Technical University Berlin

Telecommunication Networks Group

Measurements and Stochastic Modeling of a Wireless Link in an Industrial Environment

A. Willig, M. Kubisch and A. Wolisz

{willig,kubisch,wolisz}@ft.ee.tu-berlin.de

Berlin, March 2001

TKN Technical Report TKN-01-001

TKN Technical Reports Series

Editor: Prof. Dr.-Ing. Adam Wolisz

Abstract

The design and simulation of coding schemes, medium access control and link layer protocols for future industrial wireless local area networks can be supported by some understanding of the statistic properties of the bit error patterns delivered by a wireless link (which is an ensemble of transmitter, channel, receiver, modems). We present results of bit error measurements taken with an IEEE 802.11-compliant physical layer hardware in an industrial environment. In addition to reporting the most important results, we draw some conclusions for the design of protocols and for the stochastic simulation of wireless channels. Besides discussing some well-known bit error models, we introduce the class of bipartite models.

Contents

Chapter 1

Introduction

There is currently an increasing interest in making wireless transmission technologies available not only in office and home environments, but also in industrial environments. Two especially attractive features of wireless technologies are the reduced need for cabling and the potential for truly mobile stations. However, due to the special constraints in industrial applications, e.g., hard real-time requirements, it is probably not the best solution to simply use existing wireless technologies, which often are designed for different purposes. Instead, we identify a strong need to design and develop specialized protocols, specifically for medium access control (MAC) and link layer protocols, which take both the characteristics of the wireless medium and the hard real-time requirements into account.

For design of MAC and link layer protocols, two reasons make it vital to have some understanding of the error patterns delivered by the physical layer (PHY). The first reason is that the same protocol can show different behavior and performance for different error characteristics. For example, in references [39] and [38] it is shown that bursty (Markovian) channel errors are beneficial for the performance of TCP, as compared to the case of independent errors with the same mean bit error rate. For the PROFIBUS (a popular fieldbus system in Europe) independent errors result in better delay performance and stability of the logical token passing ring than bursty errors [33]. The second reason is that advance knowledge of the error characteristics can help the protocol designer to select appropriate protocol mechanisms, e.g., to choose suitable forward error correction (FEC) schemes or to find good rules on when to perform retransmissions.

It is often convenient for the protocol designer to evaluate protocols with simulations before developing a prototype implementation and performing complex measurements. With simulation studies, a first assessment of the protocol performance can be done. A key part of such simulations are channel error models, which, in principle, determine for a transmitted packet which of the receiver stations see bit errors; often the exact position of errors within a packet is also of interest. In almost all cases simulation-based performance evaluations are done with *stochastic* error models.[1] Here, a simple stochastic process which often can be described in terms of a few parameters, is employed to generate bit error patters with some given characteristics. Most of these stochastic models are not designed to reflect or model physical phenomena, but simply to reproduce the statistics of given or imagined error patterns with some accuracy. Hence, their computational complexity is low as compared to, e.g., realistic channel models based on ray tracing. This simplicity is beneficial for packet level simulations, since often the model has to be applied to several thousands of packets and simulation time is becoming an issue. Some models are frequently used in performance studies of MAC or link layer protocols, e.g. the *Gilbert/Elliot model*, however, often the model parameters are not grounded on "real data".

Three important questions regarding stochastic bit error models are: How accurate are they? How are they parameterized? What is their computational complexity? One attractive but complex way to address the first two questions is to perform measurements and to use the measured data to parameterize different models and to compare the models statistics with the measured statistics. In this report we describe measurements of error characteristics made in an industrial environment, using an IEEE 802.11-compliant direct sequence spread spectrum (DSSS) PHY. More precisely, we are interested in the statistics of the packet loss and bit error patterns delivered by the *physical link*, which we use as an abstraction of the ensemble of transmitter, receiver, spread spectrum modem, the channel, the scrambler, high and intermediate frequency circuitry and more. Our interest is in what the physical link (or the *baseband processor*) delivers to the MAC and link layer protocol, and not in the low level processes. We present the most important results and discuss some important implications for the design of MAC and link layer protocols for future industrial wireless local area networks (WLAN). Furthermore, we discuss the meaning of our results for stochastic bit error models.

This report offers several achievements. First it is, to the best of our knowledge, the first published bit level and packet level wireless measurement study taken in an industrial environment. Second, our measurement setup is built in a way, that there is no bias by upper layer protocols or operating systems. In most other measurement studies the measurements are carried out with UDP packets over a WaveLAN wireless LAN (see Section 5.3). We have fine control over the timing and content of the generated packets. Third, our results indicate

[1] An alternative approach would be to use traces, but these are only rarely available and their handling is often perceived as clumsy.

that common approaches to stochastic modeling have several shortcomings and inaccuracies. One important point is that they focus solely on bit errors and do not consider losses or impairments of whole packets. Therefore, we propose a unified framework integrating both packet impairments and bit errors. Another point is that the distributions chosen for error burst lengths and error free burst lengths in these models do not match our findings, often the model distributions are much less variable. We propose a new model class, denoted as *bipartite models* designed to attack some of the shortcomings of existing models. However, even when for simplicity one of the well-known models is chosen for a simulation study, our measurements allow to determine model parameters which have "smelled the air of reality" and are not just chosen to fit the researchers needs.

The IEEE 802.11 wireless local area network (WLAN) technology was chosen for the following reasons: it is standardized, operates in a license-free frequency band, it is widely used, and, furthermore, hardware components are commercially available. Specifically, it was possible to obtain a chipset which only implements the PHY and without any upper layer (MAC) functionality (Harris/Intersil PRISM I chipset [1] [13]). We use the results from our measurements for assessing the suitability of selected well-known stochastic models and for developing a new approach to stochastic channel modeling for the purpose of packet level simulation. This approach includes an overall model structure and a new class of stochastic models.

The report is structured as follows: in Chapter 2 we describe our measurement setup, including the most important characteristics of the PRISM I chipset. In Chapter 4 we discuss the goals of our measurements, and how these are addressed by chosing the environment and the set of fixed and variable parameters. In Chapter 3 we discuss basic approaches and notions for the evaluation of our measurement results (including the notion of *indicator sequences*), and in Chapter 5 we present the measurement results, both for packet related phenomena and bit error phenomena. Furthermore, a brief overview on related literature covering wireless measurements is given. In Chapter 6 we cover the issue of stochastic modeling of bit error and packet loss processes. After a brief overview on popular stochastic models, we introduce the class of *bipartite models*. Then we give a recommendation for the overall structure of a stochastic model capable of capturing the main findings of our measurements. Furthermore, we present the results of a simulation study, which, for a simple protocol, shows that the choice of different channel models influences selected performance parameters. Finally, in Chapter 7 we discuss our findings and give the conclusions.

A more detailed presentation of our measurements can be found in several technical reports [37], [35], [36], [34], available at our web site *http://www-tkn.ee.tu-berlin.de.* Furthermore, we provide at this site the set of traces and evaluation scripts.

Chapter 2

Measurement Setup

In this section we give a brief overview on our measurement equipment. We start in Section 2.1 with the most important properties of the IEEE 802.11 PHY. In the following Section 2.2 we describe our measurement setup, including the controllable parameters (a more detailed discussion of the setup can be found in [37]).

2.1 IEEE 802.11 / PRISM I PHY

In 1997, the IEEE 802.11 standard was finalized [21], [22], describing a WLAN operating in the license free 2.4 GHz ISM band (Industrial, Scientific and Medical band) and offering different bit rates: 1, 2, 5.5 and 11 MBit/s. The standard defines the PHY (covering, e.g., modulation, spreading and the format of PHY packets) and the MAC layer (method of arbitrating access to the wireless channel), however, this does not consider the MAC protocol. We have used a MAC-less radio modem (Harris/Intersil PRISM I chipset [1]), which is compliant with IEEE 802.11 and uses the direct sequence spread spectrum (DSSS) PHY, which is much more often used than the companion frequency hopping spread spectrum PHY.[1] It offers the following modulation types/bitrates: 1 MBit/s with (D)BPSK modulation (Differential Binary Phase Shift Keying), 2 MBit/s with (D)QPSK (Differential Quaternary Phase Shift Keying), 5.5 MBit/s with BMBOK (Binary M-ary Bi-Orthogonal Keying), 5.5 MBit/s with CCK (Complementary Code Keying), 11 MBit/s with QMBOK (Quaternary M-ary Bi-Orthogonal Keying) and 11 MBit/s with CCK modulation. The CCK modes are compliant to IEEE 802.11, the BMBOK and QMBOK modes are only present for compatibility reasons

[1] The DSSS PHY uses 11 chip barker sequences, which, by their unique autocorrelation properties, can be used for bit synchronization purposes.

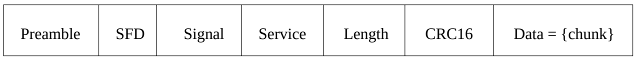

Figure 2.1: Format of a PHY frame

and not used in this study. It is possible to attach two antennas to the modem and to use receiver diversity (i.e. the receiver selects the antenna with the maximum signal level). The transmitter power was fixed at 18 dBm, corresponding to 63 mWatt. The radio modem basically consists of high frequency circuitry and a baseband processor. The latter accepts and delivers a serial bit stream from upper layers, optionally performs scrambling (employing a shift register with feedback), performs DSSS processing, and generates / receives PHY packets [13]. The characteristics of the serial bit stream is our focus of interest.

The PHY packet format is shown in Figure 2.1. It contains no MAC-related fields. A packet starts with a well-known preamble of fixed length, followed by a constant value indicating the start of a packet (start frame delimiter, SFD). The preamble and SFD allow the receiver to synchronize on the sender's clock (bit synchronization) and to determine the start of the packet. The signal field indicates the modulation type used in the data portion of the packet, while the length field indicates the length of the data portion in microseconds (the service field has no significance). The CRC16 field contains a 16 bit cyclic redundancy check (CRC) checksum which is computed from the three previous values. If the checksum is wrong or the signal field carries an unknown value, the whole packet is discarded by the baseband processor. It is important to note that while the data part can use different modulation types, the PHY header is always transmitted with BPSK modulation. When the data part uses another modulation type, both transmitter and receiver must switch the modulation type within a PHY packet (more precisely, after the CRC16 field).

2.2 Measurement setup

We used two dedicated stations, a *transmitter station* and a *receiver station*, which do not change their roles during a measurement. The setup is sketched in Figure 2.2. The basic idea is that the transmitter station sends a well-known packet stream over the wireless link, which is captured and stored by the receiver station into a logfile (the contents of the logfile is denoted as *trace*). For generation and reception of the packets we use a microcontroller board carrying the radio modem and a separate processor (Motorola PowerQUICC [18] with Tundra PCI Interface [28] and a 50 MHz PowerPC 603e processor). The coupling to the (Windows

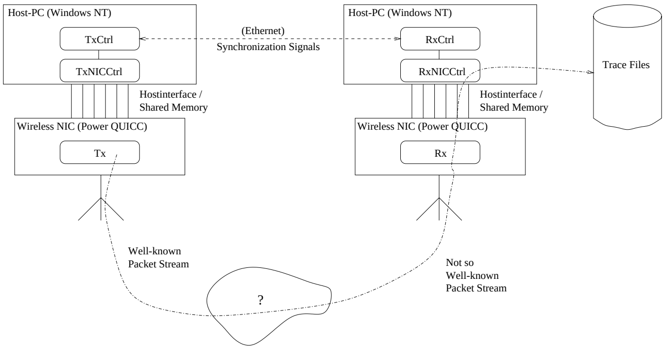

Figure 2.2: Measurement Setup

NT-based) host is achieved with a segment of 64 kByte shared memory, denoted as *host interface*. We call this board a *wireless NIC* (Network Interface Card). Please note that there is no MAC protocol implemented on the wireless NIC, nor are there any higher layer protocols or operating systems involved in the packet generation and reception process, which could bias the measurement results. Instead, the wireless NIC carries only a measurement application.

We briefly discuss the different software modules of our measurement setup, please refer to Figure 2.2. The *Tx module* is located on the wireless NIC on the transmitter station. It accepts configuration commands from the TxNICCtrl module (allowing to set the variable parameters) and generates a well-known *packet stream*. A packet stream consists of a given number of *packets*, which are transmitted at equidistant start times and with the same parameters and packet size and carrying a well-known content. A packet's data part consists of a number of *chunks*, each chunk is a one bit to eight bit coded sequence number of 32 bit length (with $0 \mapsto 11000011$ and $1 \mapsto 00111100$), along with four bytes of heading and trailing. Thus, a chunk consists of 36 bytes (288 bits). The *Rx module* is also located on the wireless NIC. Its main task is to capture packets from the wireless link, to add metainformation (e.g., timestamps, packet size, reception status, and signal strength) and passing them to the host via the host interface. The resulting stream of received packets found in the logfile is denoted as *trace*. The *TxNICCtrl module* is merely a wrapper, which offers a command line interface of the capabilities of the Tx module to Windows NT users. It allows to set transmission parameters, to start and stop packet generation, etc. The *RxNICCtrl module* serves also as a wrapper to the Rx module, which additionally can store captured packets into a logfile. The *TxCtrl module* is a script, which synchronizes itself with the RxCtrl software for controlling the measurements (using a TCP connection over the ethernet). It accepts the parameters to be used for the next packet stream and starts transmissions appropriately (using the services of the TxNICCtrl module). Finally, the *RxCtrl module* is actually controlling a whole measurement. It loops over all variable parameters; for each point of the parameter space a packet stream is started (by triggering the TxCtrl module) and the trace is logged onto the harddisk. The evaluation of the traces is done off-line, employing several perl scripts.

For gaining a better understanding of some of the observed phenomena it is beneficial to give some more details of the receive process. The Rx module is responsible for the transfer of received packets from the baseband processor into the microconcontroller's memory. The baseband processor offers a serial hardware interface and generates the corresponding clock (`CLK`) and clock envelope (`MD_RDY`) signals. Furthermore, on changes of the `MD_RDY` signal the baseband processor generates interrupts for the microcontroller. The microcontroller uses a serial communications controller (SCC, a DMA-like device) for transferring the data from the serial interface into a receive buffer (or the other direction for sending data). After the baseband processor has acquired bit synchronization and detected the SFD field, it enables the `MD_RDY` signal, generates an interrupt (denoted as *start interrupt*) and streams the following header fields into the receive buffer (including the length field), employing the SCC set up before. On receiving the interrupt, the microcontroller reprograms the SCC with the correct length (found in the receive buffer) and waits for the next interrupt (denoted as *stop interrupt*). This is generated either prematurly by the baseband processor on changing the `MD_RDY` signal (e.g., on occasion of losing bit synchronization or after an error in the PHY header), or by the SCC after transferring the requested number of bits. After this interrupt the microcontroller adds the metainformation to the packet and allocates a new receive buffer from a finite pool. Due to the lack of dedicated hardware we have to process the interrupts and baseband processor signals in software using the microcontroller, which introduces problems with interrupt latencies ($\approx 100\ \mu$s), especially in the case of an erroneous PHY header, where two interrupts occur in a short time (the time from end of SFD to the end of the CRC16 field 64 μs).

Our setup allows for variation of several parameters, which are related both to the properties of the radio modem and trace generation. The important modem related parameters are shown in Table 2.1 and the set of trace related parameters is shown in Table 2.2.

Parameter	**Description**
ScramblingEnabled	determines whether scrambling is used
DiversityEnabled	determines whether receiver antenna diversity is used
Frequency	802.11 carrier frequency used
PreambleLength	number of bits for PHY preamble
ModulationCode	distinguishes modulation used for data portion: 1 MBit/s BPSK, 2 MBit/s QPSK, 5.5 MBit/s CCK, 5.5 MBit/s BMBOK, 11 MBit/s CCK, 11 MBit/s QMBOK

Table 2.1: Adjustable radio parameters

Parameter	**Description**
NumPackets	Number of Packets
GapTime	Time gap between two packets
NumChunks	Number of chunks per packet

Table 2.2: Adjustable trace parameters

Chapter 3

Measurement Evaluation

Much of the evaluation of the measurements uses the notion of *indicator sequences* or the more special *binary indicator sequences.* In general, an indicator sequence is a finite sequence of natural numbers, a binary indicator sequence is restricted to the values zero and one. In the latter case, often we associate with a 1 an error event (e.g., an erroneous bit or a lost packet) and with a 0 the correct event. We can view a binary indicator sequence as finite subset of a sample path of a random process $\{B_n\}_{n \in \mathbb{N}}$, where each B_i is a Bernoulli random variable.

The evaluation proceeds in two steps. In the first step "suspicious" packets are identified and marked as lost packets (see Sections 5.1). In the second step we have generated two important binary indicator sequences. The *packet loss indicator sequence* of a single trace is constructed by marking lost packets with a 1 and received packets with a 0. For the received packets of a trace we generate the respective *packet bit indicator sequences* by `XOR`ing the received packet with the expected packet. The *trace indicator sequence* is then formed by concatenating all packet bit indicator sequences for all received packets of a single trace (in order of increasing packet numbers). Please note that in the trace indicator sequence any information about packet boundaries, lost packets, or packet gap times is completely ignored.

3.1 Indicator Sequences

We subdivide binary indicator sequences into *error bursts* and *error free bursts*, where intuitively an error burst is a subsequence where many errors occur. More formally, we define an error free burst of order k_0 to be an all-zero subsequence of a binary indicator sequence with a length of at least $k_0 + 1$ bits. In contrast, within the same error burst of order k_0 we allow between any two neighboured erroneous bits (indicated by a one) at most $k_0 - 1$ indicator bits of value 0.

