well suited for rate adapted video streams. The other two approaches, namely *Slow Start* and *Fast Start*, are well suited for video streams without any rate control. These approaches use the parallel channels whenever they detect a *good* channel state. All approaches can be applied in *distributed* systems. We also note that throughout our focus has been on mechanisms that are low in complexity and cost and *simple* to deploy yet give tangible performance gains, rather than finding more complex mechanisms that further enhance the performance. In addition to the SMPT approaches, we have presented and studied frame size traces of MPEG–4 and H.263 encoded videos for wireless bandwidth–limited channels in this work. We have encoded over 25 videos. For each video we have generated MPEG–4 and H.263 encodings with seven different quality levels. We have presented a detailed statistical analysis of the generated traces by calculating moments and autocorrelations as well as the long range dependence characteristics. The Hurst parameters of the traces with the R/S statistic are given. The presented traces can be easily used by other researchers for their development and evaluation of protocols and mechanisms for packet–switched networks, wireless networks, and optical networks. We conducted even a simulation interface for the PTOLEMY [159] and NS [160] simulation tool. The results were presented in a publicly available library [135]. In our analytical performance evaluation we have shown that SC–CDMA has a larger capacity than MC–CDMA for all values of the activity factor and number of parallel channels offering lower QoS in terms of loss rate or delay in comparison to SMPT. *Slow Healing* SMPT has the best performance among the SMPT approaches regarding the capacity. Taking the buffer overflow and the buffer occupancy as the QoS metric SMPT performs better than SC–CDMA. Four different SMPT approaches were evaluated by the means of simulations using the **C**onstant **B**it **R**ate (*CBR*) traffic source. It could be shown that mechanisms which use the wireless resources in a moderate way, such as *Slow Start* and *Slow Healing* achieve the best results. *Slow Start* achieves better performance for power saving and the *Slow Healing* mechanism has a smaller variance of the number of used codes. The variance of the *Slow Healing* mechanism is almost as small as for the sequential transmission. This is very important for minimizing the impact on the power control entity. We have demonstrated that using multiple channels can result in performance degradation, if the mechanism (e.g. *Fast Start*) uses the wireless resources excessively. We conclude that it is insufficient to merely equip the WTs with data rate adaptive procedures. It can be observed that the improvement of SMPT is achieved with a relatively low number of additional codes. Besides the CBR sources, we have studied the uplink performance of FTP and streaming video using TCP in cellular CDMA–based wireless systems. TCP has a number of desirable properties, such as network stability and fair bandwidth allocation, for the future Internet. SMPT improves the performance of TCP over the wireless link. While for FTP support SMPT achieves only a small performance gain, we found that SMPT gives good results for rate controlled encoded streamed video. For a given buffer underrun requirement, our *Slow Healing* SMPT approach increases the number of supported streams in a cell in a typical CDMA scenarios. We also observed that SMPT performs better with higher spreading gains. We have demonstrated that SMPT mechanisms enable the efficient transmission of video with tight real–time constraints on the order of 150 msec, thus enabling real–time applications, such as video conferencing, games, and tele–medicine. For video encoded with rate control we found that the *Slow Healing* SMPT mechanism achieves high goodput and small video frame loss probability while supporting a large number of simultaneous video streams in the cell. For the more bursty video traffic resulting from encoding without rate control we found that the *Slow Start* and *Fast Start* SMPT mechanisms provide efficient transmission scheduling to achieve a high cell capacity and good video quality. The *Slow Start* SMPT mechanism is particularly resilient and degrades gracefully as the load on the cell increases. We studied the impact of the link layer buffer, the only hardware component required by the SMPT mechanisms. We gave guidelines for the dimensioning of this buffer.

9.2 Outlook

The following has to be done in future work: 1.) To compare the SMPT approaches with the sequential transmission mode we applied a common abort criteria. Taking the cell load and the service that has to be supported into account this abort criteria can be optimized. 2.) The probing rate R_{probe} could be matched in a dynamic fashion on the wireless link. Less interference will be produced by the probing process. 3.) To increase the capacity and to achieve a graceful degradation the degree of the SMPT ramp can be adjusted according to the cell load. This might introduce further complexity to the SMPT scheme. 4.) In real communication networks a mixture of streams has to be supported. In this work we investigate always the same type of traffic at the wireless terminals. 5.) We have investigated only a single cell environment. Because of the responsive environment a multi–cell scenario has to be investigated.

In the further outlook a short description of further error correction schemes for SMPT and the application of SMPT within a new network architecture is given.

9.2.1 SMPT and Further Error Correction Schemes

In order to reduce the loss rate seen by higher protocol layers either **A**utomatic **R**epeat Re**Q**uest (*ARQ*), **F**orward **E**rror **C**orrection (*FEC*) or Hybrid ARQ can be used to transmit packets. The simulation presented here where limited to a SMPT algorithm which is only based on ARQ (*e.g.* retransmits the original packets). On the other hand it is generally accepted that FEC or a mixture of ARQ and FEC (Hybrid Type I,II) can improve the performance of wireless networks. Now we give a short outlook on FEC schemes for the SMPT algorithm like they were introduced in [21].

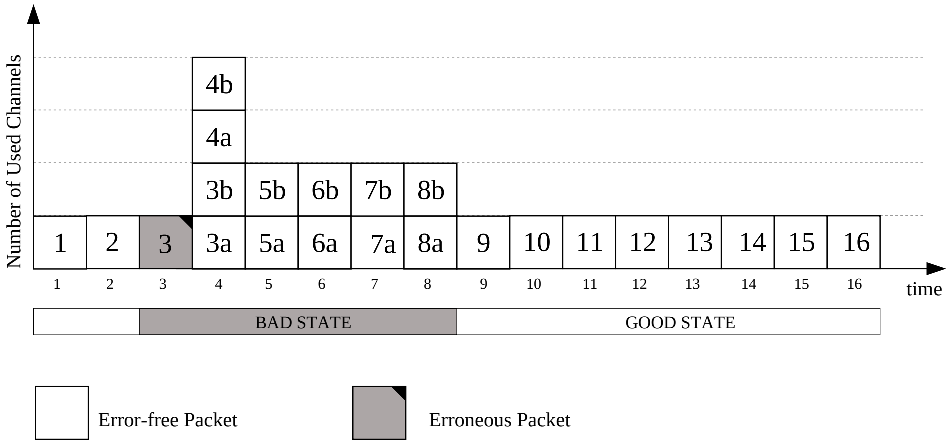


Figure 9.1: SMPT transmission methods with use of vertical FEC.

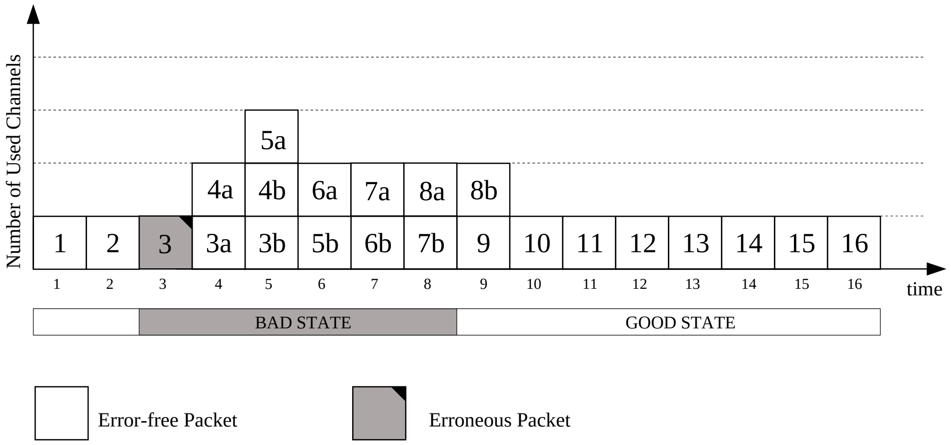


Figure 9.2: SMPT transmission methods with use of horizontal FEC.

In Figure 9.1 the vertical and in Figure 9.2 the horizontal SMPT approach are depicted. The algorithm to use FEC is explained by the example of the vertical SMPT approach:

When the first erroneous packet of one segment is detected (*e.g.* packet 3 in Figure 9.1), the retransmission of the erroneous packet (packet number three) and also the next stored packets are divided in two packets. All of these packets are protected by additional FEC code and each of them is sent on an additional channel. An important question is when to switch off the additional channels. A simple solution would be to switch off when the next segment is transmitted. In this case the FEC is switched of when the bad phase ends. The knowledge when the channels becomes good again is not easy to gather. Errors on the wireless link that occur while using FEC might be adjusted. This approach has the drawback that the FEC code would also be used when no errors occur on the wireless link, which is not very efficient for the spectrum. To overcome the drawback of inefficiency of the spectrum a more sophisticated possibility is proposed in the following.

Another possibility to solve this problem is to protect the original packets of a segment by FEC packets instead of adding FEC code to each packet [197]. Consider a scenario in which packet i of a segment is corrupted. Instead of retransmitting packet i protected with FEC code on a new channel packet $i+1$ is transmitted in combination with a second packet on a new channel which carries FEC code to repair packet i and protect packet $i+1$. The amount of FEC code (e.g. the number of FEC packets on additional channels) is increased as long as the packets are erroneous. Since the original packets are transmitted on the default channel without any additional protection changes from a bad to a good channel are signaled by error free transmission on the default channel. In this case the use of FEC is stopped after all previous packets are repaired and the actual packet do not need FEC. Figure 9.3 illustrates this algorithm. Designing rules for FEC code are out of scope of our considerations in this paper, but are given in [198].

