Chaotic Maps as Parsimonious Bit Error Models of Wireless Channels

Andreas Köpke, Andreas Willig, Holger Karl
Telecommunication Networks Group, Technische Universität Berlin
Sekr. FT5-2, Einsteinufer 25, 10587 Berlin, Germany
Email: {koepke|awillig|karl}@ee.tu-berlin.de

***Abstract*— The error patterns of a wireless digital communication channel can be described by looking at consecutively correct or erroneous bits (runs and bursts) and at the distribution function of these run and burst lengths. A number of stochastic models exist that can be used to describe these distributions for wireless channels, e.g., the Gilbert-Elliot model.**

When attempting to apply these models to actually measured error sequences, they fail: Measured data gives raise to two essentially different types of error patterns which can not be described using simple error models like Gilbert-Elliot. These two types are distinguished by their run length distribution; one type in particular is characterized by a heavy-tailed run length distribution. This paper shows how the *chaotic map* model can be used to describe these error types and how to parameterize this model on the basis of measurement data. We show that the chaotic map model is a superior stochastic bit error model for such channels by comparing it with both simple and complex error models. Chaotic maps achieve a modeling accuracy that is far superior to that of simple models and competitive with that of much more complex models, despite needing only six parameters. Furthermore, these parameters have a clear intuitive meaning and are amenable to direct manipulation.

In addition, we show how the second type of channels can be well described by a semi-Markov model using a quantized lognormal state holding time distribution.

***Index Terms*— Digital wireless channel, error model, chaotic map, heavy-tailed run length distribution, IEEE 802.11.**

I. Introduction

The bane of wireless communication is errors: Communication over a wireless channel is faced with much more frequent errors of supposedly different error characteristics, compared to errors in wire-line communication. Understanding these error characteristics is important for many purposes, e.g., for simulation-based performance evaluation of wireless communication protocols or for exploiting knowledge about these error characteristics within a protocol that dynamically adapts its behavior, e.g., the packet size, to these very characteristics. In either case, a compact representation of the error characteristics, an *error model*, is required.

Such an error model could attempt to deterministically describe the causes of errors, however, some sources of errors in wireless communication are intrinsically stochastic, e.g., thermal noise. Therefore, all relevant error models are also stochastic in nature (Figure 1 gives an overview).

A. Willig is now with the Hasso-Plattner-Institute, University of Potsdam, willig@hpi.uni-potsdam.de.

Part of this work was sponsored by a grant from the Deutsche Forschungsgemeinschaft.

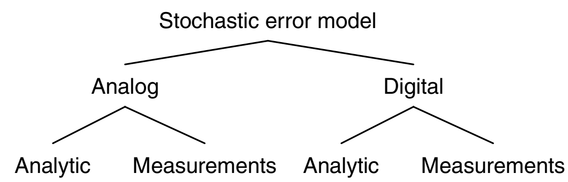

Fig. 1. Overview of possible error models

Such stochastic models can be developed on an analog level, describing the behavior of the received signal strength and/or the interference power, or they can be applied to the digital level, describing the sequence of (possibly erroneous) bits that the receiver's radio front end delivers to higher protocol layers. The advantage of digital models is that they describe exactly the errors that higher protocol layers (such as medium access control or data link control) have to contend with, already taking into account all possible impediments ranging from, e.g., varying channel characteristics to synchronization problems in the receiver hardware.

In both cases, the model can be based on analytic reasoning or on measurements. Common analog, analytic models are the Rice and Rayleigh fading models [1]; measurements for the analog level have been described, for example, in [2]. An example for a digital, analytic model is the well-known Gilbert-Elliot model [3], [4]: it is based on the observation that errors in a communication channel appear in groups and expresses this observation using a two-state Markov model.

For the purpose of this paper, we are interested in the fourth category: using actual bit error measurements to develop a stochastic model of the digital behavior of wireless transmission. We focus on this model type especially because of the ability to directly reflect the intended communication technology and deployment scenarios.

In order to perform such measurements of a digital communication system, we developed a measurement setup based on the IEEE 802.11b wireless LAN standard. The details of this setup are described in Section II; its main property is that it allows to detect differences between the ith sent bit s_i and the corresponding ith received bit r_i: if $r_i = s_i$, no error occurred, if $r_i = \neg s_i$, the ith bit has been modified during transmission. Put briefly, $r_i = s_i \ \texttt{XOR}\ e_i$, where e_i is an error bit indicating whether or not bit i has been transmitted correctly.

Using this measurement setup, we recorded a number of

0-7803-7753-2/03/$17.00 (C) 2003 IEEE

traces; a trace is the sequence of error indicator bits e_i. Such a trace can be concisely represented by counting the number of consecutive error-free and erroneous bits. A sequence of error-free bits is called a *run*, consecutive erroneous bits are called a *burst*. Using this representation, a fundamental task of a stochastic error model is to describe the random distribution of the lengths of both runs and bursts—it is a model for these distributions that we are pursuing in this paper.

In addition, an error model can also attempt to model the patterns in which runs and burst lengths follow upon each other. Inspecting our collected traces indicates that there is no clearly discernible pattern with which either runs follow upon each other (e.g., is a long run followed by another long run?) or runs and bursts follow each other (e.g., is a long run followed by a short burst?)—details are discussed in Section II. Therefore, modeling these patterns is not appropriate and not attempted here as it would needlessly add complexity to a stochastic error model.

Several stochastic models for the occurrence of bit errors exist, e.g., the Gilbert-Elliot or semi-Markov models. When using these models to describe our measured data, we find that about in one third of the traces, one specific type of semi-Markov models achieved an acceptable accuracy in describing these measurements; the other models like Gilbert-Elliot were unsatisfactory. For the remaining two thirds of our traces, none of these models resulted in an acceptable description of the data. The specific feature of these traces that is not matched by the existing models is the heavy tail of their run length distributions. This suggests the existence of two essentially different types of channel behaviors which we call *regimes*.

Such a heavy-tail behavior is not merely a modeling curiosity, it has in fact considerable impact on the actual channel behavior: Channels with a heavy-tailed run length distribution would exhibit very long periods without bit errors and this does in fact correspond to the common observation of wireless LAN users that the channel is perfect over long periods of time. Therefore, it is of practical relevance to find models that accurately capture such channel behavior.

In fact, there are some models that are capable of expressing such behavior, however, these models are either complicated due to their large number of parameters or they lack in flexibility. As the heavy-tailed regime is not captured by *simple* stochastic error models, we claim that a new model is required to describe this type of behavior.