Using this definition a binary indicator sequence $i_1 i_2 \dots i_m$ is segmented into alternating error bursts and error free bursts. The length of the j-th error free bursts is denoted as X_j, the length of the j-th error burst is denoted as Y_j and Z_j is the actual number of bit errors occuring in the j-th error burst. We write this as the *burst length sequence*[1]:

$$X_1, Y_1, Z_1 \quad X_2, Y_2, Z_2 \quad \dots \quad X_p, Y_p, Z_p$$

We denote the sequence $X_1 X_2 \dots X_p$ as the *error free burst length sequence*, $Y_1 Y_2 \dots Y_p$ as the *error burst length sequence* and $\frac{Z_1}{Y_1}\frac{Z_2}{Y_2} \dots \frac{Z_p}{Y_p}$ as the *error density sequence.* As an example, take the binary indicator sequence 001001010001100011000000. With a burst order of $k_0 = 1$ we get the burst length sequence

$$2,1,1 \quad 2,1,1 \quad 1,1,1 \quad 3,2,2 \quad 3,2,2 \quad 5,$$

while for $k_0 = 2$ we obtain:

$$2,1,1 \quad 2,3,2 \quad 3,2,2 \quad 3,2,2 \quad 5.$$

It is important to note that with only recording the number Z_j of errors within error burst Y_j we loose information about the exact error positions.

Using the burst length sequences, some simple statistics can be computed, e.g. the mean error rate $\bar{e}$ by

$$\bar{e} = \frac{\sum_{j=1}^{p} Z_j}{\sum_{j=1}^{p}(X_j + Y_j)}$$

or the mean error burst length by

$$\bar{Y} = \frac{1}{p}\sum_{j=1}^{p} Y_j.$$

Accordingly, some other simple first order statistics (variance, coefficient of variation, from now on denoted as CoV) can also be computed based on the burst length sequence.

For a binary indicator sequence $i_1 i_2 \dots i_m$ the conditional probability $\Pr[i_{n+k} = 1 | i_n = 1]$ for $(k \geq 1)$ is of some interest. It is approximated as follows (frequency based approach):

$$\begin{aligned} \Pr[i_{n+k} = 1 | i_n = 1] &\approx \frac{\#[\text{cases with } i_n = 1 \text{ and } i_{n+k} = 1]}{\#[\text{cases with } i_n = 1]} \\ &= \frac{\sum_{j=1}^{m-k} i_j \cdot i_{j+k}}{\sum_{j=1}^{p} Z_j}. \end{aligned}$$

[1] In the presentation we deliberately neglect the fact that a binary indicator sequence may start with an error burst or may end with an error free burst. Furthermore, we do not indicate explicitly the dependence on k_0 in the notation.

This conditional probability is related to the correlation function of the binary indicator sequence, since, with the assumption of equally distributed i_k (with mean $\bar{e}$ and variance $\sigma^2 = \bar{e}(1-\bar{e})$) we have:

$$\begin{aligned}
\mathrm{Corr}[i_n, i_{n+k}] &= \frac{\mathrm{Cov}[i_n, i_{n+k}]}{\sqrt{\sigma^2\sigma^2}} \\
&= \frac{\mathrm{E}[i_n i_{n+k}] - \mathrm{E}[i_n]\,\mathrm{E}[i_{n+k}]}{\bar{e}(1-\bar{e})} \\
&= \frac{\mathrm{E}[i_n i_{n+k}] - \bar{e}^2}{\bar{e}(1-\bar{e})} \\
&= \frac{\Pr[i_{n+k} = 1 | i_n = 1]\bar{e} - \bar{e}^2}{\bar{e}(1-\bar{e})} \\
&\approx \Pr[i_{n+k} = 1 | i_n = 1]
\end{aligned}$$

where the approximation holds for small $\bar{e}$ values. Here we have used that

$$\begin{aligned}
\mathrm{E}[i_n i_{n+k}] &= \sum_{x,y\in\{0,1\}} xy \Pr[i_n = x, i_{n+k} = y] \\
&= \Pr[i_n = 1, i_{n+k} = 1] \\
&= \Pr[i_{n+k} = 1 | i_n = 1] \cdot \Pr[i_n = 1] \\
&= \Pr[i_{n+k} = 1 | i_n = 1] \cdot \bar{e}
\end{aligned}$$

We analyze the traces using several values for k_0, since the results achievable when restricting to a single value of k_0 can be misleading.

For a nonbinary indicator sequence $s = s_1 \ldots s_n$ built up from finitely many values (i.e., $s_i \in \{v_1, \ldots v_k\} \subset \mathbb{N}_0$) we define a v_i-*burst* to be a subsequence s' of s such that all members of s' have the value v_i and the neighboring members of s have different values (or s' is situated at the fringe of s). We need this notion for discussion of packet impairments (see Section 5.1).

Chapter 4

Measurement Parameters and Environment

We have conducted two measurement campaigns[1] in an industrial environment, namely at the Produktionstechnisches Zentrum (PTZ) in Berlin, Germany. The PTZ is a research facility for machinery engineering, supported by industry and academia . The first campaign was performed on June 26, 2000 and its main purpose was to evaluate our measurement setup and to find out which phenomena are important [37]. The second campaign was from Aug. 28 to Aug. 30, 2000 and was aimed at results [35], [36], [34]. In this report, we focus on the second campaign.

4.1 Environment

The PTZ owns a large factory building, which contains several machines of different types and with people going around all the time. The ground plan of the building has the shape of a circle. At the fringe of the circle is a path, which can be used by small vehicles, while the inner circle contains the machinery. During both campaigns we have chosen the same positions within the building. In Figure 4.1, we show the relative position of our measurement equipment in the factory building, while in Figure 4.2 we show the close neighbourhood, especially the machines that are in close proximity. We have investigated a non line-of-sight

[1] To avoid confusion, we give the following definitions: a *measurement campaign* consists of one or more *measurements*. A measurement is distinguished by the set of fixed and variable parameters from other measurements, and it consists of a number of packet streams. A packet stream consists of a fixed number of packets, which are all transmitted with the same parameters (modulation type, packet size, gap time, and so forth).

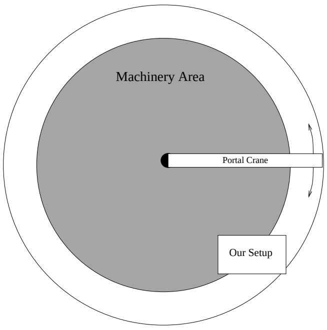

Figure 4.1: Position of our Setup within the Building

(NLOS) scenario, with a closet in between the transmitter and receiver station, and the die sinking electrical discharge machine's (EDM) working area very close to the direct path. Both stations are ≈ 7-8 meters apart and not moved during the measurement campaigns. The receiver station was in close proximity (≈ 1 m) to a cabinet containing the power supply for a huge 5 axis milling machine, which, however, was not active during the second campaign. The die sinking EDM was active most of the time, except when changing the workpiece. A second EDM machine was situated behind the first one (see Figure 4.2). It was used by PTZ staff almost all the time. At the ceiling, in a height of ≈ 8 meters, was a portal crane, capable of moving around 20 tons. Its motors are placed at the fringe's end of the portal crane. The crane was used during the first two days of the second campaign.

Instead of investigating different scenarios with a restricted set of measurements, we have chosen to focus on the single scenario described above. This concentration allowed us to get a more in-depth insight into different aspects of wireless transmission (e.g., long-term behavior). The reason for choosing a NLOS scenario is that the measurement results should help in the design of MAC protocols and coding schemes for industrial WLANs, where we have hard requirements regarding timing behavior and reliability. Hence, it is interesting to see how MAC protocols and coding schemes react in the case of bad channel conditions, which can occur in reality.

4.2 Parameters

The second measurement campaign is designed to assess: a) the packet loss and packet impairment behavior on short and long timescales; b) the long term bit error rate behavior; c) the dependency of the bit error behavior on packet sizes and modulation types; and d) the

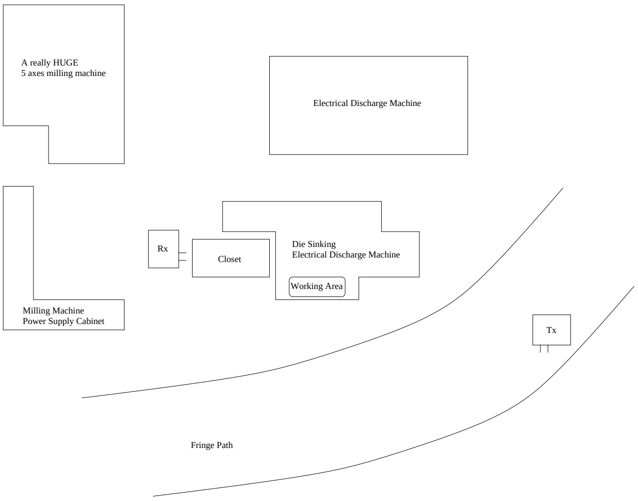

Figure 4.2: Setup of PTZ measurement

dependency of packet losses and impairments and bit error behavior on the scrambling mode. Furthermore, we wanted to find out whether the phenomena found in the first campaign [37] are due to the specific pair of modems used or not. Since we are interested in capturing the raw *link* behavior, we have assured that no external interferers in the same frequency band (e.g., IEEE 802.11 LANs) were present.

We have performed three different measurements within the second campaign: the **longterm1** measurement is a long term measurement performed with a single modulation type and packet size, only varying the scrambling mode (addressing a), b) and d)). The **longterm2** measurement is the same as the **longterm1** measurement, however, another pair of radio modems was used. In the **factorial** measurement we have varied the scrambling mode, modulation type and packet sizes, and for each point in the parameter space, the short term bit error behavior was investigated (thus addressing c)). The main purpose of the **longterm2** measurement was to confirm that the observed phenomena are not due to the particular pair of radio modems used. Indeed, our results confirm this belief and allow us to restrict the discussion to the results obtained with the first pair of radio modems (the **longterm1** measurement and **factorial** measurement).

We have chosen for the **longterm1** and **longterm2** measurements to keep all parameters fixed, except the scrambling mode (on, off) and the pair of radio modems used (see Table 4.1). In both measurements we have taken 90 traces for every scrambling mode. With 90 traces, 2 hours and 10 minutes are covered. Within a measurement the traces are numbered

Parameter	**Value**
PreambleLength	128 bits
DiversityEnabled	True
Frequency	12
NumPackets	20000
NumChunks	14 (504 bytes)
GapTime	1000 μsec
ModulationCode	2 MBit/s QPSK

Table 4.1: Fixed Parameters for **longterm1** and **longterm2** measurements

Parameter	**Value**
PreambleLength	128 bits
DiversityEnabled	True
Frequency	12
NumPackets	20000
GapTime	1000 μsec

Table 4.2: Fixed Parameters for **factorial** measurement

consecutively, thus the trace number corresponds to the time axis. Within the **longterm1** measurement the first 90 traces are taken without scrambling, the other 90 traces are taken with scrambling. For the **factorial** measurement we have chosen a factorial design, varying the modulation type, packet size, and the scrambling mode as summarized in Table 4.3, while keeping the remaining parameters fixed (Table 4.2). For every point in the parameter space two traces are taken. The traces are numbered consecutively, i.e., in the order of their occurence. The traces 1 to 56 are taken without scrambling, the traces 57 to 112 with scrambling. Within each of the two groups we have varied the modulation scheme from low bitrates to high bitrates and for each modulation scheme we have varied the packet sizes from small packets to large packets.

Parameter	**Value**
ScramblingEnabled	True, False
ModulationCode	1 MBit/s BPSK, 2 MBit/s QPSK, 5.5 MBit/s CCK, 11 MBit/s CCK
NumChunks	3, 9, 14, 28, 56, 112, 167 (corresponding to 108, 324, 504, 1008, 2016, 4032, 6012 bytes)

Table 4.3: Variable Parameters for **factorial** measurements

Chapter 5

Measurement Results

In this section, we discuss the most important results of the second measurement campaign. The presentation is divided into two parts: first we discuss *packet impairments*, i.e., some phenomena regarding whole packets (Section 5.1), then we discuss the bit error behavior of the remaining packets (Section 5.2). A more detailed presentation can be found in [35, 36, 34]. Furthermore, in this section we provide a brief overview of the literature about wireless measurements (Section 5.3)..

5.1 Packet Impairments

We have encountered five different types of packet impairments: *packet losses*, *oversized packets*, *truncated packets*, *ghost packets*, and packets with *bit shifts.* We first give a brief explanation of these phenomena including methods of detection, then we present the most important statistics.

5.1.1 Description of Phenomena

There are three major causes of transmission errors (see Sections 2.1 and 2.2): the failure of acquiring bit synchronization or to properly detecting the start frame delimiter, an error in the PHY header fields (e.g., wrong value in signal field or CRC error), and bit errors in the packet's data part. The latter are discussed in Section 5.2.

The failure of acquiring bit synchronization leads to packet losses. For a lost packet the receiver station's baseband processor does not generate any interrupt or `MD_RDY` transition, nor are any data bits passed to the microcontroller. The identification of lost packets is based on the packet timestamps and their comparison with the *InterpacketTime* (given as the sum of the *GapTime*, the fixed length PHY header and the known length of the data part). If t_1 and t_2 are timestamps of subsequently dumped packets, we calculate

$$g = round\left(\frac{t_2 - t_1}{InterpacketTime}\right).$$

The number of lost packets between t_1 and t_2 is then simply given by $g - 1$.[1] It is important to note that the packet timestamp denotes the instant of time when the packet delivery from the baseband processor to the microcontroller is completed (or aborted).

Errors in the PHY header fields lead to more subtle effects, due to problems with interrupt latencies, as explained in Section 2.2. In this case, our software, on receiving the start interrupt, reads the length field from the buffer (which, however, may be wrong) and reprograms the SCC. If now the stop interrupt immediately follows, then the SCC transfer is aborted (leaving the receive buffer in a state with old data), the metainformation is generated (including the length field read before and the timestamp) and the receive buffer is delivered. At the time of performing the measurements our software was not able to detect these circumstances as erroneous. It acts as if the stop interrupt were correct and treats the buffer as a correctly received one, which, however, contains old data. When we want to detect this behavior in an off-line evaluation, we require at least one chunk within a packet to be correct. Then we can determine whether the corresponding sequence number was an already used one.

This behavior gives rise to different phenomena: in the case of *ghost packets* the baseband processor signals the correct packet length to the microcontroller. After inspecting several traces all packets classified as ghost packets show one of two patterns: the first chunk of the packet is a correct but old chunk (thus no data bit is transferred into the buffer), or some bit errors occur within the first few bits and the second chunk is a correct but old chunk. In the latter case some bits are transferred into the buffer.

In other cases the baseband processor passes a wrong packet length to the microcontroller and transfers no data. Since there was no interferer present (e.g., an IEEE 802.11 access point producing beacon packets) it is reasonable to assume that these packets are produced by errors in the transmission channel or radio hardware. If the packet length is too short (w.r.t. *NumChunks* times the size of a chunk) we have a *truncated packet*, if it is too long we refer to this as a *oversized packet* (both classes are summarized as *missized packets*). All

[1] The timestamps generated by the Rx module of the receiver station show a little bit of jitter (well below 1% of the InterpacketTime for all traces), which can be attributed to interrupt latencies in both stations. Because of this jitter, the difference between t_1 and t_2 should be small enough for the formula to be valid.

examined missized packets would have been classified as a ghost packet, when their sizes are assumed to be correct.

A phenomenon which is probably not due to errors in the PHY header are packets with *bit shifts*. In this case, the baseband processor delivers the correct packet length and an appropriate number of bits. However, somewhere in the packet's data some random bits are inserted into or deleted from the bit stream. As a result, the following bit sequence is a left- or right-shifted version of the original sequence. From examining all the candidate packets of our traces we have learned that the first bit shift tends to occur at the beginning of packets, but this is not a general rule.[2] A further result is that for both the **longterm1** and **longterm2** measurements the data patterns of shifted packets transmitted with scrambling and without scrambling differ substantially: almost all shifted packets without scrambling show only one shift and the data patterns are very similar, while for shifted packets with scrambling the data patterns look much more diverse and frequently two or more shift events are observed within a single packet. This is a strong indication that bit shifts happen "below" the scrambling unit, and thus deeply in the radio modem. Furthermore, in a separate measurement campaign we have observed that variation of channel conditions (here: opening or closing a door between transmitter and receiver station) has an effect on the presence and rate of bit shifted packets. This is a strong indication that these packets cannot be attributed to a failure of the transmitter station.

Unfortunately, it is not easy to detect bit shifted packets. The baseband processor always gives a correct packet length and correct status values. One possible approach based on the packet's contents would be to examine the frequencies of the received byte values and to look for anomalies, however, this requires substantial computation time and a set of good heuristics. For this reason we resort to another criterion: if a packet has more than 10% erroneous bits we assume this to be induced by bit shifts and the packets are filtered out (we denote them as *dropped packets*). An assessment of the quality of this heuristic based on several **longterm1** traces with the highest numbers of dropped packets can be found in Table 5.1. The table shows the number of dropped packets (due to fraction of erroneous bits $>$ 10%), how many of them are truly bit shifted, the number of packets with a fraction of erroneous bits between 4% and 10% (denoted as *intermediate* packets) and how many of them are shifted. For the packets with more than 10% errors the matching between dropped and bit shifted packets is very good, which is not true for the intermediate packets. Hence,

[2] For example, trace 152 of the **longterm1** has 600 packets with bit shifts leading to more than 10% incorrect bits, 285 of which show the shift in the first chunk, while we have only 45 packets with between 4% and 10% incorrect bits, 31 of which can be attributed to bit shifts occuring in the last third of the packet.

trace	# dropped	thereof shifted	# intermediate	thereof shifted
3	10	10	-	-
90	14	14	-	-
139	135	122	47	10
140	43	43	16	2
141	23	23	8	-
143	30	29	6	5
149	21	21	3	1
151	33	33	16	5
152	600	598	45	31
158	70	70	5	5

Table 5.1: Evaluation of the quality of the 10% rule for selected **longterm1** traces

Code	**Meaning**
0	fully received (maybe with bit errors)
1	lost packet
2	dropped packet
3	truncated packet
4	oversized packet
5	ghost packet

Table 5.2: Packet Fate Codes

we conclude that the 10% rule is a good choice. However, for the 5.5 MBit/s CCK and 11 MBit/s CCK traces of the **factorial** measurement the rule fails, since many packets show high fractions of bit errors without being shifted.