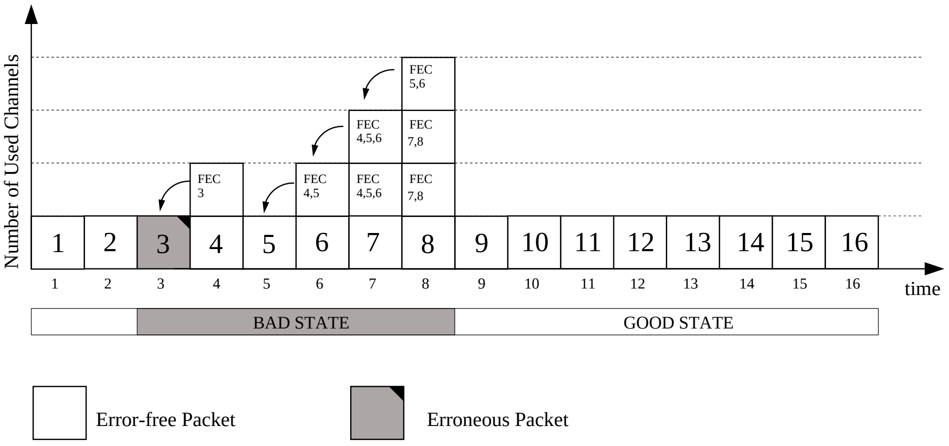


Figure 9.3: SMPT transmission method with FEC (Hybrid Type II).

9.2.2 SMPT for Ad–hoc Networks

In this work the SMPT mechanisms are only applied to cellular communication systems. Here we want to give a short outlook how SMPT can also be applied to mobile *Ad–hoc* networks. Mobile Ad–hoc networks are networks of mobile terminals with wireless interfaces that can dynamically form a network, where any kind of network is missing, incomplete, or only inadequate. Because of the potential ease of deployment, there are a lot of interesting ideas of Ad–hoc networks. Examples of such situations include disaster management, military use or in general a group of people wishing to communicate amongst themselves [199].

In these networks communication is possible in a *point–to–point* fashion between wireless terminals. If the coverage of two terminals is to short to communicate with each other, the communication is setup by *hoping* over neighboring terminals. This allows even to save energy for the terminals.

In this work we considerate always WTs sending and receiving to one central base station. But we mentioned already that we consider each communication between a WT and the BS as one sender receiver pair. While the power control entity at all WT try to meet a given SNR at the BS and other WTs interfere with the same signal energy as the transmitted signal, in an Ad–hoc mode the sender tries to meet a given SNR required by the receiver. The interference among WTs has be to calculated separately for each WT. The interference level of one WTs depends on the signal energy and its distance of each active WTs.

In Figure 9.4 eight WTs are forming one Ad–hoc network. As an example three ongoing communications are given: (a) WT 6 $\longleftrightarrow$ WT 7, (b) WT 1 $\longleftrightarrow$ WT 7, and (c) WT 2 $\longleftrightarrow$ WT 8. The network is divided into three clusters A,B, and C. Cluster A includes all WTs that are in the coverage of WT 2, Cluster B is formed by all WTs that can be reached by WT 4, and Cluster C includes all WTs that are in the range of WT 8.

For the calculation of the interference of WT k all active WTs $j = 1 \ldots J, j \neq k$ have to be taken under consideration. Considering only the free space propagation the interference level I_k for WT k equals

$$I_k = \sum_{j=1 \ldots J, j \neq k} P_j \cdot \frac{C}{d_j^2}, \tag{9.1}$$

where d_j is the distance between WT k and WT j, C a constant value based on the path loss equation given in Section 2.1.1, and P_j the transmitting power of WT j.

In Figure 9.5 one possible transmission scheme of packets in a slotted scheme between WTs in a wireless Ad–hoc network is given. Whenever a packet is transmitted all other transmissions interfere. The interference level is given in Equation 9.1. It also can be seen that the interference level for each cluster differ over the time.

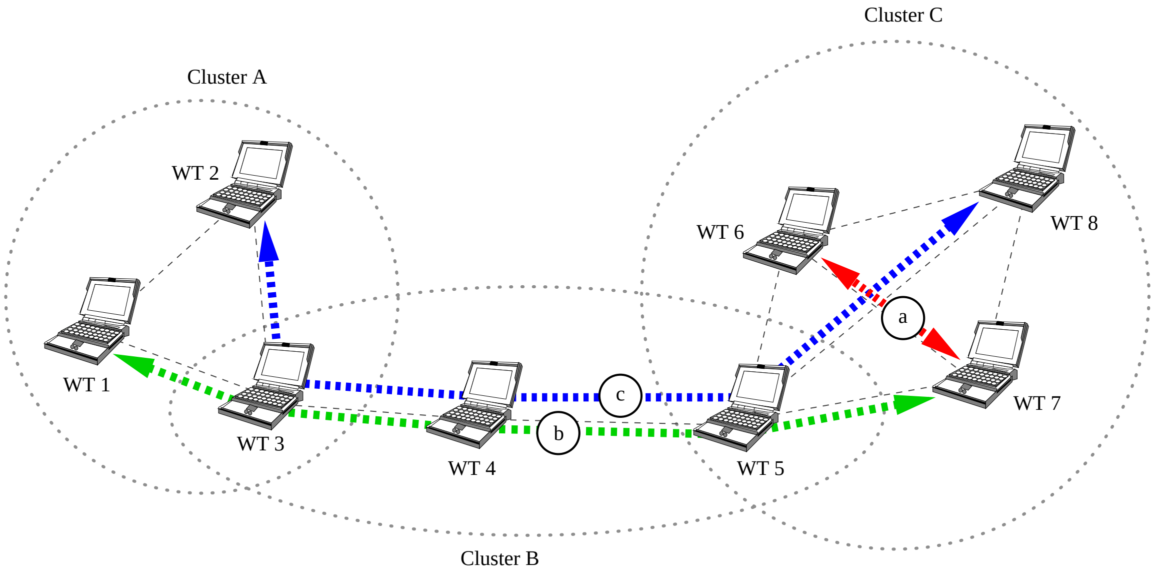


Figure 9.4: Ad–hoc network with eight wireless terminals.

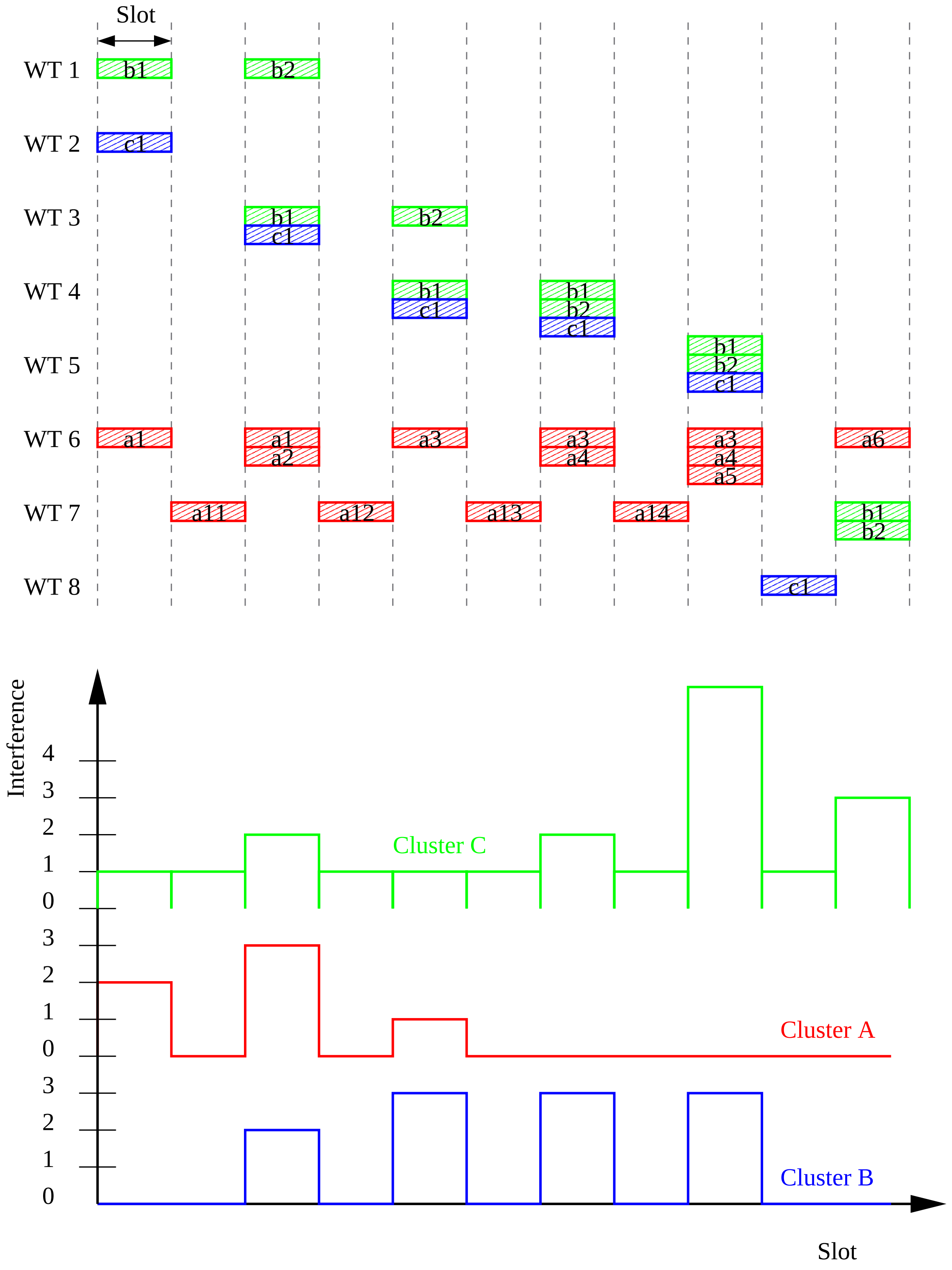


Figure 9.5: Example for Ad–hoc communication and the resulting interference per cluster.