One candidate model for describing such heavy-tailed behavior while requiring only a few parameters are the so-called *chaotic maps*. These chaotic maps have been successfully used to model arrival processes in communication networks [5] and similar distribution assumptions have been used to model errors in wired networks [6]. As both traffic and error models are often described using similar intuitive terms like "bursts", we were interested in checking whether a model capable of describing bursty traffic is also transferable to the description of the quite different phenomenom of bursty bit errors in wireless channels. Our main contribution is to show that chaotic maps are indeed a concise, efficient, and appropriate representation of error processes in wireless channels. In particular, we present algorithms to parameterize chaotic maps from measured data; we also give an explanation of the intuitive meaning of these parameters and their relation to the notion of "channel quality". We developed the model and its parameterization process on the basis of a randomly chosen subset of our traces. To assess the suitability of chaotic maps, we used *all* traces (including those which were not used in the model development) and compared the distribution functions of run and burst lengths produced by chaotic maps and several other models (ranging from simple to complex) with measured data using a metric expressing the difference of two distribution functions. We show that for the heavy-tailed regime, chaotic maps are a superior error model.

The remainder of this paper is organized as follows. Section II describes our measurement setup and the measured data in more detail. Section III considers the applicability of existing models to our data and Section IV introduces the chaotic map along with the parameterization algorithms. Formal metrics for comparing these different models are explained in Section V, the comparison results are shown in Section VI. Section VII puts the results of these evaluations into perspective with other models and related work. Finally, Section VIII contains the conclusions and possibilities for future work.

II. DATA

In this section we give a brief overview of the measured bit error data to which we apply the chaotic map approach. Besides a short sketch of the measurement setup we give a high-level overview of the most important findings regarding the run lengths and burst lengths as well as their correlation characteristics. A more detailed description of the setup and the goals of the measurement as well as the results can be found in [7].

A. Measurement setup

At the heart of our setup was a MAC-less radio modem based on the Harris/Intersil PRISM I chipset [8], which is compliant with IEEE 802.11b direct sequence spread spectrum (DSSS) technology. It offers, amongst others, 1 MBit/s BPSK modulation (differential binary phase shift keying) and 2 MBit/s QPSK modulation (differential quaternary phase shift keying). The radio modem basically consists of high-frequency circuitry and a baseband processor. The latter accepts and delivers a packetized serial bit stream from/to upper layers, optionally scrambles the data (using a shift register with feedback), and performs DSSS processing [9]. The characteristics of the received serial bit stream is our focus of interest.

We have used two dedicated stations, a *transmitter station* and a *receiver station*; the setup is sketched in Figure 2. The basic idea is that the transmitter station sends a well-known sequence of data over the wireless link, which is captured and stored by the receiver station into a log file for later evaluation. The wireless network card contains a specific measurement application and neither MAC functionality nor any higher layer protocols. It is important to emphasize that the packet data is generated exclusively by the measurement setup and that there is no MAC instance creating header fields or controlling the instant of medium access. This way we have fine-grained control over the packet generation and reception process and no bias is introduced by upper layer protocols.

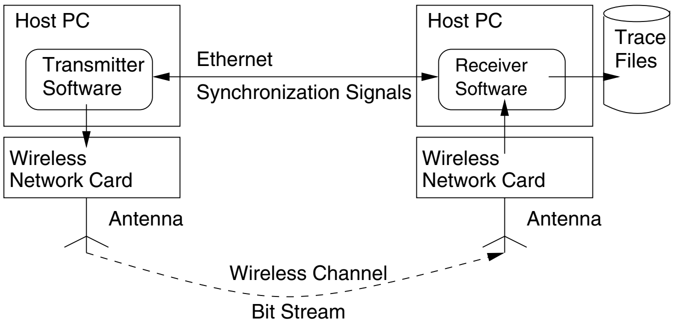

Fig. 2. Measurement setup

In order to record a trace, a fixed number of packets is transmitted; the trace itself is computed as the difference between the transmitted and the actually received data. During the recording of a trace, all packets have the same parameters (packet size, scrambling mode, modulation scheme, etc.); they are varied from trace to trace.

Our setup was tested in laboratory measurements and in other measurement campaigns in controlled environments and has shown to work correctly. For example, in a non-line-of-sight (NLOS) environment without any interferers over distances between 10 and 40 meters almost no bit errors or packet losses were observed over several hours (millions of packets).

B. Collected data

The traces were recorded in an industrial environment, all at the same position (non-line-of-sight scenario, $\approx$ 7-8 m distance between transmitter and receiver) and on the same day. As the industrial setup was operating during the measurement campaigns, the wireless medium was taxed with electromagnetic interference, however, to the best of our knowledge no competing IEEE 802.11 sources where in the vicinity of our measurements. From these traces, we discard those that have too small a number of runs (< 180) to give meaningful estimates of burst/run length distributions and mean values. The remaining 34 traces (which are no longer consecutively numbered) of interest cover different packet sizes and scrambling modes (on/off). Of these traces, 15 use BPSK, 19 use QPSK.

The first step in our evaluations is to figure out, for every received packet, which bits are incorrectly received and which ones correctly. This is done by `XOR`ing the received packet with the transmitted packet. For each packet we obtain a sequence of error bits e_i. By concatenating the error bits of all packets of a trace we get a single long error bit sequence for the entire trace. For the concatenated sequence, we count the number of consecutive error-free and erroneous bits, deriving the run lengths and burst lengths, respectively. An artificial example for a concatenated bit error sequence is 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1 (where 0 indicates a correct, 1 an incorrect bit). This corresponds to the run lengths 3, 2, 4 and burst lengths 1, 3, 1.

This concatenation disregards the packet boundaries and channel behavior during the packet gaps. However, a measurement setup that works on the digital level of a packetized communication system is intrinsically limited to such types of statements. Hence, to rest a stochastic model on such a foundation, it is appropriate to assume such a simplified representation of the channel.

We have seen some artifacts in our data which we did not expect (and which have shown up with a different set of modems of the same brand, too): The first artifact is that two single bit errors quite often have a distance of 126–128 bits with QPSK and of 63–64 bits with BPSK; we call this phenomenon "periodic errors"[1]. The second artifact is that bit errors occur with higher probability at the beginning of a packet, which makes the resulting error patterns dependent on the packet size. As there is no real reason to assume that these artifacts are due to the wireless channel itself, we conjecture both of them to be caused by a suboptimal receiver design. Also, other measurements of wireless channels do not report periodic errors or dependencies on packet content or position within a packet (cp. in particular [10, p. 276], other references like [11] are not concerned with *bit level* measurements).

Strictly speaking, it might be possible to find or develop models that are capable of expressing such behaviors. However, this would add considerable complexity to the model without capturing an effect essential to the characteristics of the wireless channel as such. Therefore, we consider these effects to be spurious and do not attempt to model them explicitly. Nevertheless, we are still faced with the problem of finding a model that captures error characteristics (as described next) that are not related to these artifacts.

C. Run and burst length distribution

Given the set of all run lengths of a particular trace, we compute this trace's empirical run length distribution by counting how many observed runs are smaller than a particular run length: For a random variable L representing the length of a run, $F_L(l) = \Pr(L \leq l)$ is the cumulative distribution function (CDF) of L, expressing the probability that a run of length up to l occurs.[2] For visualization purposes, we usually use the *complementary cumulative distribution function (CCDF)* $1 - F_L(l)$; an example is shown in Figure 3. This figure uses a double-logarithmic plot that graphically emphasizes the distribution function's behavior at small probability values, the so-called *tail* of a distribution function.