5.1.2 Statistics of Phenomena

For every trace of the **longterm1** measurement, we have determined the fate of for all packets (coded as an integer number, see Table 5.2) and built the *trace fate sequence* (a nonbinary indicator sequence built from these codes). By concatening all trace fate sequences of a measurement in consecutive order, we obtain the *full compound fate sequence*, denoted as $s = i_1 i_2 \dots i_n$. In Table 5.3 some simple statistics of the different code bursts are summarized for the **longterm1** measurement for the full compound fate sequence. There are several important points. At first, the phenomena of dropped, truncated, oversized or ghost packets

Code	% packets	BL: mean	BL: CoV	BL: max
0	93.5098	51.0045	12.5673	19999
1	6.2836	3.6111	17.3081	14937
2	0.0324	1.5116	0.8980	20
3	0.0383	1.0430	0.1946	2
4	0.0197	1.0456	0.2064	3
5	0.1160	1.0278	0.1880	4

Table 5.3: First order statistics of compound packet fate sequence of **longterm1** measurement (BL: burst length)

$$\begin{pmatrix} 0.9803 & 0.0179 & 0.0001 & 0.0003 & 0.0001 & 0.0010 \\ 0.2733 & 0.7230 & 0.0007 & 0.0003 & 0.0006 & 0.0018 \\ 0.5064 & 0.1139 & 0.3384 & 0.0068 & 0.0094 & 0.0248 \\ 0.6049 & 0.2554 & 0.0050 & 0.0412 & 0.0130 & 0.0803 \\ 0.3774 & 0.4901 & 0.0070 & 0.0295 & 0.0436 & 0.0521 \\ 0.5906 & 0.3399 & 0.0047 & 0.0256 & 0.0119 & 0.0270 \end{pmatrix}$$

Table 5.4: Transition probabilities for packetimpairments (**longterm1** measurement)

are negligible, and in our further discussion they are treated as lost packets (by simply remapping their codes in the full compound fate sequence, obtaining the *compound fate sequence*). Second, the phenomena of truncated, oversized, and ghost packets tend to occur in single packet bursts, as indicated by their small variations and means. The occurence of dropped packets is slightly more bursty. The overall packet loss rate is $\approx 6.1\%$, which is non-negligible and needs to be considered in accurate channel models. Packet losses tend to occur in short bursts, but the burst lengths are highly variable, and very long bursts can be observed. Bursts of correct packets have the highest mean length value, but are even more variable than packet loss bursts. Table 5.4 shows the matrix $\mathbf{F} = ((f_{kl}))_{k,l\in\{0,\ldots,5\}}$ with $f_{kl} = \Pr[i_{\nu+1} = l|i_\nu = k]$. Packet losses (code 1) are most likely followed by further packet losses or by correct packets. The other impairments (dropped, oversized, truncated and ghost packets) are followed by lost packets with high probability. To show that correlation of packet losses lasts over longer periods, we present in Figure 5.8 for the compound fate sequence the conditional probability that packet $\nu + k$ is lost, given that packet ν is lost. This figure and some visualizations of the compound fate sequence not shown here confirm that that packet losses tend to occur in bursts. There are also long sequences with no packet losses.

	factorial	**longterm1**	**longterm2**
w scrambling	9885	34755	26392
w/o scrambling	20411	191456	45159

Table 5.5: Number of lost packets

The packet loss behavior is thus varying over time.

For the compound fate sequence we have built the associated burst length sequence with order $k_0 = 1$ (according to Section 3.1). Let $X_1 X_2 \dots X_p$ be the packet loss free burst (PLFB) length sequence of the compound fate sequence and $Y_1 Y_2 \dots Y_p$ be the corresponding packet loss burst (PLB) length sequence. First, we have computed the histograms of the X_i's and Y_i's. The histograms, together with the assumption that all X_i (Y_i) are identically distributed, are used for computation of the cumulative density functions. These functions are shown in Figures 5.5 (for the PLB lengths) and Figure 5.6 (for the PLFB lengths). The shapes are very regular, the PLFB lengths are more heavy tailed than the PLB lengths. Furthermore, we show exemplarily in Figure 5.7 the histogram of the PLB lengths for **longterm1** trace 1 (with a packet loss rate of $\approx 32\%$) and burst order $k_0 = 1$. It is representative for the **longterm1** measurement in the sense that all traces show the same regularly decaying shape, sometimes with small fluctuations and the main mass in the first ten burst lengths.[3] In order to find out whether the burst length sequences show correlation, we have computed their autocovariance functions, using the standard approximation formula [4] (shown here for PLFB lengths):

$$R_X(k) \approx r_k = \frac{c_k}{c_0} \quad (k \in \{1, \dots, p-1\})$$

where

$$c_k = \frac{1}{p} \sum_{t=1}^{p-k} (X_t - \bar{X})(X_{t+k} - \bar{X})$$

The rule of thumb is that if for some $k \geq 1$ the absolute value of r_k exceeds 0.2 then there is more than weak correlation and thus the X_i cannot be independent. In Figure 5.10, the autocovariance function for the PLFBs is shown, in Figure 5.9, the same is displayed for the PLBs. The conclusion is that the PLB length sequence $Y_1, \dots, Y_p$ shows (for all positive lags) virtually no correlation (and thus can be modeled as independent), while for the PLFB length sequence we have strong correlation on the first five lags, then correlation gets weak.

[3] When moderately increasing the burst order k_0 (e.g. $k_0 \in \{4, 8, 15, 20, 30\}$) it can be seen that still many single packet losses occur, however, the remaining histograms "converges" towards a nearly linear shape with small negative slope.

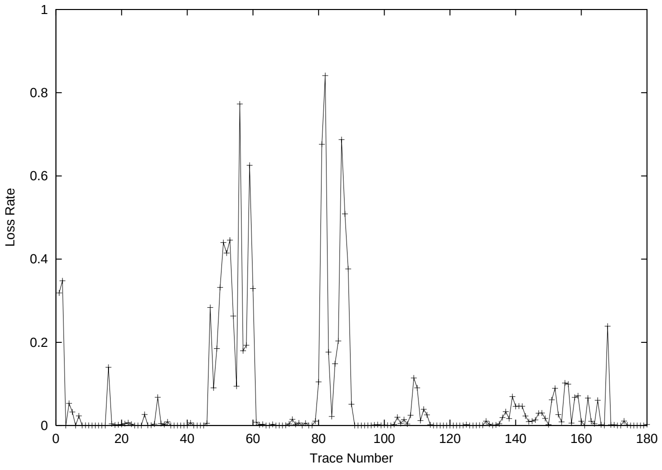

Figure 5.1: Rates of lost packets for **longterm1** measurement

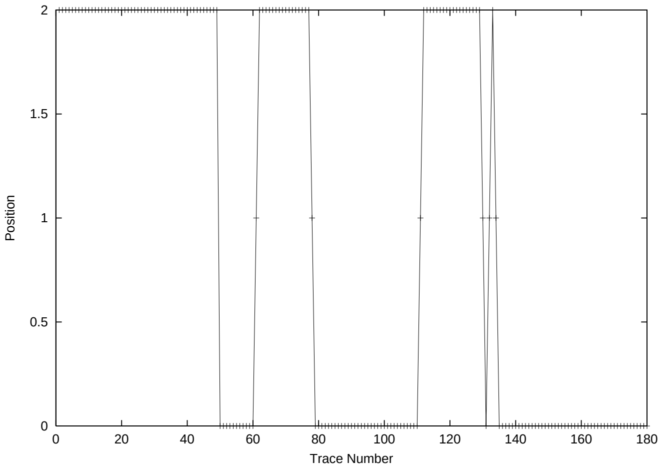

Figure 5.2: Position of portal crane (0=close proximity, 1=short distance, 2=longer distance) for **longterm1**-measurement

	w scrambling	w/o scrambling
BPSK	1017	1468
QPSK	3128	3734
BMBOK	510	20
QMBOK	5230	15189

Table 5.6: Number of lost packets for different modulation types (**factorial** measurement)

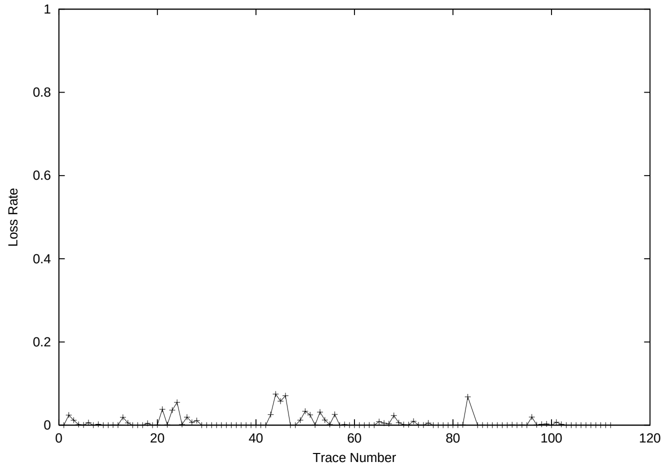

Figure 5.3: Rates of lost packets for **factorial** measurement

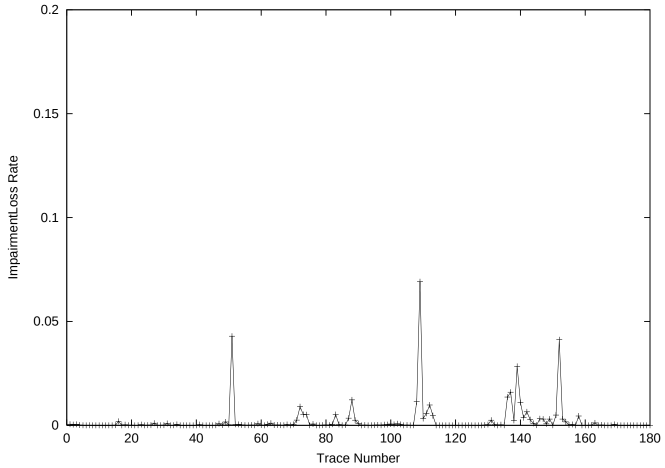

Figure 5.4: Rates of dropped, truncated, oversized and ghost packets for **longterm1** measurement

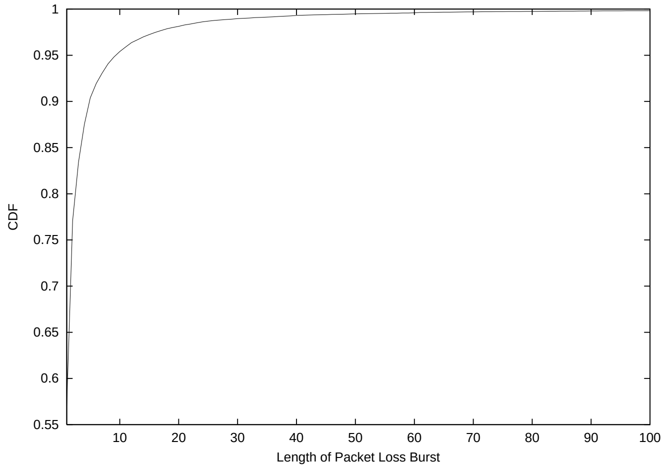

Figure 5.5: Cumulative Distribution Function of packet loss burst lengths for the **longterm1** compound fate sequence ($k_0 = 1$)

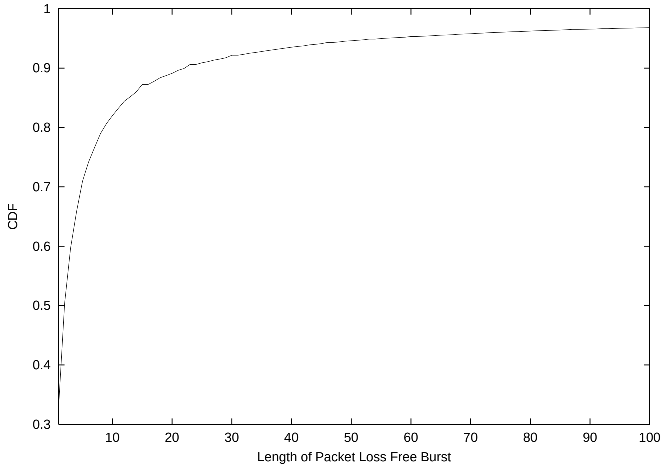

Figure 5.6: Cumulative Distribution Function of packet loss free burst lengths for the **longterm1** compound fate sequence ($k_0 = 1$)

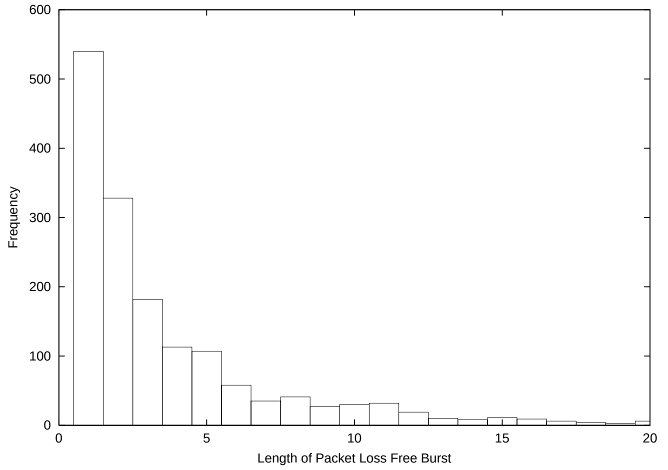

Figure 5.7: Histogram of packet loss burst lengths for **longterm1** trace 1 and $k_0 = 1$

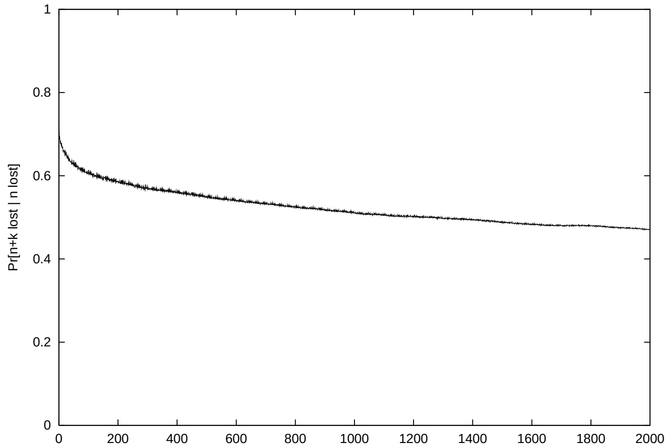

Figure 5.8: Conditional Probability that packet $n + k$ is lost given that packet n is lost/impaired for **longterm1** compound fate sequence

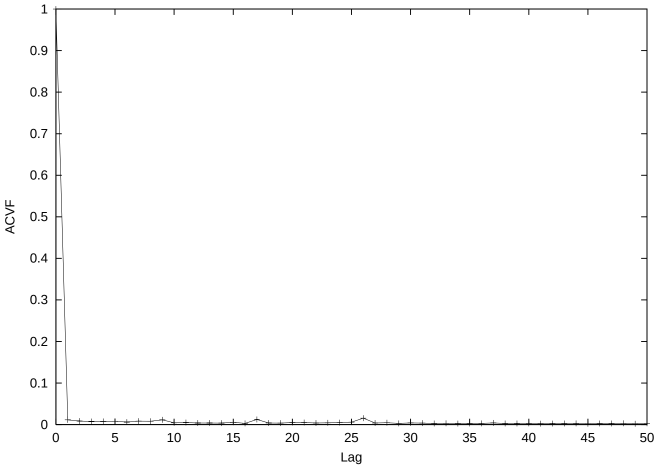

Figure 5.9: Autocovariance function of packet loss burst lengths for the compound fate sequence and $k_0 = 1$ for **longterm1** measurement

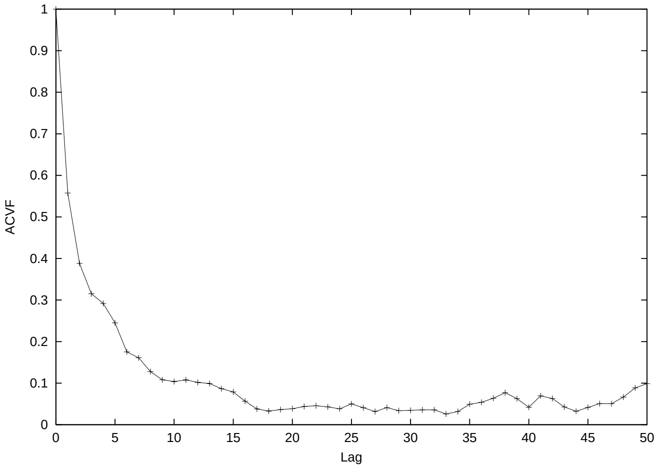

Figure 5.10: Autocovariance function of packet loss free burst lengths for the compound fate sequence and $k_0 = 1$ for **longterm1** measurement

So far we have discussed the overall statistics of the packet impairments. In order to show the time-varying nature of these phenomena, we present in Figure 5.1 the rate of lost (i.e., code 1) packets of a single trace vs. the trace number for the **longterm1** measurement; Figure 5.3 displays the same for the **factorial** measurement. The packet loss rates are sometimes very high and strongly varying. In Figure 5.4, we show the rate of the remaining packet impairments for the **longterm1** measurement. A possible explanation of the behavior of the curves offers Figure 5.2, where the "portal crane function" for the **longterm1** measurement is shown. This function displays the distance of the portal crane to our setup (0 = directly above the setup, 1 = no more than five meters away, 2 = more than five meters away). It can be seen that, except for a peak at traces one and two, the packet loss rates have the highest rates and the highest degree of fluctuation when the portal crane is close to the setup. During the **factorial** measurement the portal crane was not active. The rates of the other impairments are usually low, the peaks occur where the "portal crane function" shows transitions from very close to far away. In Table 5.5 we show the overall number of lost packets for the two scrambling modes and all measurements, while in Table 5.6 for the **factorial** measurement the numbers of lost packets for the different modulation types are shown. For the scrambling mode we have the tendency that with scrambling we have fewer packet losses, and, not shown here, for the case with scrambling the packet losses are typically shorter than without scrambling. This holds for all measurements. For the modulation scheme we have no clear trend.