A Acronyms

AMPS	Advanced Mobile Phone System
APC	Adaptive Power Control
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BAP	Bandwidth Allocation Protocol
BEP	Bit Error Probability
BPSK	Binary Phase Shift Keying
BRI	Basic Rate Interface
BS	Base Station
CBR	Constant Bit Rate
CDMA	Code Division Multiple Access
CW	Congestion Window
DCT	Discrete Cosinus Transformation
DE	Discrete Event
DLC	Data Link Control
DLL	Data Link Layer
DQRUMA	Distributed–Queueing Request Update Multiple Access
DSSS	Direct Sequence Spread Spectrum
DS	Direct Sequence
ETSI-SMG	European Telecommunications Standards Institute – Special Mobile Group
FCC	Fundamental Code Channel
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FHSS	Frequency Hopping Spread Spectrum
FTP	File Transfer Protocol
GSM	Global System for Mobile communication
GoP	Group of Picture
HDTV	High Definition Television
HTTP	Hyper Text Transfer Protocol
IBMS	Integrated Broadband Mobile System
IP	Internet Protocol
IS95	Interim Standard 95
ISDN	Integrated Services Digital Network
ISI	Inter Symbol Interference
JSQ	Join the Shortest Queue
LFSR	Linear Feedback Shift Register
LIDA	Load and Interference based Demand Assignment
LoS	Line of Sight
MAC	Medium Access Control
MAI	Multiple Access Interference

MCPA Maximum Capacity Power Allocation

MFSK Multi Frequency Shift Keying

MSC Message Sequence Chart

NACCH Network Access and Connectivity Channel

NFAS Non-Facility Associated Signaling

NLOS Non Line of Sight

NS Network Simulator

NTSC North America Television Standard Committee

OBM Overlapping Batch Means

OFDM Orthogonal Frequency Division Multiplexing

PCMA Power Controlled Multiple Access

PDC Personal Digital Cellular

PDU Packet Data Unit

PEP Packet Error Probability

PN Pseudo–Noise

PPP Point–to–Point Protocol

PRI Primary Rate Interface

QoS Quality of Service

RACH Random Access Channel

RS Reed Solomon

RTCP Real Time Control Protocol

RTO Retransmission Time Out

RTP Real Time Protocol

RTT Round Trip Time

SC–CDMA Single–Code CDMA

SCC Supplemental Code Channel

SLP Segment Loss Probability

SMPT Simultaneous MAC Packet Transmission

SNIR Signal to Noise Interference Ratio

SNR Signal to Noise Ratio

SSMA Spread–Spectrum Multiple Access

TCP Transmission Control Protocol

TDD Time Division Duplex

TDMA Time Division Multiple Access

THSS Time Hopping Spread Spectrum

TPC Transmitter Power Control

TW Transmission Window

UDP User Datagram Protocol

UMTS Universal Mobile Telecommunications System

UTRA UMTS Terrestial Radio Access

VBR Variable Bit Rate

VO Video Objects

VoIP Voice over IP

VoL Video Object Layers

VoP Video Object Planes

VSG Variable Spreading Gain

WBE Welch Bound Equality

WJSQ Wireless Join the Shortest Queue

WT Wireless Terminal

WWW World Wide Web

pdf Probability Density Function

B Statistic

In this appendix we review the statistical definitions and methods used in the analysis of the generated frame size traces. Recall that N denotes the number of frames in a given trace. Also recall that $X_n,\ n = 1, \ldots, N$, denotes the size of frame n in bit.

B.1 Mean, Coefficient of Variation, and Autocorrelation

The (arithmetic) sample mean $\bar{X}$ of a frame size trace is estimated as

$$\bar{X} = \frac{1}{N} \sum_{n=1}^{N} X_n. \tag{B.1}$$

The sample variance S_X^2 of a frame size trace is estimated as

$$S_X^2 = \frac{1}{N-1} \sum_{n=1}^{N} (X_n - \bar{X})^2. \tag{B.2}$$

A computationally more convenient expression for S_X^2 is

$$S_X^2 = \frac{1}{N-1} \left[\sum_{n=1}^{N} X_n^2 - \frac{1}{N} \left(\sum_{n=1}^{N} X_n \right)^2 \right]. \tag{B.3}$$

The coefficient of variation CoV is defined as

$$CoV = \frac{S_X}{\bar{X}}. \tag{B.4}$$

The maximum frame size $X_{\max}$ is defined as

$$X_{\max} = \max_{1 \leq n \leq N} X_n. \tag{B.5}$$

The autocorrelation coefficient $\rho_X(k)$ for lag $k,\ k = 0, 1, \ldots, N$, is estimated as

$$\rho_X(k) = \frac{1}{N-k} \sum_{n=1}^{N-k} \frac{(X_n - \bar{X})(X_{n+k} - \bar{X})}{S_X^2}. \tag{B.6}$$

These statistics are estimated in analogous fashion for the GoP size traces, the *stuffed* frame size traces, and the sampled rate traces. We refer the reader to Law and Kelton [200] for more details on these definitions.

Table B.1: Algorithm for pox diagram of R/S.

```
For $d = 10,\ 20,\ 40,\ 80, \ldots$ do
    $I = K + 1 - \lceil \frac{dK}{N} \rceil$
    For $i = 1, \ldots, I$ do
        $t_i = (i-1)\frac{N}{K} + 1$
        $\bar{X}(t_i, d) = \frac{1}{d}\sum_{j=1}^{d} X_{t_i+j}$
        $S^2(t_i, d) = \frac{1}{d}\sum_{j=1}^{d}[X_{t_i+j} - \bar{X}(t_i, d)]^2$
        $R(t_i, d) = \max\{0,\ \max_{1 \le k \le d} W(t_i, k)\} - \min\{0,\ \min_{1 \le k \le d} W(t_i, k)\}$
            $W(t_i, k) = \left(\sum_{j=1}^{k} X_{t_i+j}\right) - k\bar{X}(t_i, d)$
        plot point $\left(\log d,\ \log \frac{R(t_i,d)}{S(t_i,d)}\right)$
```

B.2 R/S Statistic

We use the R/S statistic [201, 156, 117] to investigate the long range dependence characteristics of the generated traces. The R/S statistic provides an heuristic graphical approach for estimating the Hurst parameter H. Roughly speaking, for long range dependent stochastic processes the R/S statistic is characterized by $E[R(n)/S(n)] \sim cn^H$ as $n \to \infty$ (where c is some positive finite constant). The Hurst parameter H is estimated as the slope of a log–log plot of the R/S statistic.

More formally, the *rescaled adjusted range statistic* (for short *R/S statistic*) is plotted according to the algorithm given in Table B.1.

The R/S statistic $R(t_i, d)/S(t_i, d)$ is computed for logarithmically spaced values of the lag k, starting with $d = 10$. For each lag value d as many as K samples of R/S are computed by considering different starting points t_i; we set $K = 10$ in our analysis. The starting points must satisfy $(t_i - 1) + d \le N$, hence the actual number of samples I is less than K for large lags d. Plotting $\log[R(t_i, d)/S(t_i, d)]$ as a function of $\log d$ gives the *rescaled adjusted range plot* (also referred to as *pox diagram of R/S*). A typical pox diagram starts with a transient zone representing the short range dependence characteristics of the trace. The plot then settles down and fluctuates around a straight "street" of slope H. If the plot exhibits this asymptotic behavior, the *asymptotic Hurst exponent* H is estimated from the street's slope using a least squares fit.

To verify the robustness of the estimate we repeat this procedure for each trace for different aggregation levels $a \geq 1$. The aggregated trace $X_n^{(a)}$, $n = 1, \ldots, N/a$, is obtained from the original trace X_n, $n = 1, \ldots, N$, by averaging over non–overlapping blocks of length a, i.e.,

$$X_n^{(a)} = \frac{1}{a} \sum_{j=(n-1)a+1}^{na} X_j.$$

B.3 Calculation of the Confidence Interval

In order to ensure that the made predicates have a statistical security, it is necessary to control the temporal duration of the simulation. For this purpose a module named *StatisticCalc* is integrated into each sender–receiver model where it records results and calculates half–length H for the confidence interval. The half–length is the momentary deviation from the current average value.

Given a number of values x_n then the mean value is:

$$\bar{x}(n) = \frac{\sum^n x_i}{n} \tag{B.7}$$

and the variance of the mean values $\bar{x}(n)$ is [202]:

$$\sigma_n^2 = \frac{\sum(x_i - \bar{x}(n))^2}{n-1} \tag{B.8}$$

Thus the half–length H of the confidence interval, which indicates the precision of the mean value, which is to be situated within statistical security z,

$$H = z \cdot \sqrt{\frac{\sigma_n^2}{n}} \tag{B.9}$$

whereby H is situated in both negative and positive direction of the mean value $\bar{x}(n)$.

$$[\bar{x} - H, \bar{x} + H] \tag{B.10}$$

In the above notation σ_n^2 is the variance and n the number of simulation values. The value z corresponds to a certain safety limit of the confidence interval that can be stated. The corresponding confidence level values of z can be taken from Table B.2. These values come from the inverse Student T distribution.

Table B.2: Confidence levels and appropriate z–value.

confidence level	z
0,90 %	1,645
0,95 %	1,966
0,99 %	2,576

If a simulation is started, each statistics module announces itself at the control unit. The control unit creates a table of the sender–receiver model. Now if a partial model achieves statistical security, this is announced by the respective statistics module to the control unit. The control unit notes this in the table created before. Only if all partial models achieve statistical security, the entire simulation is terminated by the control unit.

C Sophisticated Gaussian Approximations

In Section 2.3.3.1 the standard and the Holtzman Gaussian approximation is given. More sophisticated Gaussian approximations for higher modulation schemes than **B**inary **P**hase **S**hift **K**eying (*BPSK*) and RAKE systems (see Section 2.3.2.1) are discussed here. In [203], Lok and Lehnert extended both the standard Gaussian approximation and Holtzman's Improved Gaussian approximation for quadriphase modulated DS–CDMA communication systems. The extended Holtzman's Improved Gaussian approximation for BPSK is given in Section 2.3.3.1, Equation 2.24. For quadriphase modulated DS–CDMA communication systems the error probability is given in Equation C.1.

$$p_{biterror}^{Lok}(k) = \frac{2}{3}Q\left(\sqrt{\frac{G_{Spreading}}{\bar{V}}}\right) + \frac{1}{6}Q\left(\sqrt{\frac{G_{Spreading}}{\bar{V}+\sqrt{3}\sigma_V}}\right) + \frac{1}{6}Q\left(\sqrt{\frac{G_{Spreading}}{\bar{V}-\sqrt{3}\sigma_V}}\right) \qquad (C.1)$$

Table C.1: Mean and variance of $\bar{V}$ for different types of channel modulation.