A striking characteristic of Figure 3 is the apparently linear form of the distribution's tail for values of l larger than about 200 bits. This form is present in about two thirds of our traces' run length distributions and is the very property that is hardly describable using existing models (see Section III). Distributions with such a linearly decaying tail are characterized by a CCDF that follows a power law, formally $\Pr[L > l] \sim l^{-\alpha}$ with $0 < \alpha < 2$; these types of distributions are called "heavy-tailed" distributions [12].

The burst length distributions, on the other hand, are somewhat different in that the probability of encountering bursts of more than one or two erroneous bits is very low: for almost all BPSK traces, more than 95% of all bursts have a length of

[1] The term was coined because of the (almost) periodic peaks produced by this behavior when plotting the conditional probability that bit $n + k$ is erroneous given that bit n is erroneous vs. lag k.

[2] More precisely, F_L is a right-continuous step function.

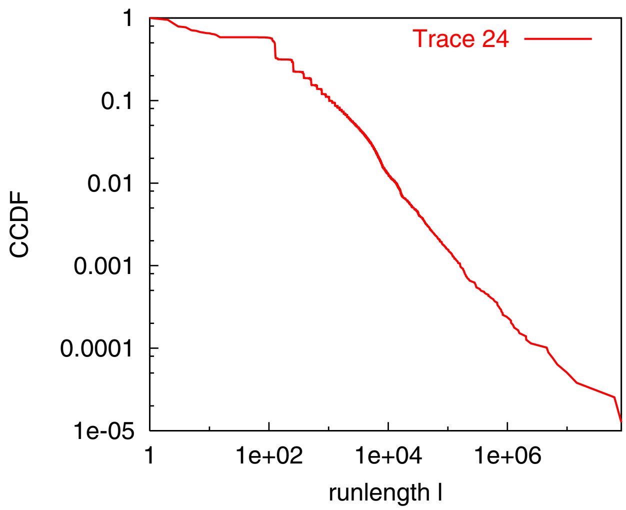

Fig. 3. Complementary cumulative distribution function of the run lengths resulting from Trace 24

one and only 6% of the QPSK traces have bursts longer than 2 bits (the maximum burst length of all BPSK and QPSK traces is 34). Hence, it is comparably easy to characterize bursts, they are almost always just a few bits long, justifying to focus our efforts mainly on the run length distribution.

D. Correlation between run lengths

When looking at the sequence of run lengths, not only their distribution is interesting, but also possible correlation between the run lengths, as expressed by their autocorrelation function[3].

Most of the BPSK traces show almost no autocorrelation, with coefficients of correlation well below 0.05. However, for two out of the 15 investigated BPSK traces a specific pattern of correlation shows up: for a single lag $k \geq 1$ a non-negligible correlation value of ≈ 0.2 is obtained, while the neighboring lags show almost no correlation (we call this a "spike"). Specifically, Traces 59 and 65 show a spike at lags $k = 12$ and $k = 6$, respectively.

For the QPSK traces we observe that traces with a single "spike" in the autocorrelation function occur more often, with no preferred value for lag k. In addition, we have some traces where non-negligible autocorrelation is present for several lags. Furthermore, 7 out of the 19 investigated QPSK traces show almost no correlation at all.

The reason for these spikes is the fact that the autocorrelation as a metric is quite sensitive to outliers. After removing the longest runs from the run-length encoded sequence, the autocorrelation disappears (for all practical purposes, these reduced traces are uncorrelated). Therefore, the observed spikes are actually only a chance product: it just so happens that in some traces, two very long runs happen to be close to each other, causing a large autocorrelation for this particular lag. Consequently, it does not stand to reason to attempt to model these patterns, least of all with an intentionally small and compact error model.[4]

[3]The autocorrelation function $R(k)$ for lag $k \in \mathbb{N}$ of a discrete, (wide-sense) stationary stochastic process $X_0, X_1, X_2, X_3, \ldots$ is defined as

$$R(k) = \frac{\mathrm{E}[(X_0 - \bar{X_0})(X_k - \bar{X_0})]}{\sqrt{\mathrm{E}[(X_0 - \bar{X_0})^2]}\sqrt{\mathrm{E}[(X_k - \bar{X_0})^2]}}$$

However, if the (common) distribution function of the r.v.'s $X_0, X_1, \ldots$ is heavy-tailed, neither mean nor variance are necessarily finite. In addition, the process we have observed in our measurements is not necessarily (wide-sense) stationary. Hence, an argument resting on the autocorrelation function is in all strictness not permissible. Nevertheless, we feel justified in using this tool as we apply it only to a finite amount of measurements, for which the sample mean and sample variance are obviously always finite.

III. Suitability of existing models

When attempting to model such data, various more or less complicated stochastic error models exist. Common to all of these models is that they are used to decide for a particular bit whether it is correct or erroneous, resulting in a sequence of error bits. This sequence in turn gives rise to runs and bursts, from which model run and burst length distributions can be computed in much the same way as the traces' empirical distributions.

The particular form of these distributions depends on the model as well on the model's parameters. Intuitively, one would attempt to choose model parameters such that the model run/burst length distribution matches the corresponding empirical distribution (a formal metric for this matching, which was also used for the evaluations briefly described here, is introduced in Section V). This section reviews some popular stochastic error models of varying complexity; for all models, we also discuss whether these models are appropriate in matching our empirical data.

A. Simple Models: BSC, Gilbert-Elliot and semi-Markov

The simplest model is the *binary symmetric channel (BSC)* [13]: for each bit, an independent Bernoulli experiment is performed to decide whether or not the bit is correct. For all experiments, the same bit error probability p is used. This single parameter allows to match the mean bit error rate (MBER) of a trace. Using this model, the run lengths are distributed following a geometrical distribution, $F_L(l) = 1 - (1-p)^l$ which corresponds to the CCDF $1 - F_L(l) = (1-p)^l$ (hence, the tail is exponentially decaying); the burst lengths are also geometrically distributed, $F_L(l) = 1 - p^l$. Because of the restricted shape of the BSC's CCDF, the generated run and burst length distributions do not match any trace's empirical distribution, none of which is even remotely similar to an exponentially decaying shape. Figure 4 shows a comparison of a trace-based and a model-generated run length CCDF.

When attempting to match the empirical run and burst length distributions separately, a model that treats the runs and bursts individually is required. One possibility to do so is to introduce a "good" and a "bad" state: in the good state, only correct bits are generated by the model, in the bad state, bits are incorrect with a certain probability $p \in (0, 1]$—this is the popular *Gilbert-Elliot model* [3], [4], which belongs to the class of Markov models. For such a model, the distribution of the times the model stays in a certain state—the *state holding time*—is an

[4]As it turns out, the chaotic map model described later can also produce such spikes when it so happens that two very long runs are close to each other. Naturally, this does not happen at the same lags as they are a product of chance.

important characteristic; for Markov models in particular, the state holding times for the good and bad state are geometrically distributed. Hence, the model needs three parameters: p, and the mean state holding times for the good state and bad state, respectively. Both the burst length and run length distributions are exponentially decaying and not heavy-tailed. Hence, just as the BSC model, the Gilbert-Elliot model fails to match the run length distribution of many of our traces (as is exemplarily shown in Figure 4).