The occurence of packet losses allows to draw an important conclusion. As explained in the previous section, packet losses are due to failure of acquiring bit synchronization. This happens already in the PHY header. As a result, no MAC protocol can protect itself against packet losses by influencing the contents of the data part of a PHY packet. Instead it is necessary to incorporate other mechanisms, e.g., variation of transmit power level, using retransmission schemes, switching on scrambling, or better shielding the radio equipment. Furthermore, for using these mechanisms a feedback from the receiver is needed, i.e., the MAC protocol has to incorporate an immediate acknowledgement mechanism.

5.2 Bit Errors

As explained in the previous sections, bit errors are restricted to the data part of a PHY packet, since errors in the packets PHY header lead to loss of the whole packet. Before starting evaluation of bit errors, we have filtered out dropped packets, ghost packets, truncated and oversized packets from the traces. For the traces considered in the remainder this is reasonable

Modulation	MBER (w/o scrambling)	MBER (w/scrambling)
BPSK	2.5571e-05	0.0003
QPSK	7.4428e-05	0.0001
BMBOK	0.0018	0.0399
QMBOK	0.0544	0.0589

Table 5.7: Mean Bit Error Rates for different modulation types (**factorial** measurement)

since the respective rates are low. For the remaining packets we have generated the *packet bit indicator sequence* by `XOR`ing the received packet with the expected packet. We define the *trace indicator sequence* $i_1 i_2 \dots i_m$ to be the concatenation of all packet bit indicator sequences for all received packets of a single trace (in order of increasing packet numbers). The burst length sequence corresponding to $i_1 i_2 \dots i_m$ is denoted as $X_1 Y_1 Z_1 X_2 Y_2 Z_2 \dots X_p Y_p Z_p$ just as described in Section 3.1. Please note that in the trace indicator sequence any information about packet boundaries, lost packets, or packet gap times is completely ignored. However, for the purpose of evaluating the performance of coding schemes, these are not important. Furthermore, for a burst order $k_0 > 1$ we loose the information about the error positions within an error burst (see Section 3.1)

We exclude the 11 MBit/s CCK and 5.5 MBit/s CCK traces from further discussion, since these traces are extremely error prone. For example, in trace 52 (11 MBit/s CCK, 2016 bytes packet size, without scrambling) 19729 out of 20000 packets do not contain a single well-formed chunk and are dropped.[4] Many of the 5.5 MBit/s CCK traces look also disastrous (specifically with scrambling) and thus are also excluded. A first conclusion is that these modulation types are not suitable for use in industrial WLANs. In Table 5.7, the relationship between the modulation type and the mean bit error rate (after removing dropped packets) is shown. In addition, this table indicates that without scrambling we have much lower mean bit error rates. We restrict the discussion to the traces without scrambling.[5]

As a shorthand, we denote the BPSK and QPSK traces of the **factorial** measurement as the *remaining traces*.

[4] For many packets it was not possible to compute the error rate, since it was not possible to determine the corresponding expected packet. For those packets where we could guess the expected packet, bit error rates of 25% to 30% are easily reached.

[5] This restriction has two reasons. The first reason is the lack of space. The second reason lies in the assumption that in running systems scrambling indeed is disabled, in order to get lower error rates. Hence, the case without scrambling is more interesting for practice.

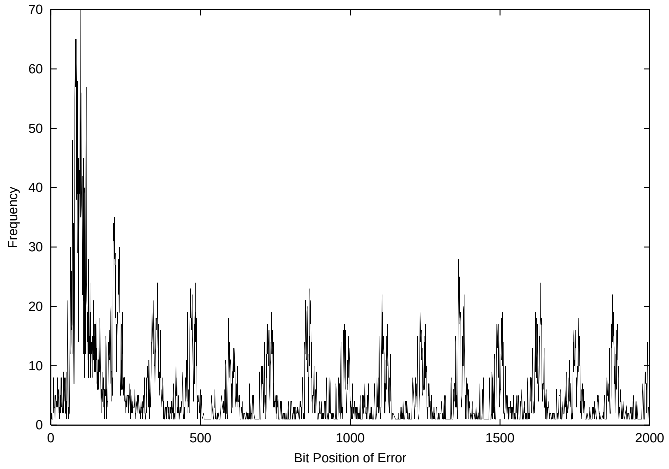

Figure 5.11: Positions of Bit Errors, **factorial** trace 83 (QPSK modulation)

5.2.1 Positions of Bit Errors

The first result is that bit errors do not occur in all positions of a frame with equal probability. This is exemplarily shown in Figure 5.11 for a QPSK trace (**factorial** measurement) and in Figure 5.12 for a 5.5 MBit/s CCK trace, where for the first 2000 bit positions within a frame the number of bit errors occuring at this position during a trace is displayed. These figures are representative for the patterns occuring for the respective modulation types.[6]

The first thing to note is that the relative beginning of a frame there is a peak. For QPSK and BPSK traces without scrambling it is frequently found between bit ≈ 200 and 250, for traces with scrambling often a peak at positions ≈ 80 to 100 is present.

Second, the figures for BPSK and QPSK traces show some periodicity. From inspection, for BPSK traces the basic period is 64 bits, for QPSK it is 128 bits. This periodicity occurs for the cases with scrambling and without scrambling, however, with scrambling the effect is more pronounced than without scrambling (where, nonetheless, the effect is clearly visible). Below we present some figures, which, for selected QPSK traces, show the conditional probability that bit $n+k$ is wrong given that bit n is wrong. These figures indicate that indeed often bit errors have a distance of 128 bits. We have no validated explanation for this phenomenon, but we think it is due to the bit synchronization algorithm used in the receiver.

In summary, the error patterns are sensitive to the modulation type. Furthermore, if stochastic models are built with the indicator sequence as the only input, the models are

[6] Provided that we are looking at those traces where the number of errors is sufficiently high to have results of statistical significance.

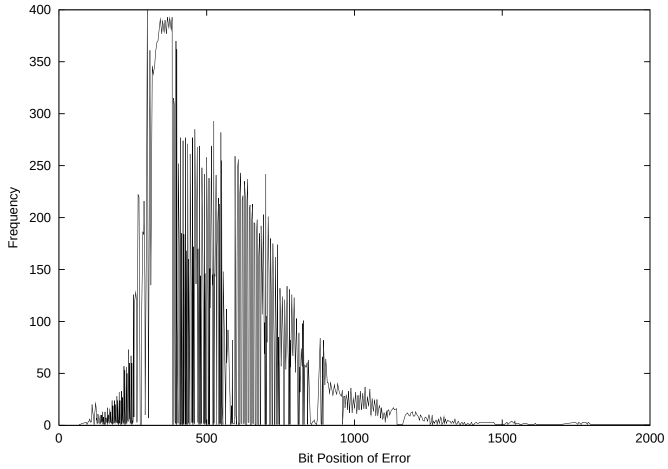

Figure 5.12: Positions of Bit Errors, **factorial** trace 37 (5.5 MBit/s CCK modulation)

sensitive to the frame size due to the peak at the beginning of frames.

5.2.2 Bit Error Statistics

The trace indicator sequence $i_1 i_2 \ldots i_m$ is a binary indicator sequence. Thus, given some burst order k_0 we can investigate error bursts and error free bursts, as defined in Section 3.1. First we discuss some summary statistics.

We exclude the 11 MBit/s CCK and 5.5 MBit/s CCK traces from further discussion, since these traces are extremely error prone. For example, in trace 52 (11 MBit/s CCK, 2016 bytes packet size, without scrambling) 19729 out of 20000 packets do not contain a single well-formed chunk.[7] Many of the 5.5 MBit/s CCK traces look also disastrous (specifically with scrambling) and thus are also excluded. A first conclusion is that these modulation types are too sensitive for use in industrial WLANs. As a shorthand, we denote the BPSK and QPSK traces of the **factorial** measurement as the *remaining traces.*

In Figure 5.13, we show the mean bit error rates (MBER) for the remaining traces. Despite the fact that the MBERs are in general higher when scrambling is enabled,[8] it can be observed that they vary over several orders of magnitude, even for the same modulation

[7] For many packets it was not possible to compute the error rate, since we could not determine the corresponding expected packet. For those packets where we could guess the expected packet, bit error rates of 25% to 30% are easily reached.

[8] This is true for both the **factorial** and **longterm1** measurements, both taken with the same modem set. For the **longterm2** measurement both scrambling modes show approximately the same mean bit error rate.

scheme and scrambling mode. In Table 5.7, the relationship between the modulation type and the mean bit error rate (after removing dropped packets) is shown. The BPSK modulation shows the best error rates, followed by the QPSK scheme.

In Figure 5.14, we show the mean error burst length vs. k_0 for the remaining BPSK traces without scrambling; in Figure 5.15, the same is shown for the remaining QPSK traces without scrambling. In general, clearly the mean error burst length increases for increasing k_0. However, it needs not to be true that error free burst lengths decrease with increasing burst order since the long bursts remain visible, while short bursts disappear. From both figures one can observe a trend to “step functions” with the steps having a distance of $\approx$ 64 for BPSK and a distance of $\approx$ 128 for QPSK. After inspection of several traces this behavior can be explained with the periodicity of bit errors described in Section 5.2.1. The figures for the remaining BPSK and QPSK traces with scrambling show the same behavior (period lengths) and are not displayed. In Figure 5.16, we show the coefficient of variation of the error burst length distribution vs. burst order k_0 for the remaining BPSK traces without scrambling, while in Figure 5.17, the same is shown for the QPSK traces without scrambling. Furthermore, in Figures 5.18 and 5.19 we show the coefficients of variation for the error free burst lengths vs k_0 for the remaining BPSK and QPSK traces without scrambling, respectively. It can be clearly seen that, in general, the error free burst lengths show a much greater variability than the error burst lengths, and that the 64/128 bit periodicity can also be observed. For $k_0 = 1$ the mean values of the error burst lengths are often very close to 1, which means that most error bursts are only one bit long. The usually small coefficients of variations (for $k_0 = 1$) supports this.

If we look at the data along the time axis (trace number), it can be observed that for fixed k_0 the mean error burst length, the mean error free burst length and the corresponding coefficients of variation vary substantially over time (especially, the mean error free burst lengths for fixed k_0 fluctuates over several orders of magnitude, not shown here). The plot of the mean bit error rates (Figure 5.13) is a good indication of this fact. Thus, the bit error characteristics vary not only on short timescales (from burst to burst, range of milliseconds), but also on larger timescales (trace order, range of minutes).

Feasibility of Block FEC Schemes

We investigate the feasibility of some simple block FEC schemes [16], where a block of k user bits is mapped onto n code bits, with $n > k$. Our main tool is the *Hamming Bound* [16, chap. 3], which states that up to t errors can be corrected in a codeword of n bits length and

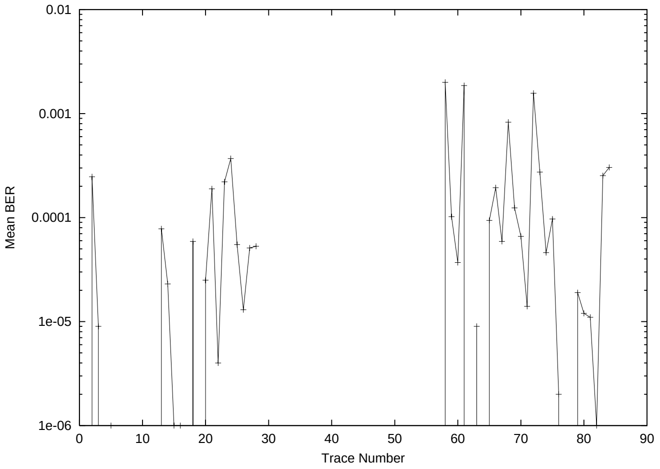

Figure 5.13: Mean bit error rate vs. trace number for remaining traces (logarithmic scale)

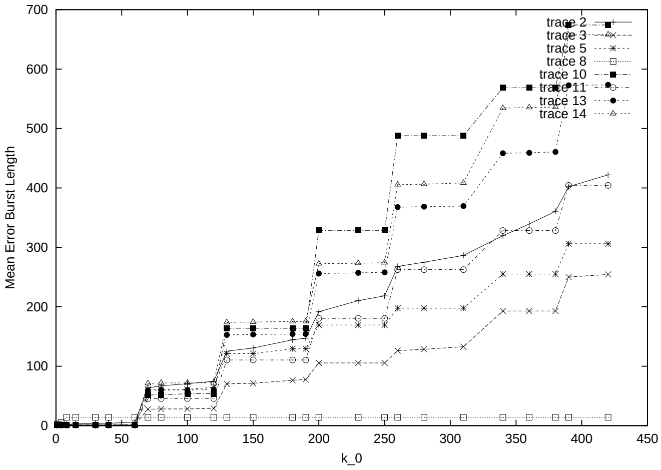

Figure 5.14: Mean error burst length vs. k_0 for remaining BPSK traces without scrambling

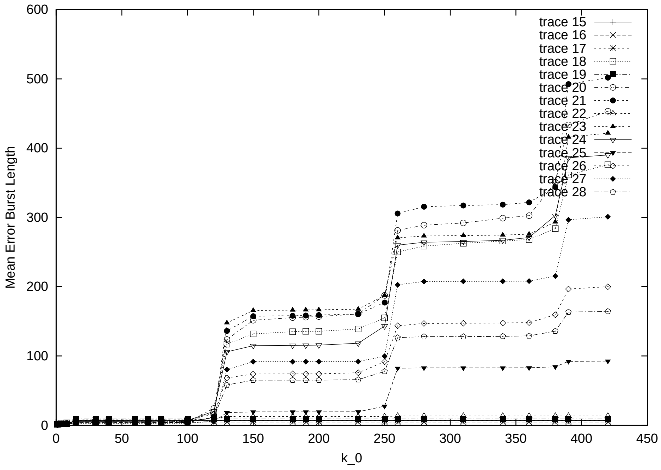

Figure 5.15: Mean error burst length vs. k_0 for remaining QPSK traces without scrambling

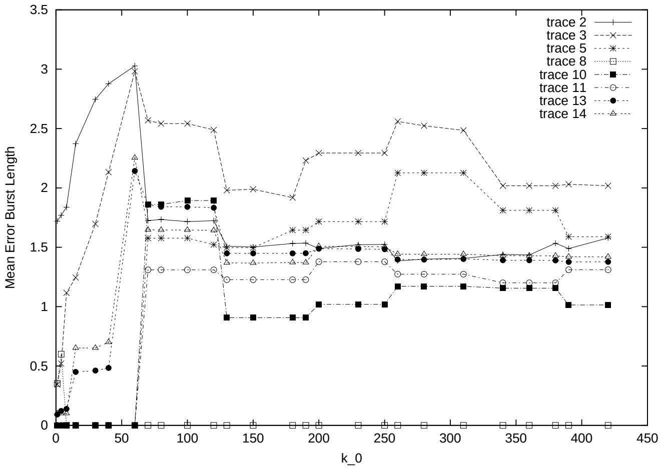

Figure 5.16: CoV of error burst length vs. k_0 for remaining BPSK traces without scrambling

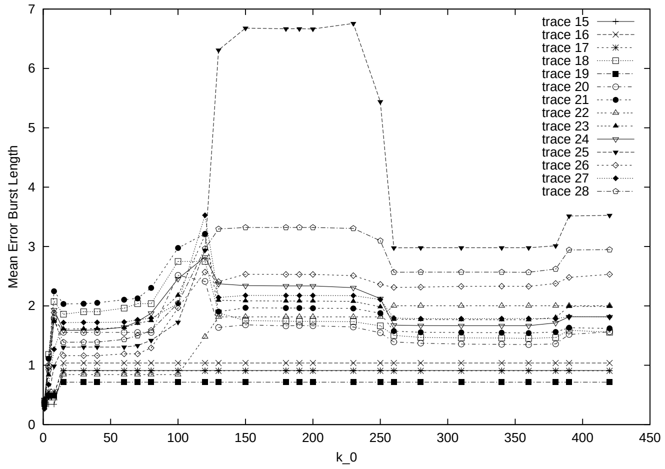

Figure 5.17: CoV of error burst length vs. k_0 for remaining QPSK traces without scrambling

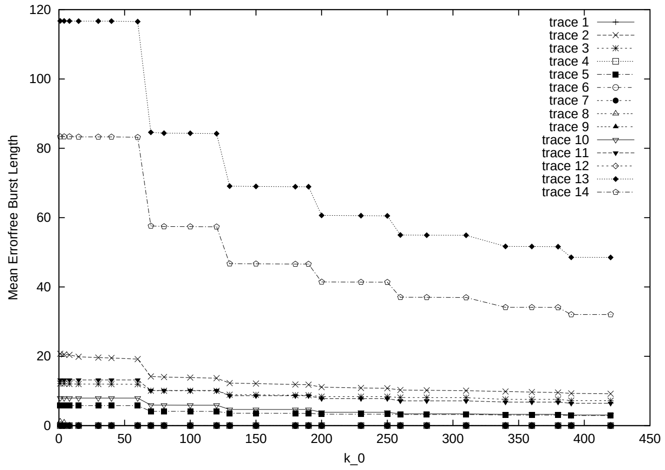

Figure 5.18: CoV of error free burst length vs. k_0 for remaining BPSK traces without scrambling

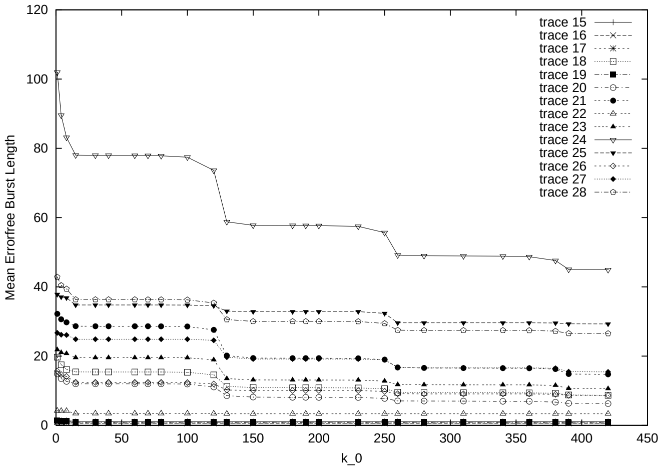

Figure 5.19: CoV of error free burst length vs. k_0 for remaining QPSK traces without scrambling

Modulation	len=1	len=2	len >2
BPSK/wo	95782	657	250
BPSK/w	987863	615	379
QPSK/wo	191740	65611	7979
QPSK/w	264891	123225	13833

Table 5.8: Burst lengths of error burst with density one for QPSK and BPSK traces

k user bits, only if the following relation holds:

$$2^{n-k} \geq \sum_{i=0}^{t} \binom{n}{i}$$

Please note that the fact that a triple (n, k, t) satisfies this relation, does not imply that a code with this properties really exists. The ratio $\frac{k}{n}$ is denoted as the *code rate*. In the following, we will restrict ourselves to $n \leq 32$. This choice is somewhat arbitrary, but can be justified by the observation that in industrial communications we have often very short packets (i.e. small k).