Modulation	$\bar{V}$	σ_V^2
QPSK	$\frac{2}{3}(k-1)$	$\frac{1}{45}(k-1)$
OQPSK	$\frac{2}{3}(k-1)$	$\frac{17}{1440}(k-1)$
MSK	$\frac{15+2\pi^2}{6\pi^2}(k-1)$	$\frac{-16830+1842\pi^2+77\pi^4}{2304\pi^4}(k-1)$

Furthermore in [204], Kaiser introduced a bit error probability calculation for Rayleigh fading channels using RAKE systems. The error probability for k WTs is given in Equation C.2

$$p_{biterror}^{Rake}(k) = \left(\frac{1}{2} - \frac{1}{2}\sqrt{\frac{2(k-1)}{3GD}+1}\right)^{D} \cdot \sum_{j=0}^{D-1} \binom{D-1+j}{j} \left(\frac{1}{2} + \frac{1}{2}\sqrt{\frac{2(k-1)}{3GD}+1}\right)^{j}, \qquad (C.2)$$

where D equals the number of branches in a RAKE system.

D Deutsche Zusammenfassung

Der Mobilfunk hat in dem letzten Jahrzehnten eine bemerkenswerte Entwicklung vollzogen. Aufgrund der allgemeinen Nachfrage der Mobilkunden nach mehr Mobilität erfreuen sich drahtlose Endgeräte vermehrter Beliebtheit. Auch zukünftig wird die Anzahl der drahtlosen Endgeräte stetig steigen. In einer Studie [2] wurde die Anzahl der Mobilkunden auf über eine Milliarde für das Jahr 2010 prognostiziert. In Abbildung D.1 ist die weltweite Entwicklung der drahtlosen Mobilkunden in Regionen aufgezeigt. Parallel zu der Entwicklung des Mobilfunks wird auch die Anzahl der Benutzer des Internets stark ansteigen. In einer weiteren Studie [1] wird davon ausgegangen, daß bereits in diesem Jahrzehnt mehr Mobilkunden das Internet über drahtlose als über drahtgebundene Endsysteme benutzen werden. Abbildung D.2 zeigt die prognostizierte weltweite Entwicklung der Internet–Nutzer in den nächsten Jahren. Dabei wird der Anteil der drahtlosen und drahtgebundenen Endsysteme mit aufgezeigt.

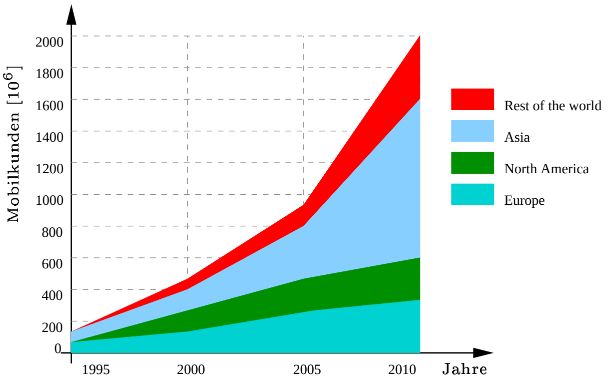


Figure D.1: Weltweite Entwicklung der drahtlosen Mobilkunden in Regionen [1].

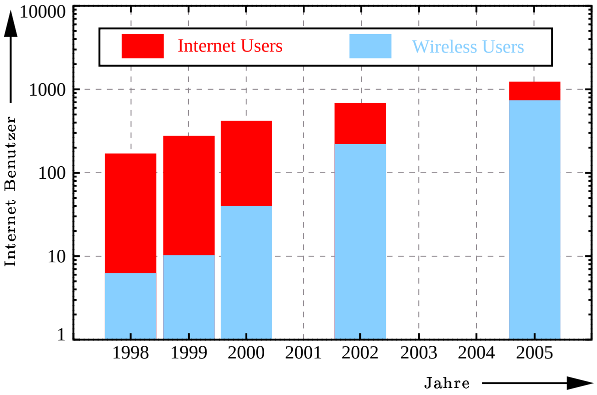


Figure D.2: Weltweite Entwicklung der Internet–Nutzer [2].

Die erste und zweite Mobilfunkgeneration war durch das Bedürfnis des Kunden nach Sprachdiensten geprägt. Zukünftige Mobilfunkgenerationen werden darüber hinaus auch Multimedia–Dienste über das **I**nternet **P**rotocol (*IP*) anbieten. Multimedia–Dienste umfassen sowohl Sprach– als auch Video–, Daten– und Gaming–Dienste. Multimedia–Dienste ohne die Sprachdienste sollen im Jahr 2010 nach einer Studie [8] einen Anteil von 60% ausmachen. Im Gegensatz zu reinen Sprachdiensten werden Multimedia–Dienste sich durch heterogene **Q**uality **o**f **S**ervice (*QoS*) Anforderungen in Hinblick auf Datenrate, verträgliche Verzögerung und verträgliche Verlustwahrscheinlichkeit auszeichnen [6]. Das Zusammenführen von Multimedia–Diensten und dem Mobilfunk stellt erhöhte Ansprüche an die Kommunikationsprotokolle. Gerade durch die Eigenschaften des drahtlosen Kanals ensteht die Schwierigkeit, dem Mobilkunden die vom Festnetz bekannten Leistungsmerkmale auch auf dem drahtlosen Endgerät bereitzustellen.

Der drahtlose Kanal ist im Gegensatz zu dem drahtgebundenen Kanal fehlerbehafteter. Um diese Fehler zu korrigieren gibt es Fehlerschutzmechanismen wie **A**utomatic **R**epeat Re**Q**uest (*ARQ*) und **F**orward **E**rror **C**orrection (*FEC*) in der Datensicherungsschicht. Im Falle von FEC wird dem eigentlichen Datenwort Redundanz hinzugefügt, mit der der Empfänger das richtige Datenwort dekodieren kann, selbst dann wenn Fehler auf dem drahtlosen Kanal entstanden sind. Je mehr Redundanz hinzugefügt wird desto fehlerunanfälliger wird die Übertragung. Gleichzeitig sinkt aber auch die Netto–Datenrate und damit die Verzögerung mit mehr Redundanz. Der Vorteil von FEC ist ein stabiler Durchsatz mit stabilen und daher voraussehbaren Verzögerungen. Der Nachteil liegt in der erhöhten Redundanz bei gutem Kanalzustand, da wo die Redundanz möglicherweise nicht gebraucht würde. In solchen Situationen würde Bandbreite verschwendet werden. ARQ Verfahren wiederholen fehlerhafte Datenworte, während weitere Datenworte in der Warteschlange zurückgehalten werden. Dieses Verfahren ist in Abbildung D.3 dargestellt. Somit umgehen die ARQ–Verfahren den Nachteil von FEC Verfahren, indem sie nur fehlerhafte Datenworte behandeln. Andererseits wird dadurch der Durchsatz und die Verzögerung variabel. Beide Verfahren ARQ und FEC sind zur Unterstützung von Multimedia–Diensten aufgrund des verringerten Durchsatzes oder der variable Verzögerung nicht optimal. Deshalb wird in dieser Arbeit ein Verfahren ausgearbeitet, daß die Nachteile der beiden oben benannten Verfahren nicht aufweist.

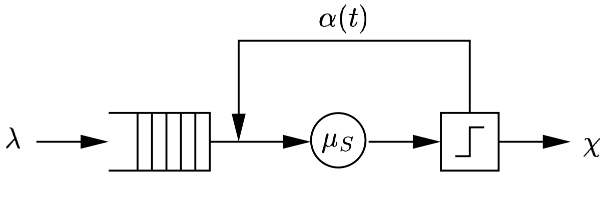


Figure D.3: Ein Server mit der Rate μ_S und der Ankunftsrate λ.

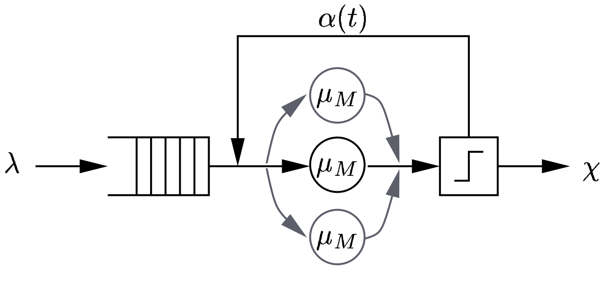


Figure D.4: Mehrere Server mit der rate μ_M und der Ankunftsrate λ.

Die Idee für den in dieser Arbeit vorgestellten Ansatz, im folgenden **S**imultaneous **MAC** **P**acket **T**ransmission (*SMPT*) genannt, basiert auf der Fähigkeit von drahtlosen Endgeräten auf mehreren **C**ode **D**ivision **M**ultiple **A**ccess (CDMA) Kanälen gleichzeitig zu senden und zu empfangen. Eine Möglichkeit dies zu erreichen basiert auf der **Multi–C**ode CDMA (*MC–CDMA*) Technologie. SMPT versendet fehlerhafte Datenworte nicht sequentiell, wie ARQ in der Zeitdomäne, sondern benutzt parallele Kanäle, um die Wiederholungen zu übertragen. Dies wird in Abbildung D.4 wiedergegeben. Dadurch werden Daten, die in der Warteschlange sind, nicht durch den ARQ–eigenen Wiederholungsprozeß beeinflußt, solange ausreichend parallele Kanäle zur Verfügung stehen. Im Gegensatz zu den in [19, 20] vorgestellten Verfahren, die ebenfalls auf MC–CDMA aufbauen, basiert der hier vorgestellte Ansatz auf der autonomen Verwaltung der Kanäle durch das drahtlose Endgerät und nicht durch eine zentrale Instanz. Dies verringert den sonst nötigen Signalisierungsoverhead und macht darüberhinaus diesen Ansatz für zellulare als auch Ad–hoc Netzwerke einsetzbar.