An obvious approach to improve the ability of the simple two-state Gilbert-Elliot model to match our traces' distributions is to replace the geometric state holding time distributions by other distributions with more adjustable parameters (formally, this turns a Markov model into a so-called semi-Markov model). Selecting an appropriate alternative distribution function requires to carefully inspect the available traces. After considering a number of different candidates, it turned out that a quantized lognormal distribution does in fact provide a reasonably good fit between model run length distribution and trace run length distribution. As no other simple distribution assumptions produced acceptable matches, we only consider such quantized lognormal semi-Markov models in the remainder of the paper and use the term semi-Markov as a shorthand for this particular form of a semi-Markov model. An example for a match produced by the semi-Markov model, consider Figure 5. On the other hand, Figure 4 contains an example where the lognormal semi-Markov completely fails in matching the trace's run length distribution. Hence, there is still a need to find more appropriate models for some of our traces.

At this state of affairs two options are interesting: to look further for alternative distribution functions to approximate the run length CDF in its entirety with only a few parameters, or to use piecewise approximations with substantially more parameters. This paper is concerned with the first option, and the search is continued in Section IV. The bipartite model, discussed next, is an example of the second approach.

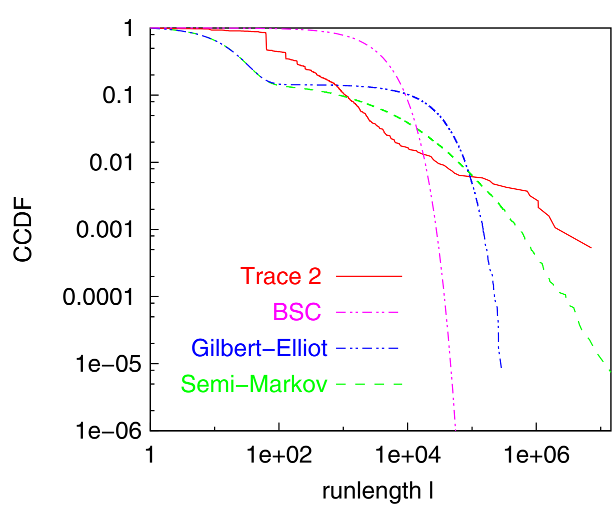

Fig. 4. Comparison of the CCDF's of Trace 2 with those of different simple error models parameterized from the same trace (semi-Markov with quantized lognormal distributions)

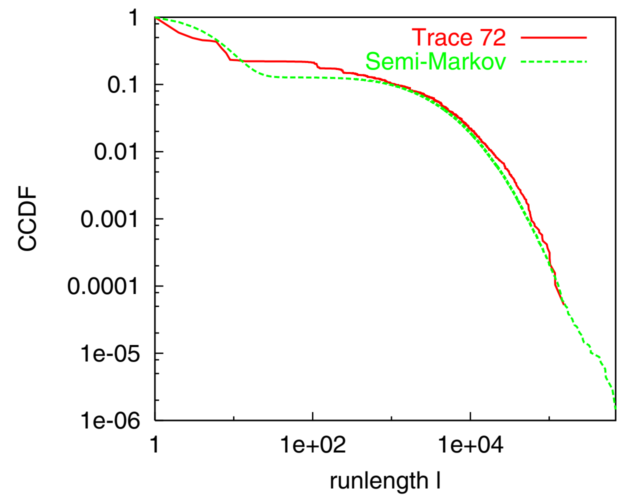

Fig. 5. Comparison of the CCDF's of Trace 72 with those generated by a semi-Markov with quantized lognormal distribution parameterized from the same trace

B. The Bipartite model

The bipartite model is a stochastic Markov model of increased (yet selectable) complexity, to be parameterized from a given trace. It allows the user to choose a proper tradeoff between model complexity and accuracy. A description of the model can be found in [14].

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). Each state s_i is associated with an interval of natural numbers of possible state holding times and a probability distribution for this interval. By properly choosing intervals and distributions, it is possible to approximate the given run or burst length distribution. The operation of the bipartite model is analog to the Markov model: when entering a state, the holding time is determined by the state's associated distribution, and for the bad states, bit errors are generated according to a state-dependent error probability p_i.

In order to build a model from a trace, n_1 and n_2, the transition matrix $\mathbf{P}$ governing the state transitions, the intervals and probability distributions, and the bit error rates in the bad states p_i have to be chosen. The procedure for determining these parameters from a given trace is described in [14], which also shows good correspondence of the bipartite model with original traces.

While this model has some advantages, in order to achieve an acceptable fit with measured data, the bipartite models typically needs over a hundred parameters. As the parameter values are tightly coupled to the traces they were generated from, the model is not amenable to simple experimentation with different parameter settings. Hence, a model that uses only a few parameters with an intuitive meaning that is capable of capturing traces from the heavy-tailed regime is desirable—chaotic maps fill this gap.

IV. CHAOTIC MAP

A. Model description

Similar to the Gilbert-Elliot model, the chaotic map model uses two states "good" and "bad". The main difference between the models lies in deciding the correctness of a particular bit and in the rules that govern the switching between the two states.

The chaotic map model works as follows (cp. [15]): in the good state all bits are correct, in the bad states, all bits are erroneous.[5] Switching between states depends on the value of an auxiliary variable x_t that is updated for each bit:

In the good state,

1) produce a correct bit
2) calculate

$$x_{t+1} = x_t + u_{\mathrm{g}} x_t^{z_{\mathrm{g}}} + \epsilon_{\mathrm{g}} \text{ where } t \in N \tag{1}$$

3) if $x_{t+1} > 1$, choose $x_{t+1} \sim U(0,1)$ and switch to the bad state; otherwise, stay in good state ($U(0,1)$ denotes the uniform distribution on the closed interval [0,1]).

In the bad state, the same rule applies (of course, an erroneous bit is generated) and the parameters u_{b}, z_{b} and ϵ_{b} are in general different from those for the good state. As the bursts are usually very short (and certainly not heavy-tailed), it is possible to use a simpler description for the bad state. We are nonetheless applying the chaotic map model to the burst length distributions as well to assess this model's flexibility and limitations.

When discussing general properties of the chaotic map model, we drop the indices g and b from the parameters. The application to the good or bad state should then be clear from context.

While applying the chaotic map as an error model is straightforward, the challenge lies in estimating the parameters z, ϵ and u for both good and bad state from a trace. As it turned out, using parameterization rules from chaotic maps as *traffic* models did not produce satisfying results (the estimators where either highly sensitive to the presence of periodic errors or asssumed the existence of the mean value of run lengths which was not necessarily the case for the bit errors). Hence, a new parameter estimation technique had to be developed to use chaotic maps as bit error models.