First we consider the remaining BPSK traces. We have for every trace built its trace indicator sequence, showing up the wrong and correct bits (see Chapter 3), and the associated burst length sequences for a given burst order k_0. If we denote the error free burst lengths as $X_1 \ldots X_p$, the error burst lengths as $Y_1 \ldots Y_p$ and the number of erroneous bits in the j-th error burst as $Z_1 \ldots Z_p$, we can calculate the *error density* of the j-th error burst as $\frac{Z_j}{Y_j}$. Instead

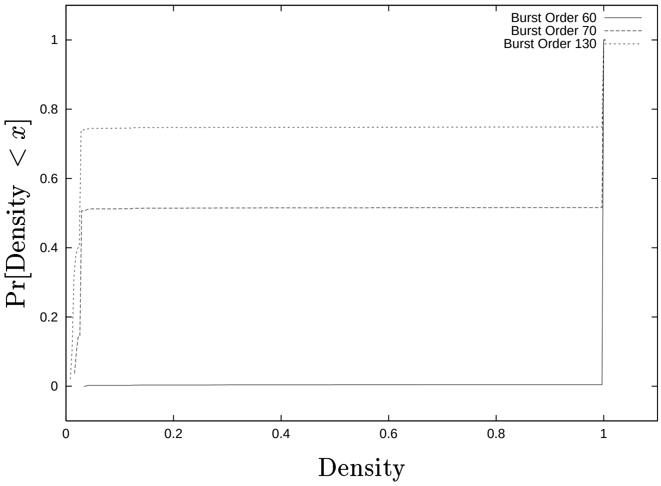

Figure 5.20: Cumulative Distribution Function for the error densities of all remaining BPSK traces

of looking at a single trace, we have joined the error densities of all remaining BPSK traces. In Figure 5.20 we show the cumulative distribution function $D(x) = \Pr[\frac{Z_i}{Y_i} < x]$ of all error densities for different burst orders. The curve for burst order $k_0 = 60$ is representative for all burst orders $k_0 < 64$,[9] accordingly, the curve for burst order $k_0 = 70$ matches very closely the curves for $64 < k_0 < 128$, and the curve for $k_0 = 130$ is representative for the curves with $k_0 > 128$. It can be seen that for the short burst order $k_0 = 60$ most of the mass is on density one, in almost all cases corresponding to single bit errors (see Table 5.8), while the remaining densities have comparably low frequencies. This means that in almost all cases a single bit error is surrounded by two error free bursts of at least 61 bits length. These errors can be corrected with a weak (i.e. high code rate) FEC code. The results shown for the larger burst orders k_0 do not contradict this finding, since the mass shifted from density 1 to densities of below 5% corresponds to those bursts, where by the 64 bit periodicity phenomenon a single burst consists of two (or very few) erroneous bits with distance of 64 bit.

However, it is questionable whether it makes sense to apply FEC to all packets. For all remaining BPSK traces without scrambling the packet error rate (PER, this includes all packets with at least one bit error) is below 5.5%. In contrast, by the hamming bound, the best code rate achievable for correcting $t = 1$ bit in $n \in \{8, \ldots 32\}$ bits is $\approx 84\%$ for $n = 31$, which means that at least an overhead of 16% is permanently present. For the remaining

[9] For the BPSK traces $k_0 = 64$ is a critical value because of the 64 bit periodicity described in Section 5.2.1. Correspondingly, for QPSK $k_0 = 128$ is a critical value.

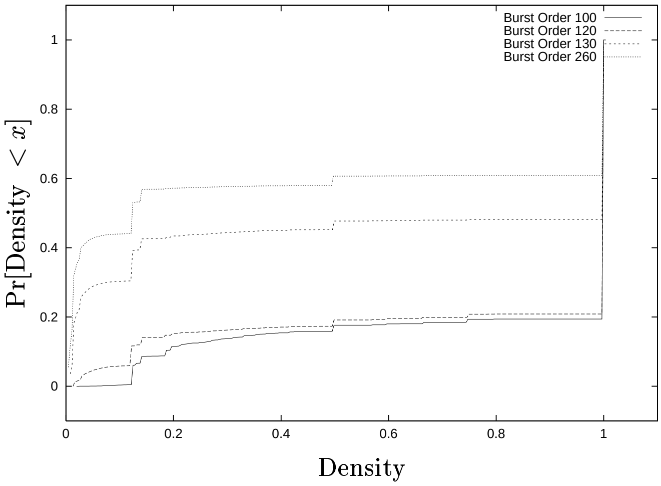

Figure 5.21: Cumulative Distribution Function for the error densities of all remaining QPSK traces

BPSK traces with scrambling some traces have PERs of up to 28%. Hence, using FEC makes more sense.

In Figure 5.20 we show the cumulative distribution function $D(x)$ for the remaining QPSK traces without scrambling and burst orders $k_0 = 120$ (below the critical value $k_0 = 128$), $k_0 = 130$ and $k_0 = 260$. Again, the curves are typical for the curves in their respective classes. Our interest is on those bursts with densities between 10% and 80%, which make up $\approx$ 20% of all bursts.[10] By inspection, 90% of these bursts have a length of 16 bits or fewer, with the dominant burst lengths being 4, 14 and 16 (the other lengths play no role). From the bursts of 14 or 16 bits length 94% have only two bit errors (at both ends of the burst). For correcting $t = 2$ bits in a block of size $n \in \{6, \ldots, 32\}$ the best code rate achievable is $\approx$ 71% for $n = 31$ (with max. $k = 22$). The maximum PER of all remaining QPSK traces without scrambling is $\approx$ 20%, while the mean PER is $\approx$ 7.1%. For the remaining QPSK traces with scrambling the maximum PER is $\approx$ 17%, while the mean PER is $\approx$ 4.3%. Hence, again the usage of FEC for all packets is questionable.

	trace 18 ($k_0 : 100$)	trace 18 ($k_0 : 150$)
MBER	0.000059	0.000059
mean EBL	5.770	131.788
CoV EBL	2.749	1.751
max. EBL	151	1920
mean EFBL	35472.570	69714.857
CoV EFBL	15.341	10.933

Table 5.9: Summary statisitis of trace 18 (EBL=error burst length, EFBL=errorfree burst length, MBER=mean bit error rate)

	trace 24 ($k_0 : 100$)	trace 24 ($k_0 : 150$)
MBER	0.000370	0.000370
mean EBL	6.873	114.807
CoV EBL	2.457	2.341
max. EBL	229	6529
mean EFBL	6353.049	11514.112
CoV EFBL	77.441	57.775

Table 5.10: Summary statisitis of trace 24 (EBL=error burst length, EFBL=errorfree burst length, MBER=mean bit error rate)

Investigation of Selected Traces

Finally, in order to present a more detailed view on the bit error behavior, we present some results for two "interesting" QPSK traces without scrambling and with different packet sizes.[11] Specifically we have chosen traces 18 (324 bytes packet size) and 24 (2016 bytes packet size). Some summary statistics of these traces are displayed in Tables 5.9 and 5.10. In both traces, the error burst lengths for $k_0 = 100$ and $k_0 = 150$ have approximately the same mean values and variability, while for the error free burst lengths we can observe that trace 18 has significantly higher mean values and smaller variability. In Figures 5.22 and 5.24 we show the cumulative distribution functions of the error free burst lengths for burst orders $k_0 = 100$ and $k_0 = 150$, respectively, while in Figures 5.23 and 5.25 we show the corresponding curves for the error burst lengths. For the error free burst lengths it can be seen that the distributions have a large range, i.e. very long error free burst lengths are present. However, since the measurement time was bounded, the number of datapoints constituting the tail is very low, and the distribution function lacks precision for high burst lengths. For the error burst lengths we can observe that there is significant mass on the shortest possible burst length 1, and that much mass is shifted away from 1 when increasing k_0 from 100 to 150.

In Figure 5.26, we show the conditional probability $\Pr[i_{n+k} = 1 | i_n = 1]$ which is, under certain assumptions, equivalent to the correlation function $\mathrm{Corr}[i_n, i_{n+k}]$ (see Section 3.1). The 128 bit periodicity is clearly visible, in addition, for short lags $k < 50$ weak correlation is present. This is typical for QPSK traces and missing for the BPSK traces of sufficiently high error rate. Despite the different error free burst length characteristics, both traces have the same correlation structure. If we look at the autocovariance function of the burst length sequences of the respective trace indicator sequences (shown in Figures 5.27 and 5.29 for the error free burst lengths and in Figures 5.28 and 5.30 for the error burst lengths) it can be seen that both traces show only weak correlation for error burst lengths for all lags $k > 0$, while for the error free burst lengths for short lags $k \leq 5$ we have substantial weak correlation. For most of the BPSK traces the error free burst lengths have correlation values close to zero for all lags $k > 0$ and the same correlation structure as the QPSK traces for the error burst

[10] The mass shift from density one to densities below 5% when switching from $k_0 = 120$ to $k_0 = 130$ can be explained by the 128 bit periodicity typical for QPSK

[11] The restriction to QPSK is due to several reasons. First, QPSK modulation is more interesting for applications because of the higher bitrate. Second, if one looks at the correlation function of the trace indicator sequence, this is pretty flat for most BPSK traces, except for the 64 bit periodicity. In contrast, the QPSK traces exhibit a richer correlation structure. And lastly. since QPSK shows more error events, the confidence in the bit error statistics is higher.

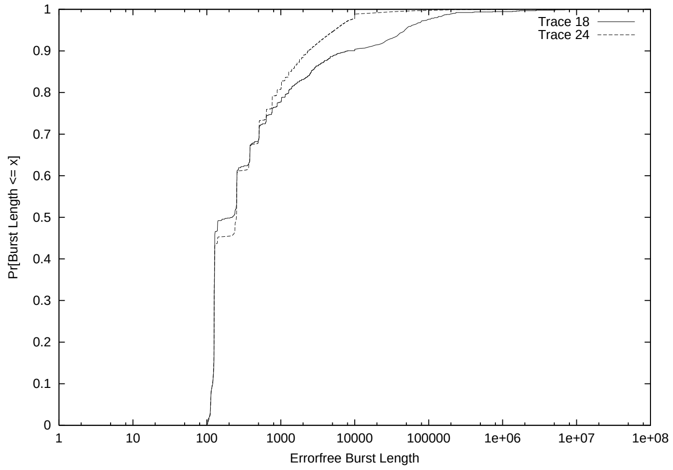

Figure 5.22: Cumulative Distribution Function for error free burst lengths (burst order $k_0 = 100$) for Traces 18 and 24

lengths.

5.3 Related Work

In this section some other measurement studies are summarized, with focus on packet level or bit level measurements. For lower level (wave propagation) measurements of indoor scenarios see, e.g., reference [3] for measurements on channel impulse response, and reference [2] for an overview of propagation measurements and models. In the following, we restrict ourselves to indoor measurements.

In a recent paper of Eckhardt and Steenkiste [7] adaptive error correction techniques are applied to WLAN traces, recorded in measurements using WaveLAN (902-928 MHz frequency band, 2 MBit/s QPSK modulation, receiver antenna diversity). They generate a specific UDP/IP packet stream, the underlying WaveLAN uses a CSMA/CA variant without retransmission on the MAC level. This stream is captured and stored by a special receiver station, even if the frame checksum generated by the WaveLAN MAC is wrong. The main findings are: a) bit errors are insensitive to the bit value; b) at short distances with no interferers the packet loss rate is zero and the packet error rate (PER, rate of packets with at least one bit error) is negligible, while with co-channel interferers the packet loss rates go up to 31%, a lot of truncated packets occur, and the PER is strongly varying. Almost all packets with corrupted bits have fewer than 5% of their bits corrupted. Bit errors do

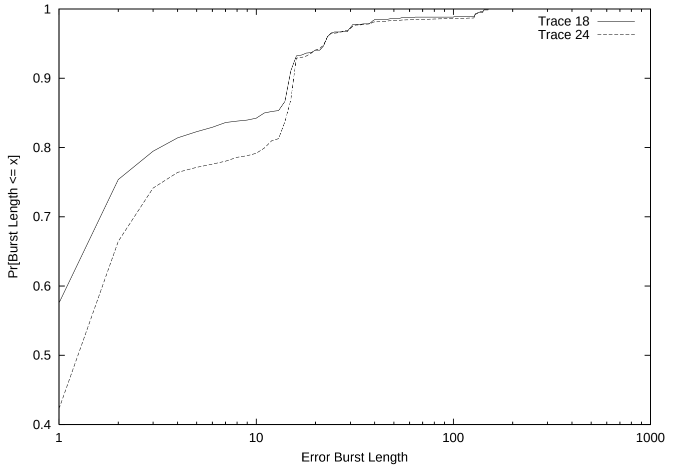

Figure 5.23: Cumulative Distribution Function for error burst lengths (burst order $k_0 = 100$) for Traces 18 and 24

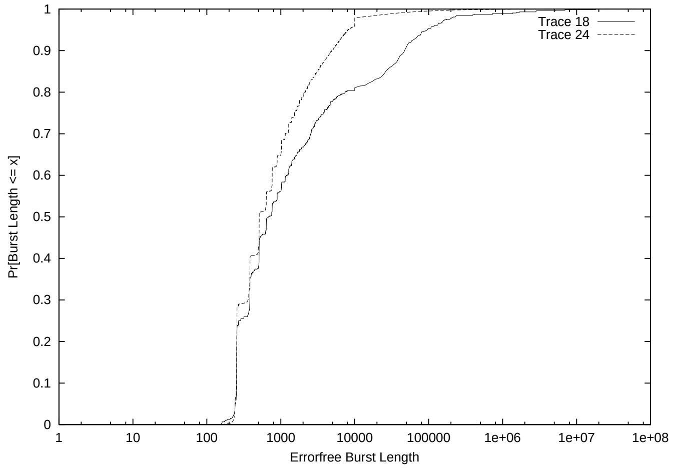

Figure 5.24: Cumulative Distribution Function for error free burst lengths (burst order $k_0 = 150$) for Traces 18 and 24

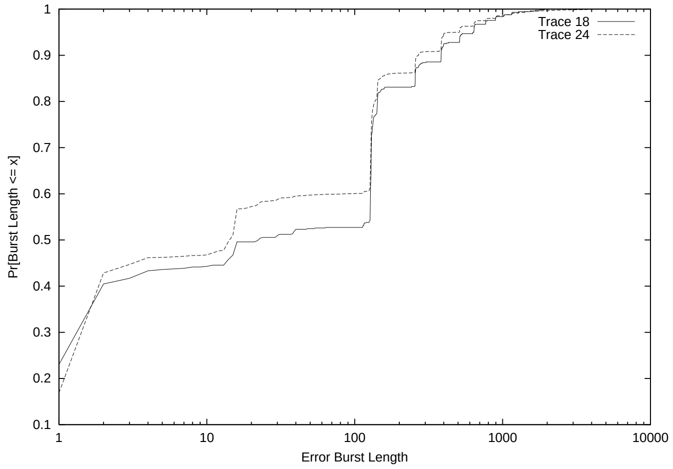

Figure 5.25: Cumulative Distribution Function for error burst lengths (burst order $k_0 = 150$) for Traces 18 and 24

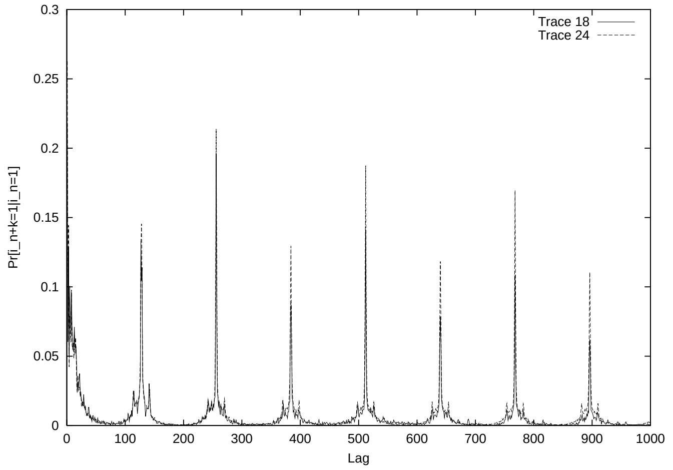

Figure 5.26: Conditional Error Probabilities for trace indicator sequences for Traces 18 and 24

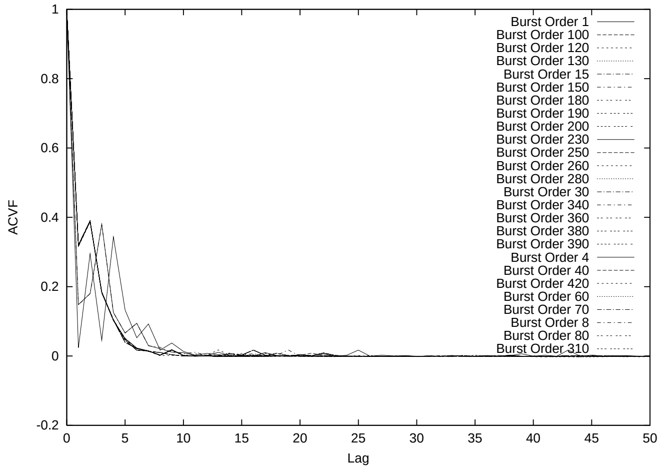

Figure 5.27: Autocovariance function for error free burst lengths and different burst orders k_0, Trace 18