Die Untersuchungen dieser Arbeit beschränken sich auf den Uplink unter Verwendung von **P**seudo–**N**oise (*PN*) Sequenzen. Die Verwendung von PN Sequenzen hat den Vorteil, daß die Kanäle nicht nach Absprache mit allen anderen an der Kommunikation beteiligten Endgeräten zugewiesen werden müssen. Die verwendete PN Sequenz muß bei dem Sender und dem Empfänger bekannt sein, dadurch wird der Signalisierungsaufwand gering gehalten. Auf der anderen Seite steigt in einem CDMA System mit PN Sequenzen die Fehlerwahrscheinlichkeit mit jedem aktiv benutzten Kanal. Daher muß immer abgewogen werden, ob die Verwendung eines weiteren parallelen Kanals zu schlechteren Bedingungen auf dem Kanal führt. Dies bedeutet, daß die in Abbildung D.4 dargestellten Server mit einem jeweils verringertem μ_S arbeiten, sobald mehrere Kanäle benutzt werden. Dies gilt darüberhinaus für alle aktiven Kanäle in einer Zelle. Würde jedes Endgerät für sich die Kanäle zuschalten ohne die Existenz anderer Endgeräte zu berücksichtigen, könnte dies zur Instabilität des Mobilfunksystems führen. Es wurden vier verschiedene SMPT Verfahren entwickelt, die dieser Eigenschaft Rechnung tragen. Die Verfahren unterscheiden sich in der Art und Weise wie die parallelen Kanäle eingesetzt werden.

Zur Leistungsbewertung der verschiedenen SMPT Ansätze wurde ein Simulationsmodell entworfen. Als Modell für den drahtlosen Kanal wurden eine Markov Kette mit zwei Zuständen verwendet [162, 28, 163, 164, 165]. Die zwei Zustände sollen einen *guten* und einen *schlechten* Kanalzustand wiedergeben. Es wird davon ausgegangen, daß im *schlechten* Kanalzustand keine Kommunikation möglich ist. Im *guten* Kanalzustand hängt die Fehlerwahrscheinlichkeit von der Anzahl der aktiv belegten CDMA Kanäle ab. Zur Modellierung der Fehlerwahrscheinlichkeit wurde auf die *Improved Gaussian Approximation* von Holtzmann [76] zurückgegriffen. Darüberhinaus wurden verschiedene Lastmodelle eingesetzt. Die Lastmodelle basieren hauptsächlich auf Tracedateien von verschiedenen Videoströmen. Dazu wurden 25 Filmsequenzen von bis zu 60 Minuten mit H.263 und MPEG–4 in unterschiedlichen Qualitätsstufen kodiert. Die Bildgröße wurde auf QCIF eingestellt, damit kleinere Bitraten erzeugt werden, die sich wiederum leichter über den drahtlosen Kanal übertragen lassen. Die kodierten Videoströme wurden dann mit Hilfe eines Parsers in Tracedateien umgewandelt. Die Tracedateien werden anderen Wissenschaftlern über einen öffentlichen Web–Server zur Verfügung gestellt [135]. Je nach Applikationsart (Real–Time oder Streaming) wurde die entsprechende Transportschicht gewählt (respektive **U**ser **D**atagram **P**rotocol (*UDP*) oder **T**ransport **C**ontrol **P**rotocol (*TCP*)).

Bei der Untersuchung von Video–Streamingapplikationen über TCP konnte für eine vorgegebene Systemkonfiguration eine Kapazitätserhöhung durch den Einsatz eines bestimmten SMPT Mechanismuses erreicht werden. Es konnte darüber hinaus gezeigt werden, daß nicht ratenadaptive Videoströme nicht geeignet sind, um von TCP unterstützt zu werden. Bei Echtzeit–Videoapplikationen auf der Basis von UDP konnte gezeigt werden, daß sowohl ratenadaptierte, als auch nicht ratenadaptierte Videoströme durch den Einsatz von SMPT unterstützt werden können. Durch die SMPT Verfahren kann nicht nur die Videoqualität des einzelnen Mobilkunden verbessert werden, sondern bei bestimmten Systemkonfigurationen auch eine Kapazitätserweiterung erreicht werden. Im Verlauf der Untersuchungen wurde herausgestellt, daß einige SMPT Verfahren geringe

Interferenzen auf die Nachbarzellen und die eigene Zelle aufweisen. Es konnte gezeigt werden, in wieweit bestimmte SMPT Mechanismen Einfluß auf die Leistungsregelung des CDMA Empfängers nehmen. Ein wesentliches Ergebnis ist, daß die SMPT Verfahren ihre volle Leistungsfähigkeit bereits mit drei parallelen Kanälen erreichen. Existierenden Mobilfunksysteme wie IS–95 **I**nterim **S**tandard 95 (*IS95*) [9] sehen bereits acht Kanäle pro Endgerät vor.

Die in dieser Arbeit gemachten simulativen Untersuchungen beschränken sich auf zellulare Systeme. Durch den Einsatz der **P**seudo–**N**oise (*PN*) Sequenzen könnten die SMPT Mechanismen aber auch für Ad–hoc Systeme zum Einsatz kommen.

Index

Bibliography

[1] L. Roberts and M. Tarsala. Inktomi goes wireless; forms alliances. In *CBS Market-Watch*, March 14th 2000.

[2] eTForecasts. Wireless Internet Users. http://www.eTForecasts.com/, feb 2001.

[3] 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*, pages 32–1–32–5, Hamburg, Germany, September 2001.

[4] V. Kravcenko and F. Fitzek. CDMA Sequences for SMPT. Talk at TransiNet workshop, Oktober 2001.

[5] Commission of the European Communities. Digital Land Mobile Radio Communications - COST 207. Technical report, Office for Official Publications of the European Communities, Luxembourg, 1989. Final Report.

[6] Frank H.P. Fitzek, A. Köpsel, Adam Wolisz, M. Reisslein, and M. A. Krishnam. Providing Application–Level QoS in 3G/4G Wireless Systems: A Comprehensive Framework Based on Multi–Rate CDMA. In *IEEE International Conference on Third Generation Wireless Communications*, pages 344–349, June 2001.

[7] F. Fitzek and M. Reisslein. A Prefetching Protocol for Continuous Media Streaming in Wireless Environments. *JSAC – Mobility and Resource Management in Next Generation Wireless Systems*, 19(6):2015–2028, October 2001.

[8] UMTS Forum. The Future Mobile Market. Technical Report 8, UMTS Forum, mar 1999.

[9] *Mobile station–base station compatibility standard for dual–mode wideband spread spectrum cellular systems*, 1997.

[10] Dorgham Sisalem. End-To-End Quality of Service Control Using Adaptive Applications. In *IFIP Fifth International Workshop on Quality of Service IWQOS '97*. IFIP WG 6.1, May 1997. USA.

[11] Campbell, Hutchinson, and Aurrecoechea. Dynamic QoS management for scalable video flows. *Lecture Notes in Computer Science*, pages 107–118, 1995. Springer.

[12] 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.

[13] A. Ortega and M. Khansari. Rate control for video coding over variable bit rate channels with applications to wireless transmission. In *2nd IEEE Conference on Image Processing*, 1995. Washington, DC.

[14] Theodore S. Rappaport. *Wireless Communications*, volume 1. Prentice Hall, 1996. jan.

[15] John G. Proakis. *Digital Communications.* McGraw-Hill International Edition, third edition edition, 1995. USA.

[16] M. G. Jansen and R. Prasad. Capacity, Throughput, and Delay Analysis of a Cellular DS CDMA System With Imperfect Power Control and Imperfect Sectorization. *IEEE TRANSACTION ON VEHICULAR TECHNOLOGY*, 44(1):67–75, feb 1995.

[17] David J. Goodman. Addison–wesley wireless communication series. In *Wireless Personal Communication Systems*, 2. Addison–Wesley, 1997.

[18] Jerry D. Gibson. *Mobile Communications - Handbook*, volume 2. IEEE Press, 1996.

[19] I. Chih-Lin and S Nanda. Load and Interference Based Demand Assignment (LIDA) for Integrated Services in CDMA Wireless Systems. Technical report, Lucent, 1997.

[20] R. P. Ejzak, D.N. Knisely, S.Kumar, S. Laha, and S. Nanda. BALI: A Solution for High–Speed CDMA Data. Technical Report 3, Bell Labs Tech, 1997. Summer 97.

[21] F. Fitzek, B. Rathke, M. Schläger, and A. Wolisz. Quality of Service Support for Real-Time Multimedia Applications over Wireless Links using the Simultaneous MAC-Packet Transmission (SMPT) in a CDMA Environment. In *Proc. MoMuC 1998*, pages 367–378, October 1998.

[22] Frank H.P. Fitzek and Adam Wolisz. QoS Support In Wireless Networks Using Simultanoues MAC Packet Transmission (SMPT). In *Applied Telecommunication Symposium (ATS 99)*, volume 31, pages 185–190. SCS, April 1999. ISBN2-912328-07-1.

[23] F. Fitzek, R. Morich, and A. Wolisz. Comparison of Multi-Code Link-Layer Transmission Strategies in 3Gwireless CDMA. *IEEE Communication Magazine*, 38(10):58–64, oct 2000. Technologies on Broadband Wireless Mobile: 3Gwireless and Beyond.

[24] 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*, 39(5):794–807, May 1991.

[25] M. Aldinger. Die Simulation des MobilfunkKanals auf einem Digitalrechner. In *FREQUENZ*, volume 36, pages 145–152, 1982.

[26] L. Kittel. Analoge und diskrete Kanalmodelle für die Signalübertragung beim beweglichen Funk. In *FREQUENZ*, volume 36, pages 152–160, 1982.

[27] Matthias Pätzold. *Mobilfunkkanäle – Modellierung, Analyse und Simulation.* vieweg Verlag, 1999. ISBN 3-528-03892-6.

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

[29] H.P. Kuchenbecker. Statistische Eigenschaften von Fadingprozessen zur Beschreibung eines Landmobilfunkkanals. In *FREQUENZ*, volume 36, pages 138–144, 1982.

[30] Free bsd. http://www.freeBSD.org/.

[31] V. Jacobson. Congestion avoidance and control. In *Proceedings of SIGCOMM '88 Symposium*, pages 314–329, August 1988.

[32] Information Sciences Institute University of Southern California. Transmission Control Protocol. RFC 793, September 1981.

[33] M.E. Crovella and A. Bestavros. Self–Similarity in the World–Wide Web: Evidence and Possible Causes. *IEEE/ACM Transactions on Networking*, 5(5):835–846, December 1997.