B. Parameter Estimation

The distributions of state holding times of a chaotic map model have a characteristic form: in the double-logarithmic CCDF representation from a certain point on it is a straight line with a cut-off. This lines' position, slope, and cut-off depend on the model parameters u, z, and ϵ; this is illustrated in Figure 6 which also shows a trace's run length CCDF for comparison.

The parameter u "shifts" the CCDF along the length axis. A larger value of u means that there are many short runs in a trace, whereas smaller values of u lead to longer runs.

The parameter z determines the slope of the CCDF. The larger the value for z, the higher the probability to find very

[5] Hence, the state holding time directly corresponds to the length of a particular run (unlike the previously described models); the two terms are interchangeable in the chaotic map model.

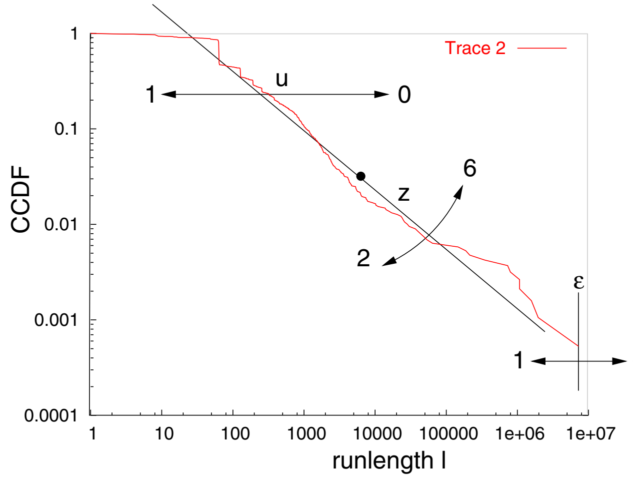

Fig. 6. Influence of parameters

long error-free runs, which corresponds to a channel of higher transmission quality.

The parameter ϵ represents a minimal increment in Eq. (1). Therefore, a state change happens at the very latest after $1/\epsilon$ iterations, limiting the maximum length of an error-free run.

To be able to numerically estimate these parameter values, it is necessary to formalize this intuitive relationship between the shape of the run length CCDF and the parameter values. The goal is to choose these parameters such that the model's CCDF and the trace's CCDF match.

1) Estimation of z: In order to estimate the parameter z, we use the relationship between the CCDF $1 - F_L(l)$ and z [5, p. 83]:

$$1 - F_L(l) \propto l^{-\frac{1}{z-1}} \tag{2}$$

Taking the logarithm of Equation (2) shows that it is possible to estimate the parameter z from the slope of the least squares straight line fitted to the CCDF in the double-logarithmic plot. As can be seen from Figure 6, this is not without problems: For short runs the CCDF is not even approximately linear. Therefore, as a heuristic, we use only the longest 15% of all runs in the calculation of z_{g}. The same problem occurs for the very long runs. We assumed that the deviation from the straight line is due to the small number of samples and excluded the ten longest runs from this calculation. However, these cut-offs should not result in a total number of runs less than about thirty; otherwise, the estimation becomes somewhat arbitrary. For the bad state the estimation of z_{b} is similar, however, due to the small range of burst lengths we do not exclude extreme values.

Besides these restrictions of the data set, the estimation of the parameter z is straightforward. For the other parameters, the situation is more complicated: an initial parameter estimate can be computed directly for both u and ϵ, but this estimate has to be iteratively improved for the good state parameters. This iteration simultaneously computes correction factors for u_{g} and ϵ_{g}.

2) Estimation of u: A major problem with estimating u is that it cannot be isolated from the estimation of ϵ, since both together determine the maximum run length that can be generated. Therefore, we have chosen to use a heuristic to get an initial guess of u, which does not depend on ϵ, which is then iteratively improved.

The initial estimate for $u_{\rm g}$ exploits its relationship with the mean run length $\bar{l}$. The rationale is that a small $u_{\rm g}$ leads to a long run, directly influencing the average run length. After some experimentation we found that

$$u_{\rm g} = \bar{l}^{\,\frac{-1}{3}\frac{z_{\rm g}}{z_{\rm g}-1}}$$

provides a satisfactorily initial guess for $u_{\rm g}$. The idea to take z into account was inspired by a comparable relationship for $\epsilon_{\rm g}$, see Equation (4). The estimator for the bad state is similar:

$$u_{\rm b} = \bar{l}^{\,\frac{1}{z_{\rm b}-1}+1}.$$

It is necessary to further improve the estimate of $u_{\rm g}$. A suboptimal value for $u_{\rm g}$ results in a run length CCDF that does not directly match the trace's run length CCDF, more specifically, the two CCDFs tend to be "parallel" to each other in the double-logarithmic plot (shifted horizontally). The idea how to improve the parameter value is to determine the distance between these curves and to use this distance to modify $u_{\rm g}$. This approach is related to some control-theoretic concepts for chaotic systems [15].

The iteration for $u_{\rm g}$ works as follows: Consider F_t, the run length CDF resulting from the trace, and F_m, the run length CDF resulting from the model with the initial parameter estimates (the model CDF is obtained by simulation). Each CDF consists of a finite number N_t or N_m of samples. Taking the difference in the double-logarithmic plot corresponds to the ratio of the CCDFs. In particular, we use N ratios, where $N = \min\{N_t, N_m\}$. Ratio j is computed as the ratio of the j/N-quantiles of the model CCDF, $F_m^{-1}(j/N)$, and the trace CDF, $F_t^{-1}(j/N)$.[6] The correction factor r_u is then defined as the arithmetic average of these ratios:

$$r_u = \frac{1}{N}\sum_{j=1}^{N}\frac{F_m^{-1}(j/N)}{F_t^{-1}(j/N)} \tag{3}$$

The iteration for $u_{\rm g}$ is then simply a low-pass filtered multiplication with the correction factor:

$$u'_{\rm g} = \frac{1+r_u}{2}\cdot u_{\rm g},$$

which, together with a new estimation for ϵ (see below) produces a new model with CCDF F'_m. The iteration continues until $0.995 < r_u < 1.005$ (and the corresponding correction factor for $\epsilon_{\rm g}$ has also converged).

3) Estimation of ϵ: Recall the role of ϵ in Eq. (1): if $\epsilon = 0$, the generated CCDF would be heavy-tailed in the strict sense, i.e. the mean value does not exist for $z > 2$. Hence, if we employ such a model in a finite-length simulation, the generated bit error rate depends on how long the simulation is run.[7] The choice of $\epsilon > 0$ places an upper bound on the generated run lengths and thus allows to adjust the generated bit error rate.