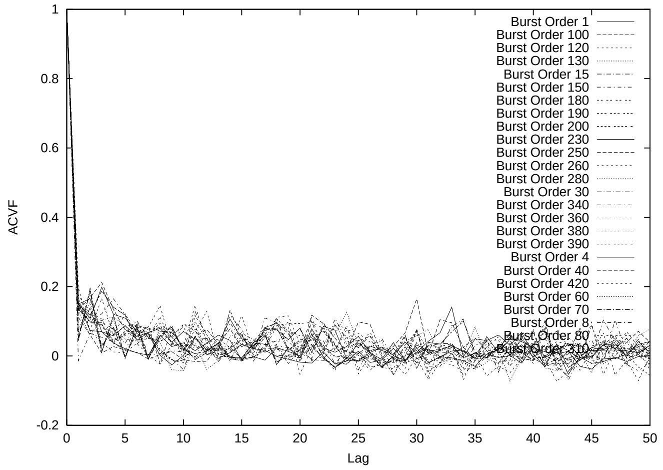

Figure 5.28: Autocovariance function for error burst lengths and different burst orders k_0, Trace 18

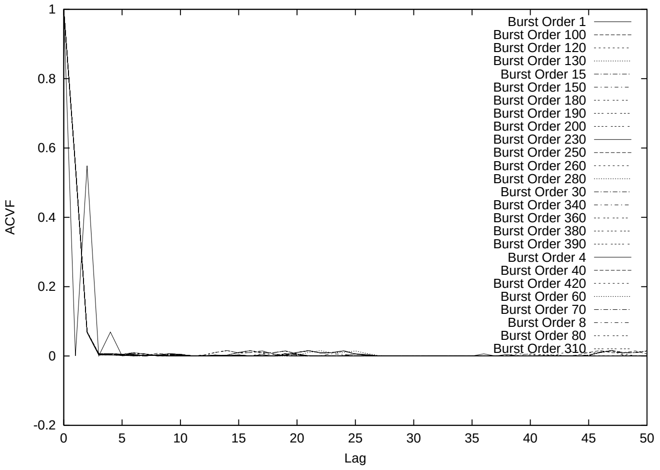

Figure 5.29: Autocovariance function for error free burst lengths and different burst orders k_0, Trace 24

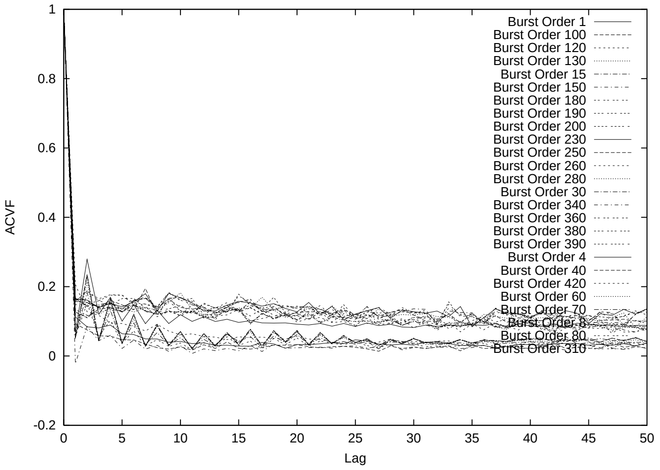

Figure 5.30: Autocovariance function for error burst lengths and different burst orders k_0, Trace 24

not have a trend to cluster in specific bit positions within a packet. Errors tend to occur in bursts, which are most often restricted to one or two bytes length (burst order $k_0 = 7$). The packet loss rate and bit error rate are insensitive to the packet size. The same authors have published another set of results on WaveLAN measurements before [6], using the same measurement setup. They have investigated signal quality parameters in an in-room line of sight scenario, a scenario with passive obstacles, and a third scenario with active interferers.

The work described in reference [20] is focused on tracing and modeling of wireless channel errors on a packet level, incorporating a full UDP/IP protocol stack over WaveLAN (902-928 MHz frequency band, DSSS, QPSK, 2 MBit/s). For each trace a transmitter sends a sequence of fixed-size UDP packets with a fixed generation rate to a receiver. All other interference and packet sources are suppressed. When only the load is varied, the PER rate does not change. When varying the packet size, the PER doubles with every 300 byte increase of packet size, reaching $\approx 10^{-3}$ for 1400 bytes. When only varying the distance, the PER doubles every 17 feet, up to ≈ 0.08 at 130 feet. They defined a binary indicator sequence by assigning a one for an erroneous packet and a zero for an error free packet. The mean error burst length was in most cases between two and three, while the mean error free length seems to decay almost linearly with increasing distance. The authors calculated suitable parameters for semi-Markov models for generating binary indicator sequences from their measurements. The choice of semi-Markov models was natural, since the error burst length distribution and error free burst length distribution are not geometric, as would be required for Markovian models.

One of the earliest WLAN packet-level studies is [5]. Again, a 902-928 MHz WaveLAN with 2 MBit/s QPSK, DSSS, and receiver antenna diversity was used. The authors have focused on varying the distance. For increasing distance the PER increases, however, there is a sharp cutoff, since it increases dramatically within a few meters, while before the increase rate was low. Applying a burst order of $k_0 = 7$ to their trace indicator sequence, they found that errors tend to be non-consecutive, typically only the minimum number of bits for constituting an error burst is erroneous (minimum error density). Furthermore, most packets have only a single error burst, which, in most cases, is short (length of 13 or 14). Interestingly, for longer runs the density function of the error burst lengths has peaks at multiples of 8, however, with decreasing amplitude. Hence, the authors also found some periodicity in the bit error behavior. The mean bit error rate can be regarded as constant over all packet sizes and distances.

Chapter 6

Stochastic Modeling

In discussing our measurement results, we have distinguished between packet losses and bit errors. This distinction makes sense for the designer of MAC protocols and coding schemes, since packet losses cannot be combatted with FEC schemes. Possible countermeasures include retransmissions, variation of the power level, changing the frequency channel or better shielding the radio equipment. In contrast, for combatting bit errors a wide spectrum of FEC techniques can be applied. After separating both issues, we can describe packet losses by means of (binary) indicator sequences, and we can do the same for bit errors. Hence, for stochastic channel modeling we are interested in methods for generating binary indicator sequences, matching some statistics of a given binary indicator sequence (describing packet losses or bit errors).

One of the most important goals of stochastic channel modeling is to find a simple stochastic process for generation of binary indicator sequences.[1] There are several requirements for such a process: a) it should have a low computational complexity; b) it can be easily implemented on a computer; c) its parameters can be computed from our measurement results; d) the models output should match certain channel statistics with sufficient accuracy; and e) preferrably it is conceptually simple.

In the following Section 6.1, we first give a brief overview of the most popular stochastic models for generating bit errors (or binary indicator sequences). Where appropriate, we explain how these models can be parameterized from our data. Then, in Section 6.2 we introduce an alternative model type for generating binary indicator sequences, called “bipartite” models. It is targeted to remove some of the other model’s limitations. In Section 6.3, we

[1] Other types of models, e.g., those directly targeting wave propagation aspects [23], are beyond the scope of this report.

propose an overall structure of a stochastic channel model, which uses explicit submodels for the phenomena of packet losses and bit errors, and combines these submodels in a unified framework. Finally, in Section 6.4 we compare for a simple example protocol the performance obtained when using different channel models and when using directly a trace from our measurements for generating errors. Our results show, that the selected performance parameter can vary substantially and that proper model selection is an issue. Furthermore, the results obtained for the bipartite model indicate that further work on approximating the underlying distribution functions is required.

6.1 Overview of common Stochastic Channel Models

We briefly present some of the most commonly used stochastic processes used for generating binary indicator sequences (in most cases interpreted as bit errors). The majority of these models use time-homogeneous Markov chains (discrete or continuous).

Assume that a binary indicator sequence $i_1 i_2 \dots i_m$ is given, and the associated burst length sequence is $X_1, Y_1, Z_1 \dots X_p, Y_p Z_p$. Let the mean error rate be $\bar{e}$, the mean error free burst length be $\bar{X}$ and the mean error burst length be $\bar{Y}$ (see Section 3.1).

The first and most simple model is the *independent model*, where one fixed bit error probability $p \in (0,1)$ is given, and, conceptually, for every bit a Bernoulli experiment is carried out, such that every experiment has the same parameter p and is independent of all other experiments. In order to match at least the mean value between model and given indicator sequence, clearly $p = \bar{e}$ must be chosen.

This model is quite simple to implement, however, does not capture the “bursty” nature of channel errors observed in low level measurements or predicted from propagation models. A quite common approach is to introduce some additional “channel state”.

6.1.1 Two-State Models

A very popular model is the two-state *Gilbert model* [12] or *Gilbert/Elliot model* [8]. It assumes a “good” channel state and a “bad” channel state. Within every state, bit errors occur according to the independent model with rates e_{g} and e_{b}, respectively ($e_{\mathrm{g}} \ll e_{\mathrm{b}}$). Conceptually, the next channel state is determined after every bit according to a discrete two-state time-homogeneous Markov chain with transition matrix

$$\mathbf{P} = \begin{pmatrix} p_{\mathrm{gg}} & 1 - p_{\mathrm{gg}} \\ 1 - p_{\mathrm{bb}} & p_{\mathrm{bb}} \end{pmatrix}$$

(with p_{xy} being the probability, that the current state is x and the next state is y). From the Markov property, the state holding times have a geometric distribution and are independent of each other. The steady state probability of being in the good state is given by

$$p_{\mathrm{g}} = \frac{1 - p_{\mathrm{bb}}}{2 - (p_{\mathrm{gg}} + p_{\mathrm{bb}})},$$

and the steady state probability of being in the bad state is given by

$$p_{\mathrm{g}} = \frac{1 - p_{\mathrm{gg}}}{2 - (p_{\mathrm{gg}} + p_{\mathrm{bb}})}.$$

The mean bit error rate can then be calculated with $\bar{e} = p_{\mathrm{g}} e_{\mathrm{g}} + p_{\mathrm{b}} e_{\mathrm{b}}$.

It is easy to see, that the Gilbert/Elliot model has short term correlation properties for bit errors (i.e. $\Pr[i_{n+k} = 1 | i_n = 1]$), but not for the error free burst length sequence or the error burst length sequences. The Gilbert model uses $e_{\mathrm{g}} = 0$ and $e_{\mathrm{b}} = 0.5$ while in the Gilbert/Elliot model these values can be chosen arbitrarily. To determine the matrix $\mathbf{P}$ it is sufficient to know the mean state holding times $\frac{1}{1-p_{\mathrm{gg}}}$ for the good state and $\frac{1}{1-p_{\mathrm{bb}}}$ for the bad state. In our setting it is natural to associate the "good" state with the error free bursts and the "bad" state with the error burst. Thus, we choose $e_{\mathrm{g}} = 0$, $e_{\mathrm{b}} = \frac{\sum_{i=1}^{p} Z_i}{\sum_{i=1}^{p} Y_i}$, $p_{\mathrm{gg}} = 1 - (\bar{X})^{-1}$ and $p_{\mathrm{bb}} = 1 - (\bar{Y}^{-1})$. It is then easy to see that the mean bit error rate $\bar{e}$ and the mean bit error rate generated by the model are the same. Clearly, the same holds for the mean error burst length and mean error free burst length.

If either the error free burst length or error burst length distribution is not geometric, it is appropriate to drop the Markov assumption and to use other distributions, which better match the first and second moments of the error free burst length and error burst length distributions. Candidate distributions are, e.g., the binomial or geometric distributions, or quantized versions of continuous distributions, e.g., the lognormal distribution [37]. We denote this class of models as *semi-Markovian models*. However, it is important to note that this type of models also has short-term correlation properties for the bit errors, but, since all burst lengths are independent, the model allows no correlation for the burst length sequences.

6.1.2 N-State Models

A popular model using an N-state Markov chain is described in [32]. In this model, the Markov chain is derived from modeling the instantaneous signal-to-noise ratio at the receiver (R-SNR) under Rayleigh fading. The range of possible R-SNR values is partitioned into intervals, each interval is associated with a state of the Markov chain and a bit error rate value. Since the R-SNR can be assumed to be time-continuous, transitions are only allowed between neighboring states, and thus the transition matrix has tridiagonal structure. The possibility to generate this model from a few simple physical parameters (e.g., mean R-SNR value, and doppler frequency) makes it attractive. However, since these values are not available from our measurements, we cannot parameterize this model.

In [10], a Markov model with N states is described, subdivided into two state classes, namely error free states (class A) and error states (class B). This class of models is called *Fritchman models*. If the system's current state is in class B, the transmitted symbols are erroneous with probability 1. In general, the possible state transitions are not restricted. An application of Fritchman models to measurements of a 142 MHz land mobile channel can be found in [27]. A similar model is described in [17], however, uses two matrices P and Q, where P is used every time the preceding channel symbol was in error, while Q is used otherwise. Here it is allowed to have bit error rates different from 0 and 1.

Kim and Li propose [14, 15] to use a *Markov modulated process* (MMP) for approximating the first and second order statistics of packet error rate measurements. They employ an N-state time-continuous Markov chain, for which the generator matrix $\mathbf{Q}$ is of circulant type (i.e., each row is the previous row, shifted by one element), and within each state i the channel has a packet service rate of γ_i (channels currently subject to errors have a lower service rate) [24]. As input data they use the service rate process $\{R_c(t)\}_{t\in\mathbb{R}}$, obtained from measurements [25]. They use the fact that the power spectral density function $R(t)$ of the MMP generated by $\mathbf{Q}$ and $(\gamma_0, \ldots, \gamma_{N-1})$ can be explictly represented by the eigenvalues of $\mathbf{Q}$. These values are chosen such that $R(t)$ matches the measured power spectral density function of $\{R_c(t)\}_{t\in\mathbb{R}}$ as closely as possible. Then $\mathbf{Q}$ can be constructed from the chosen eigenvalues. Although this approach has attractive features, it is not easily adapted to our methodology and notions.

Another class of models are the *Hidden Markov Models* (e.g. [11, 29, 31], an in-depth treatment can be found in [30]). The methodology proposed in [11], however, uses only one state for the error free bursts, and thus the error free burst lengths are a prioriy independent. Furthermore, the Hidden Markov Models lack a direct intuition between the channel behaviour and the underlying Markov chain. We will not discuss these models further.

6.2 Bipartite Channel Model

We introduce a special class of Markov models, called "bipartite models". This name stems from the fact that the corresponding Markov chain forms a bipartite graph. This model offers different advantages over the models discussed so far: a) the underlying distribution functions for the error burst lengths and the error free burst lengths can be approximated with the desired degree of accuracy (at the cost of increasing the memory needed for the model); and b) the model can express short term correlation for the burst length sequences. Furthermore, the model is conceptually related to our notion of binary indicator sequences and burst length sequences, its parameterization from our traces is straightforward, and it is intuitively appealing.

The bipartite model is a special class of Fritchman models [10], with the distinguishing features that the shape of the transition matrix is explicitly restricted to periodic ones, and that bit errors do not occur necessarily with probability 1 in bad states.

6.2.1 Model Description

The approach is to employ a number n_1 of "bad" states and n_2 of "good" states and to allow state transitions only from good states to bad states and vice versa (forming a bipartite graph). An example model with three good states and four bad states is shown in Figure 6.1 with all possible state transitions for the first good state and bad state respectively. When states are numbered $s_1, \ldots, s_{n_1}, s_{n_1+1}, \ldots s_{n_1+n_2}$, the transition matrix has the form:

$$\mathbf{P} = \begin{pmatrix} \mathbf{0} & \mathbf{Q}_1 \\ \mathbf{Q}_2 & \mathbf{0} \end{pmatrix}$$

where $\mathbf{Q}_1$ is an $n_1 \times n_2$ stochastic matrix[2] describing the state transitions from the bad states to the good states, while $\mathbf{Q}_2$ is an $n_2 \times n_1$ stochastic matrix for the transitions from the good states to the bad states.

The operation of this model is as follows: every state s_i is assigned a discrete random variable p_i with probability distribution $p_i(k) = \Pr[p_i = k]$ (with $k \in \mathbb{N}$) and associated distribution function $F_i(x) = \Pr[p_i \leq x]$. This random variable takes values on a finite interval on the natural numbers. When the system enters a specific good state s_ν, a random number is drawn from the distribution p_ν. This random number is then interpreted as the number of bits for which no errors occur. When the system enters a specific bad state s_μ, again a random number is generated according to p_μ, determining the error burst length in bits. For an error burst we make the assumption that at least at both ends an error occurs, in the remaining burst the bit errors occur independently with a fixed rate r_i.

In order to build a model from the traces we need to choose the numbers of states n_1 and n_2, the matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$, the probability distributions p_i and the bit error rates in the

[2] In a *stochastic matrix* all elements are nonnegative and the rows sum up to 1. An analogous definition holds for a *stochastic vector*.