[34] V. Paxson and S. Floyd. Wide Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

[35] G. Bao. Performance evaluation of TCP/RLP protocol stack over CDMA wireless link. *Wireless Networks*, pages 229–237, 1996.

[36] B.Rathke, M.Schläger, and A.Wolisz. Systematic Measurement of TCP Performance over Wireless LANs. Technical report, TKN, December 1998. TKN01BR98.

[37] A. DeSimone, M. C. Chuah, and O. Yue. Throughput Performancce of Transport-Layer Protocols over Wireless LANs. In *Proceedings IEEE GLOBECOM 93*, 1993.

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

[39] H. Balakrishnan, S. Seshan, and R.H. Katz E. Amir. Improving TCP/IP Performance over Wireless Networks. In *Proceedings of Mobicom*, November 1995.

[40] M. Zorzi and R. R. Rao. The effect of correlated errors on the performance of TCP. *IEEE Communications Letters*, 1(September):127–129, 1997.

[41] Henning Sanneck. *Packet Loss Recovery and Control for Voice Transmission over the Internet.* PhD thesis, GMD Focus, 2000.

[42] W. Stevens. *TCP/IP Illustrated.* Addison-Wesley, 1994.

[43] Thomas Jüngling. Webgames für zwischendurch. Welt am Sonntag, July 2001. No. 30.

[44] M. S. Borella. Source Models of Network Game Traffic. *Computer Communications*, 23(4):403–410, feb 2000.

[45] J. Munkki. Design and Implementation of Networked Games. http://www.hut.fi/~jmunkki/netgames, December 1997.

[46] F. Fitzek and M. Reisslein. MPEG–4 and H.263 traces for network performance evaluation (extended version). Technical Report TKN–00–06, Technical University Berlin, Dept. of Electrical Eng., Germany, October 2000. Traces available at `http://www-tkn.ee.tu-berlin.de/research/trace/trace.html` and `http://www.eas.asu.edu/trace`.

[47] P. T. Brady. Effects of Transmission Delay on Conversational Behavior on Echo-Free Telephone Circuits. *Bell System Technical Journal*, 50:115–134, 1971.

[48] Ralf Schäfer. Video encoding for mobile communication. TUB Colloquium, May 2001.

[49] K. N. Ngan, C. W. Yap, and K. T. Tan. *Video Coding for Wireless Communication Systems.* Signal Processing and communication Series. Marcel Dekker, 2001.

[50] G. R. Cooper and C. D. McGillem. *Modern Communications and Spread Spectrum.* Electrical and Electronic Engineering Series. McGraw–Hill Series in Electrical Engineering, 1986.

[51] Ramjee Prasad. *CDMA for wireless personal communications.* Mobile Communications Series. Artech House Publishers, 1996.

[52] Tero Ojanperae and Ramjee Prasad. *Wideband for third generation mobile communications.* Number 0-89006-735-X. ArTech House Publisher, universal personal communications edition, 1998.

[53] Andreas Ploydoros. An Integrated Physical/Link–Access Layer Model of Packet Radio Architectures. Technical report, Communication Sciences Institute – University of Southern California, 1994.

[54] Andre Noll Barretto and Gerhard Fettweis. Performance Improvement in DS–Spread Spectrum CDMA Systems using a PRE- and a POST–RAKE. In *IZS2000*, pages 39–46. ETH Zurich, Febr. 2000.

[55] Dr. J. Shapira. *Advanced CDMA Workshop.* Wireless Institute of Technology, San Jose, Carlifornia, July 1999.

[56] M. G. Jansen, R. Prasad, and A. Kegel. Capacity Analysis of a Cellular Direct Sequence Code Division Multiple Access System with Imperfect Power Control. *IEEE Trans. Communication*, E67-B(8):894–905, 1993.

[57] N. Bambos and S. Kandukuri. Power Controlled Multiple Access (PCMA) in Wireless Communication Networks. In *Proc. INFOCOM 2000*, pages 386–395, 2000.

[58] S. W. Golomb. *Shift Register Sequences.* Aegean Park Press, 1992.

[59] W. Stahnke. Primitive binary polynomials. *Math. Comp.*, 27:977–980, oct 1973.

[60] R. Gold. Optimal binary sequences for spread spectrum multiplexing. *IEEE Trans. Info. Theory*, IT–B:619–621, oct 1967.

[61] T. Kasami. Weight distribution formula for some class of cyclic codes. Technical Report R-285, Coordinated Science Lab., Univ. IL, Urbana, 1996.

[62] F. Fitzek and M. Reisslein. A Prefetching Protocol for Streaming of Prerecorded Continuous Media in Wireless Environments. In *SPIE 2001*, pages 121–129, Aug 2001.

[63] L. Yun and D. Messerschmitt. Power control for variable QoS on a CDMA channel. In *Proc. IEEE Milcom Conf.*, pages 178–182, october 1994.

[64] J. Wu and R. Kohno. A wireless multimedia CDMA system based on transmission power control. *IEEE J. Select. Areas Communication*, 14(4):683–691, May 1996.

[65] H. Morikawa, T. Kajiya, T. Aoyama, and A. Campbell. Distributed Power Control for Various QoS in a CDMA Wireless System. *IEICE Trans. Fundamentals*, E80-A(12):2429–2436, December 1997.

[66] L. Yun and D. Messerschmitt. Variable quality of service in CDMA systems by statistical power control. In *Proc. IEEE International Conf. Commun (ICC'95)*, pages 713–719, June 1995.

[67] P. Immaneni and J. M. Capone. A framework for analysis of VBR traffic over CDMA. In *Proceedings of 2000 IEEE Wireless Communications and Networking Conference*, Chicago, IL, September 2000.

[68] C.-L. I and K. K. Sabnani. Variable spreading gain CDMA with adaptive control for true packet switching wireless networks. In *Proc. ICC'95*, pages 725–730, June 1995.

[69] I. Chih-Lin and R. D. Gitlin. Multi-code (cdma) wireless personal communications networks. In *IEEE International Conference on Communications '95*, pages 1060–1064, Seattle, Washington, June 1995.

[70] Slawomir Stanczak, Holger Boche, Frank H.P. Fitzek, and Adam Wolisz. Design of spreading sequences for SMPT–based CDMA systems. In *Proc. 34th ASILOMAR Conference on Signals Systems and Computers*, pages 1622–1626, October 2000.

[71] S. Ramakrishna and J. Holtzman. A Comparison between Single Code and Multiple Code Transmission Schemes in a CDMA System. In *Proc. IEEE Vehicular Technology Conference – VTC 98*, pages 791–795, 1998.

[72] C. Webb, H. Huang, and S. Brink. Rake Receiver Architectures for Multi–Code CDMA. In *Proceedings of the Sixth WINLAB Workshop on Third Generation Wireless Information Networks*, pages 229–238, March 1997.

[73] UMTS 30.03. Universal Mobile Telecommunications System (UMTS); Selection Procedures for the Choice of Radio Transmission Technologies of the UMTS.

[74] Prof. Peter Noll. *Nachrichten Übertragung II*, volume 2. FT, 1999. http://www-ft.ee.tu-berlin.de.

[75] M. B. Pursley. Performance evaluation for phase–coded spread spectrum multiple–access communication – Part I:System analysis. *Transaction on Communications*, COM–25(8):795–799, 1977.

[76] J. Holtzman. A Simple, Accurate Method to Calculate Spread Spectrum Multiple Access Error Probabilities. *IEEE Trans. Commun.*, 40(3):461–464, March 1992.

[77] R. Buehrer and B. Woerner. Analysis of adaptive multistage interference cancellation for CDMA using an improved Gaussian approximation. *IEEE Trans. Commun.*, 44:1308- 1321, Oct 1996.

[78] James L. Massey and Thomas Mittelholzer. Welch's Bound and Sequence Sets for Code Division Multiple Access Systems. *Signal and Information Processing Laboratory*, pages 63–78, 1900.

[79] X. Gu and Olafsson S. A simplified and accurate method to analyse a code division multiple-access performance. In *UCL and IEE London Communications Symposium*, pages 31–36, 2000.

[80] P.-R. Chang and C.-F. Lin. Wireless ATM-Based Multicode CDMA Transport Architecture for MPEG-2 Video Transmission. *SPECIAL ISSUE ON VIDEO TRANSMISSION FOR MOBILE MULTIMEDIA APPLICATIONS*, 87(10), October 1999.

[81] Moncef Elaoud and Parameswaran Ramanathan. Adaptive Allocation of CDMA Resources for Network–level QoS Assurances. *MobiCom 2000*, 2000.

[82] G. J. Foschini and Z. Miljanic. A Simple Distributed Autonomous Power Control Algorithm and its Covergence. *IEEE Trans. on Veh.*, 42(4):641–646, 1993.

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

[84] H. Liu, H.Ma, M. E. Zarki, and S. Gupta. Error control schemes for networks: An overview. *Mobile Networks and Applications*, (2):167–182, 1997.

[85] J. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive FEC-based error control for Internet Telephony. In *Proc. of Infocom'99*, March 1999.

[86] S. Shenker. Fundamental design issues for the future internet. *IEEE J. on Selected Areas in Communications*, 13(7):1176–1188, September 1995.

[87] J. Bito, H. Schulz, and P. Noll. Simulation Study of Integrated Video, Voice and Data Transmission in Common-Code DS/CDMA Systems. *Mobile Multimedia Communications (Ed.: D. Goodman and D. Raychaudhuri)*, pages 223–230, 1997. Plenum Press.

[88] Jungbauer R., Herzog R., Schmidbauer A., Hagenauer J., and Riedel S. Coding for a CDMA-system with higher user data rates by combining several traffic channels. *Proc. of the European Personal Mobile Communications Conference (EPMCC)*, pages 523–529, September 1997.

[89] K. Sklower, B. Lloyd, G. McGregor, and D. Carr. The PPP Multilink Protocol (MP). RFC 1717, nov 1994.

[90] K. Sklower, B. Lloyd, D. Carr, and T. Coradetti. The PPP Multilink Protocol (MP). RFC 1990, aug 1996.

[91] W. Simpson. The Point–to–Point Protocol (PPP). RFC 1661, jul 1994.