Therefore, the longest observed run or burst can be used to estimate ϵ. A trivial estimator would be to just use $\epsilon = 1/l_{\max}$ For the good state's $\epsilon_{\rm g}$ value, it turns out that also taking into account the impact of $z_{\rm g}$ gives a better initial guess for $\epsilon_{\rm g}$ as the minimal increment in Equation (1) is not only due to $\epsilon_{\rm g}$. Some numerical experimentation showed that the following equation:

$$\epsilon_{\rm g} = l_{\max}^{\,\frac{-4}{5}\frac{z_{\rm g}}{z_{\rm g}-1}} \tag{4}$$

serves as a good starting point for $\epsilon_{\rm g}$.[8] This estimator was not only chosen with respect to $\epsilon_{\rm g}$, but the given joint initial estimations for $u_{\rm g}$ and $\epsilon_{\rm g}$ have the positive effect that our iteration process for $u_{\rm g}$ converges comparably fast. With other choices for $\epsilon_{\rm g}$ it takes much longer for $u_{\rm g}$ to converge.

Similar to $u_{\rm g}$, this initial estimate for $\epsilon_{\rm g}$ needs to be improved. To find a good optimization criterion, we recall that $\epsilon_{\rm g}$ and the generated bit error rate are related. We use this relation to guide our search for a better $\epsilon_{\rm g}$. While it is possible to use this relationship to attempt to match the bit error rate, it seems more meaningful to look at packet errors instead of bit errors, since packet errors are of primary interest for the behavior of protocols.[9] This gives us a simple approach for improving $\epsilon_{\rm g}$: the packet error rate PER_t of a trace is known, and the packet error rate PER_m of the chaotic map model can be obtained using the simulations already necessary to compute the correction factor r_u (using the trace's packet size to divide the error indicator sequence into packets). The correction factor r_ϵ is then simply:

$$r_\epsilon = \frac{\mathrm{PER}_t}{\mathrm{PER}_m}$$

and we set

$$\epsilon'_{\rm g} = \frac{1+r_\epsilon}{2}\cdot\epsilon_{\rm g}.$$

Here, the iteration is terminated when $0.9 < r_\epsilon < 1.1$ (and r_u has also converged). The correction factors for $\epsilon_{\rm g}$ and $u_{\rm g}$ are estimated simultaneously and a new model is computed with both corrections. For all traces of the heavy-tailed regime the joint iteration finishes after 10 to 30 steps; for the other traces the iteration of r_u converges fast, but for r_ϵ sometimes no clear limit is approached.

Regarding the burst lengths, the initial estimates for $u_{\rm b}$ and

$$\epsilon_{\rm b} = \frac{1}{l_{\max}u_{\rm b}}$$

are usually sufficiently good so no additional iterations are computed here.

As a first impression of the results achievable with the chaotic map model, Figure 7 shows the run length CCDFs for Trace 2 and the corresponding chaotic map model. Comparing this figure with Figure 4 immediately reveals a much better fit of the trace CCDF than what is achieved using other models.

[6]For a stepwise defined distribution function F, the quantile q_x is defined via the generalized inverse distribution function: $q_x = F^{-1}(x) = \inf\{q : F(q) > x\}$.

[7]For the special case of a Pareto state holding time distribution (which corresponds to $u = 1$ and $\epsilon = 0$) the bit error rate μ depends on the number of simulated bits T as follows (compare [6, p. 143]):

$$\mu = T^{-2+z}$$

[8]This formula is a variation of a formula given in [15, p. 88].

[9]We consider a packet as erroneous if it contains at least one bit error.

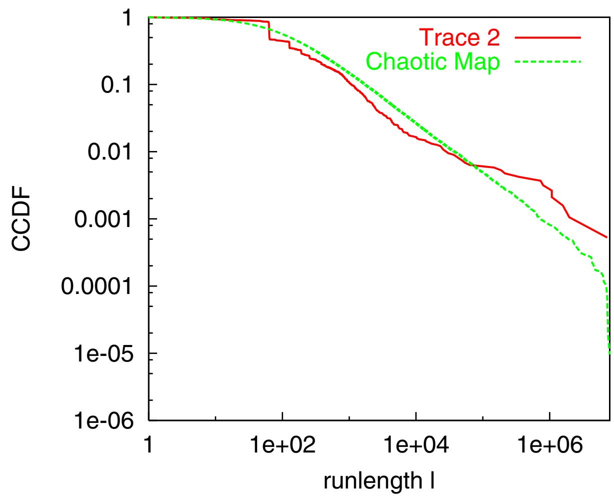

Fig. 7. Comparison of Trace 2 and the chaotic map model

V. Metrics

Evaluating the appropriateness of these models can happen under different perspectives. One such perspective is to check how well a model reproduces a trace's run length distribution function; this aspect is covered by an area metric. Alternatively, a perspective that reflects the interests of a user of a stochastic error model can be taken who wants to simulate packets traveling across a wireless link; an appropriate metric for this purpose would be to compare packet error rates.

A. Area

In the literature many possible metrics are known for comparing two (distribution) functions, e.g. the L_2-norm, the L_∞-norm etc. These metrics have a severe drawback for our application. Consider the case that a single very long run is not reproduced by the model. As all common metrics are sensitive to such very large runs, they would consider these two distributions as vastly different. Often, this problem can be circumvented by regarding such a single large run as an outlier. However, this is not possible here as these large runs are a typical feature of the measurements and should not be neglected.

A convenient measure to compare two distribution functions is to look at their quantiles: The closer the lengths corresponding to a given quantile are, the better the match; if the two lengths are equal, the match is perfect. Furthermore, in order to reduce the influence of single long runs, we do not look at the lengths themselves but at their logarithms. As a single quantile does not include a sufficient amount of information, the average of these quantiles or, similarly, the area between the two distribution functions can be used. The area is approximated with the well-known trapezium formula:

$$
\begin{aligned}
A &= \frac{1}{N}\sum_{j=1}^{N}\left|\log(F_m^{-1}(j/N)) - \log(F_t^{-1}(j/N))\right| - \\
&\quad \frac{|\log(F_m^{-1}(1/N)) - \log(F_t^{-1}(1/N))|}{2N} - \\
&\quad \frac{|\log(F_m^{-1}(1)) - \log(F_t^{-1}(1))|}{2N}
\end{aligned}
$$

where $N = \min\{N_t, N_m\}$ as above.

B. Packet Error Rate

The second measure is a result of the intuitive wish that the output generated by a model matches certain "simple" characteristics of the trace, e.g. its mean bit error rate. However, based on the observations that: a) the packet error process is more meaningful to protocols; b) many protocols are sensitive to the packet error rate (a simple example is the throughput of the send+wait protocol); and c) the packet error rate directly depends on the generated burst and run lengths of the (bit error) model and the packet size, we have chosen to use the packet error rate as a simple measure of model quality. The last point might be the decisive one as even models with the same bit error rate can behave quite differently on a packet error level when the model fails to correctly model error burstiness.

More specifically, for a set of different packet sizes ranging from single bit errors to packets sizes corresponding to Ethernet packets (1, 5, 10, 50, 100, 500, 1000, 5000, 10000, 15000 bits) we determine the number of erroneous packets when packets of this size are transmitted over a channel described by the error model under investigation.