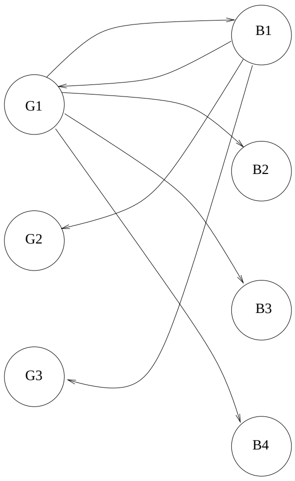

Figure 6.1: A sample bipartite Markov chain

bad states. Assume that for the error free burst lengths $X_1 \dots X_p$ the distribution function is $F_X(\cdot)$ and for the error burst lengths $Y_1 \dots Y_p$ the distribution function is $F_Y(\cdot)$. A simple approach, which linearly approximates the shape of $F_X(\cdot)$ and $F_Y(\cdot)$, can be summarized as follows:

- Select a number of good states and bad states.
- Partition the range of possible error free burst lengths into subintervals $[a_i, b_i)$ such that for every subinterval we have $F_X(b_i) - F_X(a_i) \approx \frac{1}{n_2}$. Do the same for the error free burst lengths. To every error free subinterval we assign one error free state, and to every error subinterval one error state. Hence, every state of the markov chain is associated with an interval.
- Construct the transition matrix $\mathbf{P}$ by simply counting for every state i the number of times it is left towards every possible target state j and divide this by the total number of times the system has left state i.
- Assign to every state i a uniform random variable p_i (this is the linearization) generating the burst lengths of the corresponding interval.
- We assume, for the error states that errors occur independently with a fixed rate. For every error state i, let $\Gamma_i \subset \{1, \dots, p\}$ denote the subset of all error bursts which belong to state i and use

$$r_i = \frac{\sum_{k \in \Gamma_i} Z_k}{\sum_{k \in \Gamma_i} Y_k}.$$

The procedure for constructing the transition matrix $\mathbf{P}$ is described in some more detail in [37]. Please note that the model allows to choose distributions other than uniform ones for the subintervals.

The accuracy of the model depends to a large extent on how good the distributions $F_X(\cdot)$ and $F_Y(\cdot)$ are approximated. Especially for the error free bursts with their long, but sparsely covered tail sometimes many states are needed. As an example, for trace 24 we have computed two models (used in Section 6.4), one with 27 good states, the other with 1084. For the error free bursts the "small" model generates a mean value of $\approx$ 1.6 million bits, while the "large model" has a mean value of $\approx$ 13800. The "true" value is $\approx$ 11500. For the error burst lengths with their considerable smaller tail the approximation was rather good, even for small numbers of states. Further effort is necessary to find better approximation schemes, which e.g. take special attention to those points where $F_X(\cdot)$ and $F_Y(\cdot)$ show jumps.

6.2.2 Main Model Characteristics

In this section we analyze the asymptotic behavior of the bipartite model and its correlation characteristics.

From the description given in the previous section it is easy to see that $\mathbf{P}$ generates a periodic Markov chain with period 2, and thus has no steady state. For the following we need the observation that for $k \in \mathbb{N}_0$ we have:

$$\mathbf{P}^{2k} = \begin{pmatrix} (\mathbf{Q}_1 \cdot \mathbf{Q}_2)^k & \mathbf{0} \\ \mathbf{0} & (\mathbf{Q}_2 \cdot \mathbf{Q}_1)^k \end{pmatrix}$$

and

$$\mathbf{P}^{2k+1} = \begin{pmatrix} \mathbf{0} & \mathbf{Q}_1 \cdot (\mathbf{Q}_2 \cdot \mathbf{Q}_1)^k \\ \mathbf{Q}_2 \cdot (\mathbf{Q}_1 \cdot \mathbf{Q}_2)^k & \mathbf{0} \end{pmatrix}$$

which can easily be proved by induction. We assume that the Markov chains generated by $\mathbf{Q}_1 \cdot \mathbf{Q}_2$ and $\mathbf{Q}_2 \cdot \mathbf{Q}_1$ are ergodic and thus the limits $\mathbf{A} := \lim_{k\to\infty}(\mathbf{Q}_1 \cdot \mathbf{Q}_2)^k$ and $\mathbf{B} := \lim_{k\to\infty}(\mathbf{Q}_2 \cdot \mathbf{Q}_1)^k$ exist (clearly, $\mathbf{A}$ and $\mathbf{B}$ are also stochastic matrices). The matrices $\mathbf{A}$ and $\mathbf{B}$ have the specific feature that in every row all elements have the same value [19, chap. 8]. Using this, and the fact that $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are stochastic matrices, it is easy to see that $A' := \mathbf{Q}_2 \cdot \mathbf{A}$ has the same number of rows as $\mathbf{A}$, and again all elements of a single row have the same value, namely the value that the corresponding row of $\mathbf{A}$ has. An analogous property holds for $B' := \mathbf{Q}_1 \cdot \mathbf{B}$.

Be $n := n_1 + n_2$, be ${\pi_0}^T \in \mathbb{R}^n$ with $\pi_0 = (\pi_1^0, \ldots, \pi_n^0)$ a stochastic vector describing the initial state distribution, let $z_0 z_1 z_2 z_3 \ldots$ be a sample path of the Markov chain (i.e.

$z_j \in \{s_1, \dots, s_n\}$), and $\mathbf{e}_1, \dots, \mathbf{e}_n$ denote the unit vectors of the $\mathbb{R}^n$. For simplicity we take the numbers 1 up to n for the states s_1 to s_n. Furthermore, denote $\mathbf{1}_S(x)$ the indicator function of the set S, i.e. $\mathbf{1}_S(x) = 1$ if $x \in S$ and 0 otherwise. Define the vector $\mathbf{a}_k$ by

$$\mathbf{a}_k = \mathbf{e}_{z_0} + \sum_{i=1}^{k} \sum_{j=1}^{n} \mathbf{1}_{\{s_j\}}(z_i)\mathbf{e}_j = \mathbf{e}_{z_0} + \sum_{j=1}^{n} \mathbf{e}_j \sum_{i=1}^{k} \mathbf{1}_{\{s_j\}}(z_i)$$

i.e. it counts in coordinate j of $\mathbf{a}_k$ how often the system was in state j during the first k state transitions of the given sample path. We define $\mathbf{a}'_k := \frac{1}{k}\mathbf{a}_k$ and interpret the j-th coordinate of $\mathbf{a}'_k$ as the fraction of the number of visits in state j w.r.t. to the total number k of state transitions so far. We write the j-th coordinate of a vector $\mathbf{a}$ as $[\mathbf{a}]_j$, and for a matrix $\mathbf{O}$ the matrix element on the i-th column and j-th row is written as $[[\mathbf{O}]]_{i,j}$. The long term fraction of visits in state j (exemplarily we choose $j = 1$) is then defined by:

$$a_1 := \lim_{k \to \infty} \left[\mathbf{a}'_k\right]_1$$

provided the limit exists. Be $\nu \in \mathbb{N}$, for $1 < 2\nu < k$ we can write:

$$\begin{aligned} \left[\mathbf{a}'_k\right]_1 &= \frac{1}{k}\left(\left[\mathbf{e}_{z_0}\right]_1 + \sum_{i=1}^{2\nu-1} \mathbf{1}_{\{s_1\}}(z_i)\right) \\ &\quad + \frac{1}{k}\left(\sum_{i=2\nu}^{k+2\nu} \mathbf{1}_{\{s_1\}}(z_i)\right) \\ &\quad - \frac{1}{k}\left(\sum_{i=k+1}^{k+2\nu} \mathbf{1}_{\{s_1\}}(z_i)\right) \end{aligned}$$

For k large enough we can drop the first and last term. Without loss of generality we assume k to be even, $k = 2l$ and $l > 2\nu$. Then we have

$$\left[\mathbf{a}'_k\right]_1 = \frac{1}{2l}\left(\sum_{i=0}^{l} \mathbf{1}_{\{s_1\}}(z_{2\nu+2i}) + \sum_{i=0}^{l} \mathbf{1}_{\{s_1\}}(z_{2\nu+2i+1})\right) \tag{6.1}$$

The terms in the first sum are independent random variables, the same holds for the second sum. We consider the first sum, denoted as $S_l = \sum_{i=0}^{l} \mathbf{1}_{\{s_1\}}(z_{2\nu+2i})$. For every random variable $\mathbf{1}_{\{s_1\}}(z_{2\nu+2i})$ we have that

$$\begin{gathered} \Pr[\mathbf{1}_{\{s_1\}}(z_{2\nu+2i}) = 1] = \left[\pi_0 \cdot \mathbf{P}^{2\nu+2i}\right]_1 = \\ \left[\pi_0 \cdot \begin{pmatrix} (\mathbf{Q}_1 \cdot \mathbf{Q}_2)^{\nu+i} & \mathbf{0} \\ \mathbf{0} & (\mathbf{Q}_2 \cdot \mathbf{Q}_1)^{\nu+i} \end{pmatrix}\right]_1 \end{gathered}$$

An analogous equation holds for the second sum. Since we have assumed the existence of $\mathbf{A}$ and $\mathbf{B}$, for ν large enough we can replace $(\mathbf{Q}_1 \cdot \mathbf{Q}_2)^{\nu+i}$ by $\mathbf{A}$ and $(\mathbf{Q}_2 \cdot \mathbf{Q}_1)^{\nu+i}$ by $\mathbf{B}$. Now each random variable $\mathbf{1}_{\{s_1\}}(z_{2\nu+2i})$ is an independent Bernoulli random variable with fixed probability

$$c := \Pr[\mathbf{1}_{\{s_1\}}(z_{2\nu+2i}) = 1] = \left[\pi_0 \cdot \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix}\right]_1$$

Thus the first sum of equation 6.1 is a sum of independent Bernoulli random variables. The strong law of large numbers [9, chap. 8] asserts that for $l \to \infty$ the random variable $\frac{S_l}{l}$ converges to $\mathrm{E}\left[\mathbf{1}_{\{s_1\}}(z_{2\nu+2i})\right] = c$ with probability one, i.e., for almost all sample paths. The same calculations can be done for the second sum of equation 6.1. Putting this together, it is shown that

$$a_1 = \lim_{k\to\infty} \left[\mathbf{a}'_k\right]_1 \xrightarrow{a.s.}$$
$$\frac{1}{2}\left(\left[\pi_0 \cdot \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix}\right]_1 + \left[\pi_0 \cdot \begin{pmatrix} \mathbf{0} & \mathbf{B}' \\ \mathbf{A}' & \mathbf{0} \end{pmatrix}\right]_1\right)$$

If we have the values a_1 up to a_n it is straightforward to compute e.g. the mean bit error rate, the mean error burst length or the mean error free burst lengths from the process generated by the bipartite model. If $P_i := \mathrm{E}[p_i]$ is the mean state holding time of state i, then the long term fraction of time the system is in state i is given by $a_i P_i$, and the mean bit error rate is given by

$$\bar{m} = \frac{\sum_{i=1}^{n} r_i a_i P_i}{\sum_{i=1}^{n} a_i P_i}$$

However, a drawback of the bipartite model is that these mean values cannot be represented in a “simple” and “intuitive” manner from the values of $\mathbf{P}$, as is the case for the Gilbert/Elliot model. Furthermore, since $F_X(\cdot)$ and $F_Y(\cdot)$ are only approximated, the resulting mean values for the burst lengths may differ. The distribution function of the generated error burst lengths Y' in step k can be calculated with the law of total probability:

$$\begin{aligned}
&\Pr[Y' \le y | z_k \in \{1, \ldots, n_1\}] = \\
&\quad \frac{\Pr[Y' \le y \cap (\{z_k = 1\} \cup \ldots \cup \{z_k = n_1\})]}{\Pr[\{z_k = 1\} \cup \ldots \cup \{z_k = n_1\}]} \\
&= \frac{\Pr[Y' \le y \cap \{z_k = 1\}] + \ldots + \Pr[Y' \le y \cap \{z_k = n_1\}]}{\Pr[z_k = 1] + \ldots + \Pr[z_k = n_1]} \\
&= \frac{\Pr[Y' \le y | z_k = 1]\Pr[z_k = 1] + \ldots + \Pr[Y' \le y | z_k = n_1]\Pr[z_k = n_1]}{\Pr[z_k = 1] + \ldots + \Pr[z_k = n_1]} \\
&= \frac{F_1(y)\Pr[z_k = 1] + \ldots + F_{n_1}(y)\Pr[z_k = n_1]}{\Pr[z_k = 1] + \ldots + \Pr[z_k = n_1]}
\end{aligned}$$

A similar calculation can be carried out for the error free bursts.

In order to show that the state process generated by $\mathbf{P}$ has short term but no long term correlation properties, we assume that $\mathbf{P}$ is diagonalizable, i.e. there exists two $n \times n$ matrices $\mathbf{R}$ and $\mathbf{D}$ such that $\mathbf{R}^{-1}$ exists and $\mathbf{D}$ is a diagonal matrix with the eigenvalues λ_i of $\mathbf{P}$ on the diagonale. Since $\mathbf{P}$ is a stochastic matrix, for all eigenvalues $|\lambda_i| \leq 1$ holds [26, chapter 1.6]. Under these assumptions we have that $\mathbf{P} = \mathbf{R}^{-1} \cdot \mathbf{D} \cdot \mathbf{R}$. Then the autocorrelation function of the generated process can be represented by

$$\begin{aligned} R(k) &= \mathrm{E}[z_0 z_k] = \sum_{i=1}^{n} \sum_{j=1}^{n} ij \,\Pr[z_0 = i, z_k = j] \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} ij \,\Pr[z_0 = i] \Pr[z_k = j | z_0 = i] \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} ij \,[\pi_0]_i \left[e_i \cdot \mathbf{P}^k \right]_j \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} ij \,[\pi_0]_i \left[e_i \cdot \mathbf{R}^{-1} \cdot \mathbf{D}^k \cdot \mathbf{R} \right]_j \end{aligned}$$

The influence of the eigenvalues λ_i with $|\lambda_i| < 1$ play a role only for small k and vanish for $k \to \infty$.

For calculating the long term correlation we first observe that for the case of k even and large enough we have

$$\begin{aligned} \mathrm{E}[z_0 z_k] &= \sum_{i=1}^{n} \sum_{j=1}^{n} ij \,[\pi_0]_i \left[e_i \cdot \mathbf{P}^k \right]_j \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} ij \,[\pi_0]_i \left[e_i \cdot \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix} \right]_j \\ &= \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} ij \,[\pi_0]_i \,[[\mathbf{A}]]_{1,j} \\ &\quad + \sum_{i=n_1+1}^{n} \sum_{j=n_1+1}^{n} ij \,[\pi_0]_i \,[[\mathbf{B}]]_{1,j-n_1}. \end{aligned}$$

In the last equation we have used that for each row $\mathbf{A}$ and $\mathbf{B}$ all elements have the same value. On the other hand we have

$$
\begin{aligned}
\mathrm{E}[z_0]\,\mathrm{E}[z_k] &= \left(\sum_{i=1}^{n} i\,[\pi_0]_i\right) \cdot \left(\sum_{j=1}^{n} j \left[\pi_0 \cdot \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix}\right]_j\right) \\
&= \left(\sum_{i=1}^{n} i\,[\pi_0]_i\right) \cdot \\
&\quad \left(\sum_{j=1}^{n_1} j[[\mathbf{A}]]_{1,j} \sum_{l=1}^{n_1} [\pi_0]_l \right. \\
&\quad \left. + \sum_{j=n_1+1}^{n} j[[\mathbf{B}]]_{1,j-n_1} \sum_{l=n_1+1}^{n} [\pi_0]_l \right)
\end{aligned}
$$

Using both equations, it is straightforward to verify that for all possible system start states $\pi_0 = \mathbf{e}_\nu$ with $\nu \in \{1, \ldots, n\}$ the relation $\mathrm{E}[z_0 z_k] = \mathrm{E}[z_0]\,\mathrm{E}[z_k]$ holds and thus there is no correlation. Furthermore, this calculation goes the same way for k odd. Hence, there is no long term correlation.

6.3 Overall Channel Model Structure

As basic cornerstone for a stochastic model we use the observations that, on the one hand, there are the packet-related phenomena, specifically packet losses, and on the other hand there are bit errors. For simplicity, we make the following additional assumptions:

- There are only packet losses; all other packet impairments are treated as such.
- Packet losses and bit errors are statistically independent from each other.[3]
- The bit error models depend on the modulation type (QPSK, BPSK) but not on the packet size.[4]

We propose to compose the channel model of three different submodels: the *packet loss submodel* generates a binary indicator sequence, which determines for every packet handed to the channel model, whether it should get lost or not. If the packet is not lost, then one of the two remaining *bit error submodels* is applied to the packet. If the packet should be transmitted with BPSK, then the *BPSK submodel* is applied, otherwise the *QPSK submodel* is applied. In principle, for each submodel one of the models cited in Section 6.1 can be instantiated. In

[3] Our data shows no evidence for dependencies or their absence.

[4] This contradicts our findings in Section 5.2.1, but leads to simpler models.

order to keep things simple, we assume that the BPSK and QPSK submodels are independent of each other.[5]

6.4 A Comparison of Different Models

In this section we give a brief demonstration of how the choice of error models affects the performance of a selected system.

For this purpose we have simulated a simple scenario: one transmitter and one receiver station are connected via a wireless link. The transmitter wishes to transfer a file of 1 GB size to the receiver. To this end the transmitter uses a simple Send and Wait protocol, there are no further stations present. We do not consider any MAC protocol or propagation delay. The transmitter splits the file into data frames of 1000 bits size (protocol overhead is neglected). He sends a data frame and waits for an acknowledgement. If this does not arrive within two bit times the packet is repeated, otherwise the next packet is transmitted. Data packets can be subject to errors, acknowledgements are always transmitted error free and have negligible size. The receiver only acks a frame, if it contains no errors. The number or retransmissions is not bounded. The transmission rate was 2 MBit/s (according to QPSK modulation). The performance figure of interest is the time needed to transfer the file over the channel.

For modeling the wireless channel we have chosen to use trace 24 of the **factorial** measurement as the basis, analyzed with burst order $k_0 = 150$ (the basic statistics are shown in Table 5.10, the packet length was 2016 bytes). We have used the following error models: the independent model (with $p = 0.0003707$), the Gilbert/Elliot model (with $p_{\text{gg}} = 1 - \frac{1}{11514.1}$, $p_{\text{bb}} = 1 - \frac{1}{114.8}$, $e_{\text{g}} = 0$ and $e_{\text{b}} = 0.0375$), a semi-Markov model (based on lognormal distributions for both error burst lengths and error free burst lengths), a second semi-Markov model with uniform distributions, a null model with no bit errors, one bipartite model with 27 good states and 17 bad states (a rather coarse approximation of the underlying distribution functions) and a bipartite model with 1084 good states and 310 bad states, and, finally, we have used the trace itself. We have not taken packet losses into account. The performance measure of interest is the time necessary to transmit the whole file, i.e. from sending the first packet until receiving the last acknowledgement. With the exception of the null model and the trace, every simulation was performed 40 times with different seeds of the pseudo random number generator.