[92] W. Simpson. PPP over ISDN. RFC 1618, may 1994.

[93] C. Richards and K. Smith. The PPP Bandwidth Allocation Protocol (BAP) The PPP Bandwidth Allocation Control Protocol (BACP) . RFC 2125, mar 1997.

[94] Zhao Liu. *Medium Access Control Schemes for DS–CDMA Wireless Packet Networks.* Ph.d. dissertation, University of Pennsylvania, December 1995.

[95] Z. Liu, M. Karol, and K. Eng. An efficient demand–aasignment multiple access protocol for wireless packet (ATM) networks. *Wireless Networks 1*, 1:267–279, 1995.

[96] Z. Liu, M. Karol, M. El Zarki, and K. Eng. Distributed–Queueing Request Update Multiple Access (DQRUMA) for Wireless Packet (ATM) Networks. *ICC'95*, (1224-1231), 1995. Seattle.

[97] Z. Liu, M. Karol, M. El Zarki, and K. Eng. Channel access and interference issues im multi-code DS-CDMA wireless packet (ATM) networks. *Wirless Network 2*, pages 173–193, 1996.

[98] Z. Liu, M. Karol, M. El Zarki, and K. Eng. A Demand–Assignment Access Control For Multi–Code DS–CDMA Wireless Packet (ATM) Networks. *xx*, xx.

[99] Jae-Yoon Park and Kuk-Kyung Kim. An Enhanced DQRUMA/MC-CMDA Protocol for Wireless Packet Networks. *IEICE TRANS. COMMUNICATION*, E83-B(7):1567–1571, July 2000.

[100] C. Lin and R. D. Gitlin. Multi–Code CDMA wireless personal communications networks. *ICC'95*, pages 1060–1064, 1995. Seattle.

[101] Mobile Station-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular Systems, 1995.

[102] Mobile Station-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular Systems, 1997.

[103] F. Fitzek and M. Reisslein. A prefetching protocol for continuous media streaming in wireless environments (extended version. Technical Report TKN–00–05, Technical University Berlin, Dept. of Electrical Eng., Germany, July 2000.

[104] S. Nanda, K. Balachandran, and S. Kumar. Adaptation techniques in wireless packet data services. *IEEE Communications Magazine*, 38(1):54–64, Jan 2000.

[105] M. Zorzi and R.R. Rao. Error Control and Energy Consumption in Communications for Nomadic Computing. *IEEE Transaction on Information Theory*, 46:279–289, Mar 1997.

[106] C. Perkins, O. Hodson, and V. Hardman. A survey of packet-loss recovery techniques for streaming audio. *IEEE Network Magazine*, Sept./Oct. 1998.

[107] D. Ferrari. Client requirements for real–time communication services. RFC 1193, November 1990.

[108] G. Bertsekas. *Data Networks*, volume 2. Prentice Hall, 1992.

[109] IEEE Stds. Dept. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Jan. 1996.

[110] *Radio Equipment and Systems (RES); High Performance Radio Local Area Network (HIPERLAN); Type 1*, Dec. 1996. France.

[111] R.F. Ormondroyd. Performance of Low-Rate Orthogonal Convolutional Codes in DS-CDMA Applications. *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, 46(2), May 1997.

[112] Frank H.P. Fitzek, George Carle, and Adam Wolisz. Combining Transport Layer and Link Layer Mechanism for Transparent QoS Support of IP based Applications. In *IP Quality of Service for Wireless and Mobile Networks (IQWiM99)*. IQWiM99, April 1999.

[113] V. K. Gargand K. Smolik. *Applications of CDMA in Wireless/Personal Communications.* Prentice-Hall, 1997.

[114] A. J. Viterbi. *CDMA – Principles of Spread Spectrum Communication*, volume 5. Addison-Wesly Wireless Communications Series, 1995.

[115] Personal Station-Base Station Compatibility Requirements for 1.8 to 2.0 GHz Code Division Multiple Access (CDMA) Personal Communications Systems, 1995.

[116] 3GPP. Physical channels and mapping of transport channelsonto physical channels (TDD). Technical report, 3rd Generation Partnership Project, 1999.

[117] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. Long–range dependence in variable–bit–rate video traffic. *IEEE Transactions on Communications*, 43(2/3/4):1566–1579, February/March/April 1995.

[118] M. Frey and S. Nguyen-Quang. A gamma–based framework for modeling variable–rate MPEG video sources: The GOP GEAR model. *IEEE/ACM Transactions on Networking*, 8(6):710–719, December 2000.

[119] J.R. Gallardo, D. Makrakis, and L. Orozco-Barbosa. Use of alpha–stable self-similar stochastic processes for modeling traffic in broadband networks. *Performance Evaluation*, 40(1–3):71–98, March 2000.

[120] M. Degermark, T. Kohler, S. Pink, and O. Schelen. Advance reservations for predictive service in the internet. *Multimedia Systems*, 5(3):177–186, 1997.

[121] N.G. Duffield, K.K. Ramakrishnan, and A.R. Reibman. SAVE: An algorithm for smoothed adaptive video over explicit rate networks. *IEEE/ACM Transactions on Networking*, 6(6):717–728, December 1998.

[122] I. Dalgic and F.A. Tobagi. Performance evaluation of ATM networks carrying constant and variable bit rate video traffic. *IEEE Journal of Selected Areas in Communications*, 15(6):1115–1131, 1997.

[123] M. Grossglauser, S. Keshav, and D. Tse. RCBR: A simple and efficient service for multiple time–scale traffic. *IEEE/ACM Transactions on Networking*, 5(6):741–755, 1997.

[124] P. Jelenkovic, A. Lazar, and N. Semret. The effect of multiple time scales and subexponentiality in MPEG video streams on queueing behavior. *IEEE Journal of Selected Areas in Communications*, 15(6):1052–1071, 1997.

[125] J.D. Salehi, Z.L. Zhang, J. Kurose, and D. Towsley. Supporting stored video: Reducing rate variability and end–to–end resource requirements through optimal smoothing. *IEEE/ACM Transactions on Networking*, 6(4):397–410, August 1998.

[126] D.E. Wrege, E.W. Knightly, H. Zhang, and J. Liebeherr. Deterministic delay bounds for VBR video in packet–switching networks: Fundamental limits and practical trade–offs. *IEEE/ACM Transactions on Networking*, 4(3):352–362, June 1996.

[127] A. Baiocchi, F. Cuomo, and S. Bolognesi. IP QoS delivery in a broadband wireless local loop: MAC protocol definition and performance evaluation. *IEEE Journal on Selected Areas in Communications*, 18(9):1608–1622, September 2000.

[128] M. Ma and M. Hamdi. Providing deterministic quality–of–service guarantees on WDM optical networks. *IEEE Journal on Selected Areas in Communications*, 18(10):2072–2083, October 2000.

[129] M. W. Garret. *Contributions toward Real-Time Services on Packet Networks.* PhD thesis, Columbia University, May 1993. ftp address and directory of the used video trace:bellcore.com/pub/vbr.video.trace/.

[130] O. Rose. Statistical properties of MPEG video traffic and their impact on traffic modelling in ATM systems. Technical Report 101, University of Wuerzburg, Institute of Computer Science, Wuerzburg, Germany, February 1995.

[131] M. Krunz, R. Sass, and H. Hughes. Statistical characteristics and multiplexing of MPEG streams. In *Proceedings of IEEE Infocom '95*, pages 455–462, April 1995.

[132] W-C. Feng. *Video–on–Demand Services: Efficient Transportation and Decompression of Variable Bit Rate Video.* PhD thesis, University of Michigan, April 1996. `http://www.cis.ohi-state.edu/~wuchi/`.

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

[134] F. Fitzek and M. Reisslein. MPEG–4 and H.263 Video Traces for Network Performance Evaluation. *IEEE Network*, 15(6):40–54, November/December 2001. preprint and video traces available at `http://www-tkn.ee.tu-berlin.de/research/trace/trace.html` or `http://www.eas.asu.edu/trace`.

[135] F. Fitzek and M. Reisslein. MPEG–4 and H.263 Traces for Network Performance Evaluation. `http://www-tkn.ee.tu-berlin.de/research/trace/trace.html` and `http://www.eas.asu.edu/trace`, Januar 2000.

[136] A. Puri and T. Chen. *Multimedia Systems, Standards, and Networks.* Marcel Dekker, New York, 2000.

[137] R. Koenen (Editor). Overview of the MPEG–4 standard, ISO/IEC 14496, May/June 2000.

[138] R. Koenen. MPEG–4 multimedia for our time. *IEEE Spectrum*, 36(2):26–33, February 1999.

[139] L. D. Soares and F. Pereira. MPEG–4: A fexible coding standard for the emerging mobile multimedia applications. Technical report, MOMUSYS Project, 1999.

[140] F. Halsall. *Multimedia Communications: Applications, Networks, Protocols, and Standards.* Addison–Wesley, 2001.

[141] J. M. Boyce and R. D. Galginello. Packet loss effects on MPEG video sent over the public Internet. In *ACM Multimedia 98.* ACM, Sept. 1998.

[142] ITU-T/SG15. Recommendation H.263, video coding for low bitrate communication, 1996.

[143] K. Rijkse. H.263: Video coding for low–bit–rate communication. *IEEE Communications Magazine*, 34(12):42–45, December 1996.

[144] ITU-T. Recommendation H.261, video codec for audio–visual services at 64 – 1920 kbit/s, 1993.

[145] J. Walter. bttvgrab. `http://www.garni.ch/bttvgrab/`.

[146] G. Heising and M. Wollborn. MPEG–4 version 2 video reference software package, ACTS AC098 mobile multimedia systems (MOMUSYS), December 1999.

[147] K. O. Lillevold. tmn, version 2.0, June 1996. available at `http://www.nta.no/brukere/DVC`.

[148] B. Ryu. Modeling and simulation of broadband satellite networks — Part II: Traffic modeling. *IEEE Communications Magazine*, 37(7):48–56, July 1999.

[149] S. McCanne and V. Jacobson. vic: A flexible framework for packet video. In *Proceedings of ACM Multimedia*, San Francisco, CA, April 1995.