Please note that the parameterization of the chaotic map model's ϵ parameter takes only a single packet size into account. It is interesting to see how the model behaves when errors for different packet sizes are computed from the model.

VI. Model Comparison

Most of the models examined here have few parameters, the bipartite has a lot. This model is quite complex, but is this complexity really necessary or are simpler models better? In order to compare the models, we computed the model parameters for each trace and determined which model approximates the trace best.

A. Area

After we parameterized the models, we used them to generate the Cumulative Distribution Function (CDF) of the run lengths using simulations. After that, we computed the area between the model's and the trace's CDF. Table I shows the number of traces where a particular model achieved the best approximation according to the area metric.

TABLE I
Evaluation results for area metric

Model	*No. of Traces*	*w/o bipartite*
Bipartite model	17	—
Chaotic Map	13	21
Semi-Markov	3	11
Gilbert-Elliot	1	2

In Table I the bipartite model is the best approximation for many traces, but not for all. It is possible to use more parameters to increase the fit, but the bipartite model considered here already has more than 100 parameters. This high number of parameters limits the model, since it is hard to change a few in order to get a channel with different desired transmission quality. The other models can be adapted more easily. Their parameters are well understood and can be changed to get a different channel behavior.

If we drop the bipartite model because of its complexity, the chaotic map model best matches 2/3 of all traces. These traces belong to the heavy-tailed regime. The semi-Markov model is suitable for those traces that do not belong to the heavy-tailed regime.

B. Packet Error Rate

The second metric compares the packet error rates for different packet sizes. Figure 8 demonstrates the almost perfect match of the PER as a function of the packet size achieved by the chaotic map model for a heavy-tailed trace. In contrast, Figure 9 illustrates how a non heavy-tailed trace is best approximated by a semi-Markov model—the failure of the chaotic map model is not surprising as it was not built to capture such behavior.

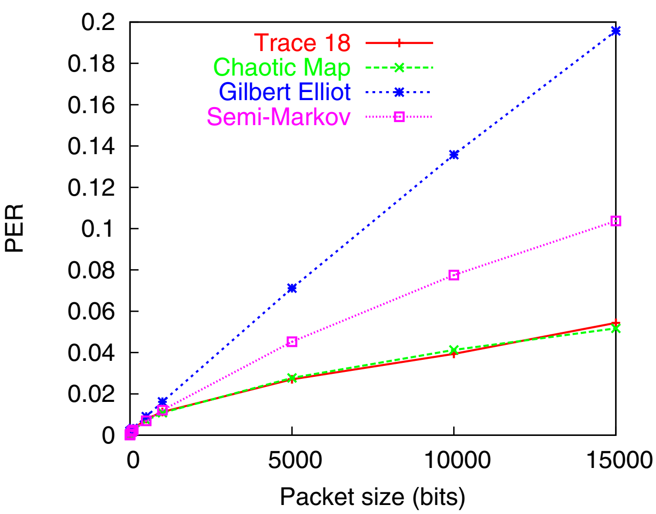

Fig. 8. PER as a function of packet size, trace 18

In general the conclusions from the area measure also hold here: for traces belonging to the heavy-tailed regime the chaotic map model is the model of choice (as illustrated by Figure 8), whereas for traces belonging to the other regime, the semi-Markov model is more appropriate (example in Figure 9).

It is interesting to consider the relationship between the area metric and the PER metric. It is always true that when the chaotic map model is the best model with respect to the area measure, it is also the best model with respect to the PER measure. For the semi-Markov model, this is also true in most cases. However, there are a few exceptions where the semi-Markov model achieves the best area metric but the PER metric is inconclusive and, usually, no model correctly describes the PER behavior. The bipartite model, owing to its high complexity and many degrees of freedom, gives for almost all traces very good results for the PER metric (not shown here). In contrast, the Gilbert-Elliott model tends to massively overestimate the PER for almost all traces. This shows that the Gilbert-Elliott model

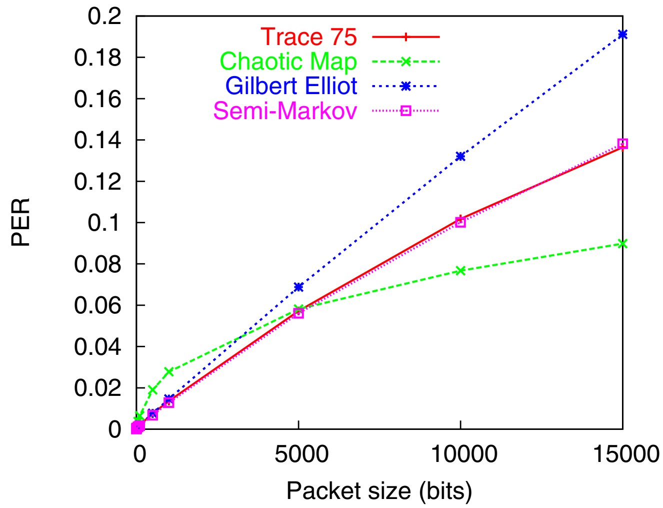

Fig. 9. PER as a function of packet size, trace 75

is not able to capture the burstiness of actually observed wireless channels, in fact, its results are quite dismal.

It is worth noting that although only a single packet size is used to estimate the chaotic map model's parameter ϵ_{g}, it gives very good predictions for all the investigated packet sizes, whenever it is applicable. In this sense, the chaotic map model and its parameterization process are robust.

VII. Discussion

As the previous section has shown the potential of the chaotic map model, it is possible to provide additional insights into the intuitive meaning of its parameters. Such an understanding allows to modify them in a meaningful way, resulting in a simple possibility to vary the simulation conditions between good and bad channels of varying burstiness.

A characteristic parameter for the chaotic map model is z. This value not only determines the slope of the CCDF's tail, but for $\epsilon = 0$ also the finiteness ($z < 2$) or infiniteness ($z \geq 2$) of a mean run length (an infinite mean run length corresponds to a BER of zero). For our traces, the parameter z_{g} ranges from ≈ 1.95 to ≈ 4.11. In reference [16] observations that correspond to z_{g} ranging from 3.5 to 6 are given. In [17], z_{g} of a wired channel is reported to be in the range of 3 to 10. The parameter z_{g} is the only parameter comparable in all papers, since it directly corresponds to the slope of the CCDF in the double-logarithmic plot.

The parameter ϵ is rather related to the principle structure of the model. It introduces a sharp cut-off for the maximum run length, which is not plausible to assume for real-world data. On the other hand, a real transmission system is always subject to thermal noise, introducing a very small, yet non-negligible bit error rate, which results in a *soft*, probabilistic cut-off. The consequence of this soft cut-off is that for extremely long run lengths, the length distribution degenerates into a geometric distribution. Empirical studies [6], [16] substantiate this claim.[10]

[10] We note in passing that this observation also rules out the use of purely Pareto-based modeling of run lengths, as a Pareto distribution would also overestimate the probability for extremely long runs.