The mean values reported in Table 6.1 show that the two state models predict the correct

[5] If this assumption is not made, and both submodels keep some channel state variable, we have the problem to decide which channel state should be entered when switching between the two submodels.

Model	Mean Time	Variance
Trace	5915.63 s	0
independent	8028.61 s	0.47
Gilbert/Elliot	6100.30 s	0.65
Semi-Markov (lognormal)	5803.03 s	137.41
Semi-Markov (uniform)	6114.74 s	0.09
Null model	5540.00 s	0
Bipartite, fine	5610.42 s	21.81
Bipartite, coarse	5570.45 s	0.02

Table 6.1: Transmission Times for 1 GB data over channels with different error models

result with only 3.1% (Gilbert/Elliot) or 1.9% (Semi-Markov) error, while the independent model generates an error of 35.7%. In this case, the bursty nature of the errors generated by the two state models captures the “real” link behaviour, since, as in the trace, during the error free periods the system can work without retransmissions, while for independent errors every packet has the same (high) probability of being erroneous and there are no periods with good conditions. What is even more interesting is the close match of the Gilbert/Elliot and the Semi-Markov model. In the Gilbert/Elliot model the error free burst length distribution is geometric, with a coefficient of variation of $\frac{1}{\sqrt{p_{gg}}}$, which in our case is close to 1 (the same holds for the error burst lengths). In contrast, the (quantized) lognormal distribution chosen in the Semi-Markov model match the coefficients of variation of the trace very closely (see Table 5.10). Hence, for the total transmission time in the sample system the variability of the burst lengths is not the dominant factor.

Unfortunately, despite the fact that the bipartite models use more memory, their predictions do not match as closely as the simple two state models. We attribute this to the problem of properly matching the distribution function of the error free burst lengths. From this result we conclude that further effort on finding better approximation schemes is required.

What is even more interesting is the close match of the Gilbert/Elliot and the Semi-Markov model. In the Gilbert/Elliot model the error free burst length distribution is geometric, with a coefficient of variation of $\frac{1}{\sqrt{p_{gg}}}$, which in our case is close to 1 (the same holds for the error burst lengths). In contrast, the (quantized) lognormal distribution chosen in the Semi-Markov model match the coefficients of variation of the trace very close (see Table 5.10). Hence, for the total transmission time in the sample system the variability of the burst lengths is not important.

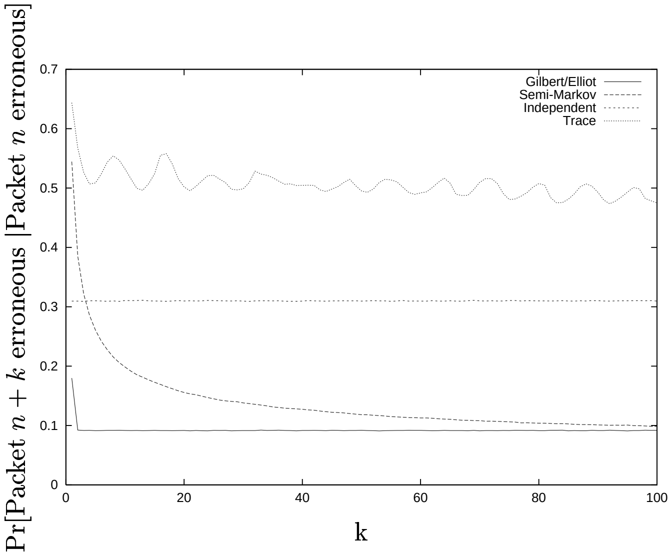

Figure 6.2: Conditional Probability that packet $n+k$ is erroneous given that packet n is erroneous

A general weakness of the two state models is that the sequence of generated error burst lengths is independent, and thus uncorrelated (the same holds for the error free burst lengths). However, in our data for the error free burst lengths we often have observed strong correlation for small lags and weak correlation for larger lags. For the error burst lengths we often have weak correlation for a longer time. Furthermore, for the Gilbert/Elliot model we have the problem that the variability of the burst length distribution of the traces is often much higher than that of the geometric distributions used by the model (typically close to 1). Especially for the error free burst lengths coefficients of variations of 20 up to 100 are typical (see Figures 5.18 and 5.19).

In order to show a counterexample, where the two state models fail to do good predictions, we have modified our simulator such that it puts out a 0 for a correctly received packet and a 1 for an erroneous packet. For this sequence we have computed the conditional probability that packet $n+k$ is erroneous, given that packet n is erroneous. The results are displayed in Figure 6.2. It can be seen that the Gilbert/Elliot model largely underestimates the correlation of the packet error behavior, whereas the independent model delivers better estimates. This can be interesting for transmission schemes, where FEC coding is enabled or disabled in response to the packet error rate over short time windows, and where the correlation information is used to decide how long FEC should stay enabled. Here we will likely get different results for the Gilbert/Elliot model and the trace.

This can have effects on performance predictions for simple retransmission schemes. As

Model	**Mean Time**
Trace	5915.63 s
Trace with FEC	5876.76 s
Gilbert/Elliot	6100.30 s
Gilbert/Elliot with FEC	6171.87 s
Semi-Markov	6127.22 s
Semi-Markov with FEC	6240.17 s

Table 6.2: Transmission Times for 1 GB data over channels with different error models with and without FEC for retransmissions

an example, we have extended our simulator with a (n, k) block FEC scheme and a simple rule: normal frames are transmitted with 1000 bits of user data, while retransmitted frames only have $1000 \cdot \frac{k}{n}$ bits of user data and the remainder is used for redundancy. Specifically, we have chosen $n = 31$ and $k = 22$, with the capability of correcting $t = 2$ errors within n bits. The results displayed in Table 6.2 show the following: if one uses a two state model for predicting the performance of the scheme with FEC retransmissions, he will see a loss in performance (although small) and thus refrain from using such schemes. However, when using the real trace, he will see (small) gain.

Chapter 7

Discussion

For the further development of industrial WLANs we can draw some conclusions from our measurement results. The first and somewhat disappointing result is that the 5.5 MBit/s CCK and 11 MBit/s CCK modulation schemes should not be considered for use, since they show really bad results. Second, our measurements indicate that scrambling should be disabled, since we have lower overall error rates without scrambling.[1] Third, the BPSK modulation scheme shows fewer bit errors than QPSK modulation (which is not surprising), but also the characteristics are different. BPSK tends to produce independent or "flat" errors (except from the periodicity observed), while in QPSK the errors are more clustered and show more short term correlation. Furthermore QPSK has higher error densities. For both modulations the error free burst lengths show great variability and a long tail, furthermore they exhibit short term correlation. The error burst lengths are much less variable and seem to be uncorrelated. Fourth, the bit error probability depends on the position in the packet. The fifth result is that MAC designers not only have bit errors to take into account, but also the loss or impairment of whole packets. Especially packet losses tend to occur in (sometimes very long) bursts. We see the need to explore ways to combat packet losses. Possible approaches include the variation of transmission power, using retransmissions, switching the frequency, increasing preamble lengths, and to better shield the radio equipment. Another general rule we can get from our results is that MAC and link layer protocols and coding schemes should incorporate some adaptability, since the channel shows up to be very time varying, both in terms of mean bit error rates and packet loss rates.

[1] This result is clearly visible for one modem set, for the other one both scrambling modes give approximately the same results.

7.1 Impact on Design of MAC and Coding Schemes

There are some things we can learn from our measurements. The presence of both packet losses and bit errors requires that any MAC or data link protocol aiming at reliable transmission cannot fully rely on FEC schemes, but has to somehow combine FEC and retransmission schemes.

The applicability of practical FEC schemes, i.e. those with a not too low code rate, to all packets is questionable. For BPSK with its “flat” errors a FEC scheme with high code rate (low protection) can do good work on saving retransmissions because of bit errors. On the other hand, the statistics cited in Section 5.2.2 state that the overall number of erroneous frames is smaller than the overhead required for some simple block FEC schemes.

For QPSK we also doubt, whether it makes sense to use the simple FEC schemes (see Section 5.2.2) for all packets. Our findings in Section 6.4, however, indicate that with restriction of block FEC schemes to retransmissions a small gain can be made.

7.2 Modeling of Wireless Links

The popular stochastic models used in the literature (independent model, Gilbert/Elliot model, Wang/Moayeri model) fail on different issues. First, they are pure bit error models and do not take the effect of packet losses into account. Second, the generated error burst lengths and error free burst lengths do not capture the corresponding distributions seen in the traces. Especially the failure of capturing the great variability of the error free burst lengths is a major drawback. Furthermore, these models generate error free burst lengths which are independent of each other and show no correlation.

The bipartite model offers opportunities to overcome these drawbacks, since it allows approximation of the underlying burst length distribution functions with arbitrary precision, however, at the cost of increasing number of states. Furthermore, the generated process is short term correlated and captures mutual dependencies between error burst lengths and error free burst lengths. However, so far the results are somewhat disappointing, since the increased model complexity does not give better approximations than the simple two state models. A major reason for this is the quality of the approximation of the underlying burst length distribution functions, especially that for the error free bursts.

7.3 Further Research

There are many interesting topics for further research, which can be subdivided into some groups. The first group is characterized by the variation of the measurement environment. For example, it would be interesting to investigate line of sight scenarios, scenarios with small scale mobility, other types of machines in close proximity and so forth. In the second group we have the influence of the PHY technology. The most interesting candidate is the new orthogonal frequency division multiplex (OFDM) PHY for IEEE 802.11 [22]. The third group deals with finding further approaches for stochastic modeling. For the bipartite model one can investigate further schemes for approximation of the underlying distribution functions. These schemes have impact on both the quality of the approximation and the number of states needed.

Chapter 8

Acknowledgments

We gratefully acknowledge the help of the group at the PTZ Berlin, namely of Dipl.-Ing. Ulrich Doll, Dr.-Ing. Hendrik Engel, Dipl.-Ing. Sascha Piltz, and Dipl.-Ing. Dirk Oberschmidt who helped us in setting up the measurement environment. Furthermore, we want to thank Christian Hoene from the Telecommunication Networks Group, who has developed and set up the first versions of the wireless NIC hardware and software. Andreas Koepke from TKN was of great help in performing the simulations. Our colleagues Holger Karl and Morten Schläger helped us with insightful comments on earlier versions of the report. Finally we gratefully acknowledge the partial sponsorship given by the Deutsche Forschungsgemeinschaft (DFG).

Bibliography

[1] Richard L. Abrahams. *2.4GHz 11Mbps MACless DSSS Radio HWB1151 Users Guide - AN9835.1.* Intersil, 1999.

[2] Jorgen Bach Andersen, Theodore S. Rappaport, and Susumu Yoshida. Propagation Measurements and Models for Wireless Communications Channels. *IEEE Communications Magazine*, 33(1):42–49, January 1995.

[3] Kenneth L. Blackard, Theodore S. Rappaport, and Charles W. Bostian. Measurements and models of radio frequency impulsive noise for indoor wireless communications. *IEEE Journal on Selected Areas in Communications*, 11(7):991–1001, September 1993.

[4] George E. P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. *Time Series Analysis – forecasting and control.* Holden-Day, San Francisco, 3 edition, 1994.

[5] D. Duchamp and N.F.Reynolds. Measured performance of wireless lan. In *Proc. of 17th Conf. on Local Computer Networks, Minneapolis*, 1992.

[6] David Eckhard and Peter Steenkiste. Measurement and analysis of the error characteristics of an in-building wireless network. In *Proc. of ACM SIGCOMM'96 Conference,*, pages 243–254, Stanford University, California, August 1996.

[7] David A. Eckhardt and Peter Steenkiste. A trace-based evaluation of adaptive error correction for a wireless local area network. *MONET - Mobile Networks and Applications*, 4:273–287, 1999.

[8] E. O. Elliot. Estimates of error rates for codes on burst-noise channels. *Bell Systems Technical Journal*, 42:1977–1997, September 1963.

[9] William Feller. *An Introduction to Probability Theory and Its Applications - Volume I.* John Wiley, New York, third edition, 1968.

[10] B. D. Fritchman. A binary channel characterisation using partitioned markov chains. *IEEE Transactions on Information Theory*, 13(2):221–227, April 1967.

[11] J. Garcia-Frias and P. M. Crespo. Hidden Markov Models for burst error characterization in indoor radio channels. *IEEE Transactions on Vehicular Technology*, 46:1006–1020, November 1997.

[12] E. N. Gilbert. Capacity of a burst-noise channel. *Bell Systems Technical Journal*, 39:1253–1265, September 1960.

[13] Intersil. *HFA3860B Data Sheet, File Number 4594.1*, 1999.

[14] Young Yong Kim and San qi Li. Modeling multipath fading channel dynamics for packet data performance analysis. In *Proc. IEEE INFOCOM 98*. IEEE, 1998.

[15] Young Yong Kim and San qi Li. Capturing Important Statistics of a Fading/Shadowing Channel for Network Performance Analysis. *IEEE Journal on Selected Areas in Communications*, 17(5):888–901, May 1999.

[16] Shu Lin and Daniel J. Costello. *Error Control Coding – Fundamentals and Applications*. Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[17] R. H. McCullough. The binary regenerative channel. *Bell Systems Technical Journal*, 47:1713–1735, October 1968.

[18] Motorola Inc. *MPC 860 PowerQUICC Users's Manual*, 1998.

[19] Randolph Nelson. *Probability, Stochastic Processes, and Queueing Theory – The Mathematics of Computer Performance Modeling*. Springer Verlag, New York, 1995.

[20] Giao T. Nguyen, , Randy H. Katz, Brian Noble, , and Mahadev Satyanarayanan. A trace-based approach for modeling wireless channel behavior. In *Proceedings of the Winter Simulation Conference*, Coronado, CA, December 1996.

[21] The Editors of IEEE 802.11. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, November 1997.

[22] The Editors of IEEE 802.11. *Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications: High Speed Physical Layer in the 5 GHz Band*, 1999.

[23] Matthias Pätzold. *Mobilfunkkanäle – Modellierung, Analyse und Simulation*. Vieweg, Braunschweig, 1999.

[24] San qi Li and Chia-Lin Hwang. Queue response to input correlation functions: Continuous spectral analysis. *IEEE/ACM Transactions on Networking*, 1:678–692, December 1993.

[25] San qi Li and Chia-Lin Hwang. On the convergence of traffic measurement and queueing analysis: A statistical-match queueing (smaq) tool. *IEEE/ACM Transactions on Networking*, 5:95–110, February 1997.

[26] William J. Stewart. *Introduction to the Numerical Solution of Markov Chains.* Princeton University Press, Princeton, New Jersey, 1994.

[27] F. Swarts and H.C. Ferreira. Markov characterization of digital fading mobile vhf channels. *IEEE Transactions on Vehicular Technology*, 43(4):977–985, November 1994.

[28] Tundra Corporation. *Reference Manual Tundra PCI Interconnect*, 1998.

[29] W. Turin and R. van Nobelen. Hidden markov modeling of fading channels. In *Proc. IEEE Vehicular Technology Conference*, pages 1234–1238, May 1998.

[30] William Turin. *Digital Transmission Systems – Performance Analysis and Modeling.* McGraw-Hill Telecommunications. McGraw-Hill, New York, 1998.

[31] William Turin and Robert van Nobelen. Hidden Markov Modeling of Flat Fading Channels. *IEEE Journal on Selected Areas in Communications*, 16(9):1809–1817, December 1998.

[32] H.S. Wang and N. Moayeri. Finite State Markov Channel - A Useful Model for Radio Communication Channels. *IEEE Transactions on Vehicular Technology*, 44(1):163–171, February 1995.

[33] Andreas Willig. Analysis of the PROFIBUS Token Passing Protocol over Error Prone Links. In *Proc. 25th Annual Conference of the IEEE Industrial Electronics Society (IECON'99)*, pages 1246 – 1252. IEEE, November 1999.

[34] Andreas Willig, Martin Kubisch, and Adam Wolisz. Bit Error Rate Measurements – Second Campaign, factorial measurement. TKN Technical Report Series TKN-00-011, Telecommunication Networks Group, Technical University Berlin, November 2000. http://www-tkn.ee.tu-berlin.de/publications/tknrreports.html.

[35] Andreas Willig, Martin Kubisch, and Adam Wolisz. Bit Error Rate Measurements – Second Campaign, longterm1 measurement. TKN Technical Report Series TKN-00-009, Telecommunication Networks Group, Technical University Berlin, November 2000. http://www-tkn.ee.tu-berlin.de/publications/tknrreports.html.

[36] Andreas Willig, Martin Kubisch, and Adam Wolisz. Bit Error Rate Measurements – Second Campaign, longterm2 measurement. TKN Technical Report Series TKN-00-010, Telecommunication Networks Group, Technical University Berlin, November 2000. http://www-tkn.ee.tu-berlin.de/publications/tknrreports.html.

[37] Andreas Willig, Martin Kubisch, and Adam Wolisz. Results of Bit Error Rate Measurements with an IEEE 802.11 compliant PHY. TKN Technical Report Series TKN-00-008, Telecommunication Networks Group, Technical University Berlin, November 2000. http://www-tkn.ee.tu-berlin.de/publications/tknrreports.html.

[38] Michele Zorzi and Ramesh R. Rao. Performance of arq go-back-n protocol in markov channels with unreliable feedback. *Wireless Networks*, 2:183–193, 1997.

[39] Michele Zorzi and Ramesh R. Rao. Perspectives on the impact of error statistics on protocols for wireless networks. *IEEE Personal Communications*, 6(5), October 1999.