[150] K. Dolzer and W. Payer. Two classes — sufficient QoS for multimedia traffic? In *Proceedings COST 257 Symposium*, Oslo, Norway, May 2000.

[151] A. Adas and A. Mukherjee. On resource management and QoS guarantees for long range dependent traffic. In *Proceedings of IEEE Infocom '95*, Boston, MA, April 1995.

[152] N. G. Duffield, J. T. Lewis, and N. O'Connell. Predicting quality of service for traffic with long–range fluctuations. In *Proceedings of IEEE ICC '95*, Seattle, WA, April 1995.

[153] N. Likhanov, B. Tsybakov, and N. D. Georganas. Analysis of an ATM buffer with self–similar (fractal) input traffic. In *Proceedings of IEEE Infocom '95*, Boston, MA, April 1995.

[154] M. Parulekar and A. M. Makowski. Buffer overflow probabilities for a multiplexer with self–similar input. In *Proceedings of IEEE Infocom '96*, San Francisco, CA, March 1996.

[155] B. Ryu and A. Elwalid. The importance of long–range dependence of VBR video traffic in ATM traffic engineering: myths and realities. In *Proceedings in ACM SIGCOMM*, pages 3–14, Stanford, CA, August 1996.

[156] J. Beran. *Statistics for long–memory processes.* Chapman and Hall, 1994.

[157] J. Roberts, U. Mocci, and J. Virtamo (Eds.). *Broadband Network Traffic: Performance Evaluation and Design of Broadband Multiservice Networks, Final Report of Action COST 242, (Lecture Notes in Computer Science, Vol. 1155).* Springer Verlag, 1996.

[158] Y. Wang and Q. Zhu. Error control and concealment for video communication: A review. *Proceedings of the IEEE*, 86(5):974–997, May 1998.

[159] Berkeley (USA) Regents of the University of California. Ptolemy. http://ptolemy.eecs.berkeley.edu.

[160] Network simulator. http://www.isi.edu.

[161] J. N. Daigle and J. D. Langford. Models for analysis of packet–voice commuication systems. *Journal on Selected Areas in Communications (JSAC)*, pages 847–855, sep 1986.

[162] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathi. Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs. *ACM Wireless Networks*, 3:91–102, 1997.

[163] R. R. Rao. Higher Layer Perspectives on Modeling the Wireless Channel. In *IEEE Information Theory Workshop*, 1998.

[164] A. Willig, M. Kubisch, and A. Wolisz. Measurements and Stochastic Modeling of a Wireless Link in an Industrial Environment. Technical report, Telecommunication Networks Group, March 2001.

[165] J. P. Ebert and A. Willig. A Gilbert-Elliot Bit Error Model and the Efficient Use in Packet Level Simulatio. Technical report, Telecommunication Networks Group, March 1999.

[166] B.D. Fritchman. *A binary channel characterization using partitioned Markov chains with applications to error correction codes.* PhD thesis, Lehig University, 1966.

[167] B.D. Fritchman. A binary channel characterization using partitioned Markov chains. *IEEE Trans. Inform. Theory*, IT-13:221–227, April 1967.

[168] E. N. Gilbert. Capacity of a burst–noise channel. *Bell Systems Technical Journal*, 39:1253–1266, September 1960.

[169] E. O. Elliot. Estimates of error rates for codes on burst–noise channels. *Bell Systems Technical Journal*, 42:1977–1997, September 1963.

[170] C. Chien, M. B. Srivastava, R. R. Jain, P. Letteri, V. Aggarwal, and R. Sternowski. Adaptive Radio for Multimedia Wireless Links. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 17(5), May 1999.

[171] N. Nikaein and C. Bonnet. On the Performance of FEC for Multicast Communication on a Fading Channel. In *Proceedings of International Conference on Telecommunications ICT*, Mexico, May 2000.

[172] H. S. Wang and N. Moayeri. Finite–state Markov channel — A useful model for radio communication channels. *IEEE Trans. Veh. Technol.*, 44:163–171, Feb 1995.

[173] Giao Thanh Nguyen, Randy H. Katz, Brian Noble, and M. Satyanarayanan. A Trace-Based Approach for Modeling Wireless Channel Behavior. In *Winter Simulation Conference*, pages 597–604, 1996.

[174] W. Kellerer, E. Steinbach, P. Eisert, and B. Girod. Testbed for Internet Media Streaming over Wireless Channels. Technical report, Image, Video, and Multimedia Systems Group - Information Systems Laboratory, December 2001.

[175] H. Woesner and A. Köpke. The ns-2/AKAROA-2 interface. Technical Report TKN-01-008, Telecommunication Networks Group, Technische Universität Berlin, July 2001.

[176] L. W. Schruben. Detecting initialization bias in simulation output. *Operations Research*, 30:569–590, 1982.

[177] K. Pawlikowski, V. Yau, and D. McNickle. Steady-state Simulation of Queueing Processes: A Survey of Problems and Solutions. *ACM Computing Surveys*, 22(123–170), 1990.

[178] M.S. Meketon and B. Schmeiser. Overlapping Batch Means: Something for Nothing ? In *Winter Simulation Conference*, pages 227–230, 1984.

[179] Frank H.P. Fitzek, Edjair Mota, Enno Ewers, Adam Wolisz, and K. Pawlikowski. An Efficient Approach For Speeding Up Simulation Of Wireless Networks. In *WMC 2000*. WMC, January 2000.

[180] M. Krishnam and F. Fitzek. Analytical Performance Evaluation for a Multi–Code CDMA based network. Technical report, Technical University Berlin, Dept. of Electrical Eng., Germany, oct 2001.

[181] K.S. Gilhousen, I.M. Jacobs, R. Padovani, A.J. Viterbi, L.A. Weaver Jr., and C.E. Wheatley III. On the capacity of a cellular CDMA system. *IEEE Trans. Vehic. Technol.*, 40:303–312, May 1991.

[182] Dr. Bose. M–comerce for 3G. talk, September 2001. 11^{th} Time-Market, Sony–Center Berlin.

[183] Real. Real player. http://www.real.com/.

[184] Microsoft. Microsoft windows media[tm] player 7.1. http://www.microsoft.com/.

[185] C. Oliveira D. Hong, C. Albuquerque and T. Suda. Evaluating the Impact of Emerging Streaming Media Applications on TCP/IP Performance. *IEEE Communications Magazine*, 39(4):76–82, April 2001.

[186] M. Hassan and R. Jain. TCP Performance in Future Networking Environments. *Guest Editorial, IEEE Communications Magazine*, 39(4):51, April 2001.

[187] C. Albuquerque, B. J. Vickers, and T. Suda. Network Border Patrol. In *IEEE INFOCOM 2000*, March 2000.

[188] S. Jacobs and A. Eleftheriadis. Streaming video using dynamic rate shaping and TCP congestion control. *Journal of Visual Communication and Image Representation*, 9(3):211–222, 1998.

[189] F. Fitzek, B. Rathke, M. Schläger, and A. Wolisz. Simultaneous MAC-Packet Transmission in Integrated Broadband Mobile System for TCP. In *ACTS SUMMIT 1998*, pages 580–586. ACTS, June 1998.

[190] A. S. Tosun and W.-C. Feng. On Improving Quality of Video for H.263 over Wireless CDMA Networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, September 2000.

[191] W. Kumwilaisak, J. W. Kim, and C.-C. J. Kuo. Reliable Wireless Video Transmission via Fading Channel Estimation and Adaptation. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, September 2000.

[192] D. Qiao and Kang G. Shin. A Two–Step Adaptive Error Recovery Scheme for Video Transmission over Wireless Networks. In *INFOCOM*, pages 1698–1704, Tel Aviv, Israel, March 2000.

[193] H. Liu and M. Zarki. Performance of H.263 video transmission over wireless channels using hybrid ARQ. *IEEE JSAC*, 15(9):1775–1786, December 1997.

[194] Frank H.P. Fitzek, Rolf Morich, and Adam Wolisz. SMPT: Comparison of Different Link-Layer Transmission Strategies in a CDMA Environment. In *IEEE International Conference on Third Generation Wireless Communications*, pages 490–497, June 2000.

[195] G. P. Fettweis, K. Iversen, M. Bronzel, H. Schubert, V. Aue, D. Mämpel, J. Voigt, A. Wolisz, G. Walf, and J.-P. Ebert. A closed solution for an integrated broadband mobile system (IBMS). In *Proceeding of the International Conference on Universal Personal Communications (ICUPC'96)*, pages 707–711, Cambridge, Massachusetts, October 1996.

[196] M. Bronzel, D. Hunold, G. Fettweis, T. Konschak, T. Dölle, V. Brankovic, H. Alikhani, J-P. Ebert, A. Festag, F. Fitzek, and A. Wolisz. Integrated Broadband Mobile System (IBMS) featuring Wireless ATM. In *ACTS Mobile Communication Summit 98*, Aalborg, Denmark, October 1998.

[197] Shyam S. Chakraborty, Erkki Yli-Juuti, and Markku Liinaharja. An ARQ Scheme with Packet Combining. *IEEE COMMUNICATIONS LETTERS*, 2, July 1998.

[198] S. Gupta and M. El Zarki. Optimal design of error control schemes for packet radio networks. In *IEEE International Conference on Personal and Wireless Communications*, 1994.

[199] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Mobile Computing and Networking*, pages 85–97, 1998.

[200] A. M. Law and W. D. Kelton. *Simulation, Modeling, and Analysis.* McGraw–Hill, second edition, 1991.

[201] B. B. Mandelbrot and M. S. Taqqu. Robust R/S analysis of long–run serial correlations. In *Proceedings of 42nd Session ISI, Vol. XLVIII, Book 2*, pages 69–99, 1979.

[202] Horst Stöcker. *Taschenbuch mathematischer Formeln und Verfahren.* Verlag Harri Deutsch, 1993.

[203] T. M. Lok and J. S. Lehnert. Error probabilities for generalized quadriphase DS/SSMA communication systems with random signature sequences. *Transaction on Communications*, 44(7):876–885, 1996.

[204] S. Kaiser. OFDM–CDMA versus DS–CDMA: Performance evaluation for fading channels. In *IEEE International Conference on Communications*, June 1995.