Such an exponentially decaying tail cannot be captured by the chaotic map model with any constant ϵ, but the ϵ-induced cutoff provides a simple approximation for this effect. In addition, as this happens only for extremely long runs (typical values are around 10^{12} to 10^{14} bits), this inaccuracy is likely not relevant for most practical purposes.

A main advantage of the chaotic map model is the flexibility for short runs due to the parameter u_{g}. It allows to generate more short run lengths when compared to a model where the Pareto distribution is used (which produces a straight line in the log-log plot). Shorter runs let bursts appear closer to each other. We observed u_{g} in the interval 2 to 0.001.

The model is also capable of expressing the most important characteristics of the bursts. The low maximum burst lengths are reflected by ϵ_{b} values several orders of magnitude larger than those for the runs. The finding that most of the bursts have only a length of one or two bits can be accommodated by choosing large values for u_{b}. In fact, $u_{\mathrm{b}} > 1$ holds for all traces. The observed behavior in our traces ranges from only single bit errors ($z_{\mathrm{b}} = 0,\ u_{\mathrm{b}} = 1,\ \epsilon_{\mathrm{b}} = 1$) as well as several bit errors in a row (e.g., $z_{\mathrm{b}} = 1.67,\ u_{\mathrm{b}} = 6.83,\ \epsilon_{\mathrm{b}} = 0.0046$).

When putting these results into practice, we would recommend to use several different parameter sets: at least a good channel with $z_{\mathrm{g}} = 4$, $\epsilon_{\mathrm{g}} = 10^{-15}$, and $u_{\mathrm{g}} = 1$ and a really bad channel with $z_{\mathrm{g}} = 2$, $\epsilon_{\mathrm{g}} = 10^{-10}$ and $u_{\mathrm{g}} = 0.01$ should be considered in a serious protocol performance evaluation. In addition, also channels from the non-heavy-tailed regime should be taken into account, using the semi-Markov model with a quantized lognormal distribution and state holding times e.g. such as follows: $\mu_{\mathrm{g}} = 70800$, $\sigma_{\mathrm{g}} = 300000$, $\mu_{\mathrm{b}} = 50$, $\sigma_{\mathrm{b}} = 60$, and $\mathrm{BER} = 0.15$ for a good channel, and $\mu_{\mathrm{g}} = 5000$, $\sigma_{\mathrm{g}} = 10000$, $\mu_{\mathrm{b}} = 50$, $\sigma_{\mathrm{b}} = 100$, and $\mathrm{BER} = 0.16$ for a bad channel.

The importance of including heavy-tailed channels in a protocol performance evaluation becomes immediately evident when considering protocols such as TCP running over a wireless channel. In a heavy-tailed channel, the protocol has a chance to reach its steady state when a long run of correct bits is present; in a bursty phase, a protocol like TCP will not produce any goodput at all. Considering the PER behavior as illustrated by Figures 8 and 9, evaluating TCP over a Gilbert-Elliot channel would lead to very misleading conclusions as such simple channel models are not capable to model long-term dependencies of a real channel which are an important characteristic for a non-trivial communication protocol.

The main disadvantage of the chaotic map model is the runtime cost: for each simulated bit two additions, one multiplication and a power must be computed.

VIII. Conclusions and Outlook

When weighing the pros and cons of the various bit error models, the chaotic map model appears to be superior in many respects. It succeeds in describing heavy-tailed channel behaviors for which other models fail or are too complex. As this type of channel is not only the dominant one in our measurements, but is reported in quite different environments as well, a compact, economic model for this channel type is an important contribution. In addition, the chaotic map model is an appropriate model not only for wireless channels but for other types of communication channels as well.

A large practical advantage of the chaotic map model is its small, easy to comprehend set of parameters. These parameters allow a performance evaluation based on this model to be run over, different clearly identifiable channels of varying quality; we have shown some recommendable parameter settings for this purpose. Furthermore, at least one of these parameters (z_{g}) is easily compared across different studies and is in fact the parameter that has the largest impact on the channel quality. Despite its simplicity, the model is capable of expressing a wide variety of channel types as exemplified by its ability to model the burst length distributions.

Nonetheless, a practical performance evaluation should also include non-heavy-tailed channels. For this type of channels, we recommend semi-Markov models with a quantized lognormal distribution. The need to handle these different types of channels in a performance evaluation study is clearly indicated by our measurement data.

The existence of these two quite different regimes of channels is one issue for future work. What is the fundamental reason for the occurence of heavy tails in the run lengths? Is this behavior observable in other, non-industrial environments as well—apparently yes, as other sources [6], [16] have reported similar findings. How can they be explained, and under what circumstances does a channel switch from one regime to another? To what degree does it depend on the usage of different modulation or coding techniques? What are the stochastic properties of this change, can this be captured in a simple metamodel (following the work of [18])?

A more technical item for future work pertains to the parameterization process of the chaotic map. In case the model is not applicable (the trace is not heavy-tailed), the iteration process for ϵ_{g} does usually not converge. It would be interesting to see whether this fact can be exploited to automatically detect which regime is present and which model type should be used. Additionally, an exponentially decaying tail could be added to the run length distribution in order to remove the strict limit on the run lengths; for practical purposes, however, the effect should be quite small.

References

[1] Theodore S. Rappaport, Rias Muhamed, and Varun Kapoor, “Propagation models,” in *The Communications Handbook*, Jerry D. Gibson, Ed., pp. 1182–1196. CRC Press / IEEE Press, Boca Raton, Florida, 1996.

[2] 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*, vol. 11, no. 7, pp. 991–1001, Sept. 1993.

[3] E. N. Gilbert, “Capacity of a burst-noise channel,” *Bell Systems Technical Journal*, vol. 39, pp. 1253–1265, Sept. 1960.

[4] E. O. Elliot, “Estimates of error rates for codes on burst-noise channels,” *Bell Systems Technical Journal*, vol. 42, pp. 1977–1997, Sept. 1963.

[5] Parag Pruthi, *An Application of Chaotic Maps to Packet Traffic Modeling*, Ph.D. thesis, Kungl Tekniska Högskolan, Stockholm, Sweden, October 1995.

[6] J. M. Berger and Benoît B. Mandelbrot, “New model for the clustering of errors on telephone circuits,” In *Multifractals and 1/f Noise* [19], pp. 132–155, originally appeared in: IBM Journal of Research and Development 7, 1963, pp. 224–236.

[7] Andreas Willig, Martin Kubisch, Christian Hoene, and Adam Wolisz, “Measurements of a Wireless Link in an Industrial Environment using an IEEE 802.11-Compliant Physical Layer,” *IEEE Transactions on Industrial Electronics*, 2002, accepted for publication.

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

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

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

[11] 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.

[12] Benoît B. Mandelbrot, "Long-run linearity, locally gaussian process, H-spectra and infinite variance," *Intl. Economic Review*, vol. 10, no. 1, pp. 82–111, Feb. 1969.

[13] John G. Proakis, *Digital Communications*, McGraw-Hill, New York, third edition, 1995.

[14] Andreas Willig, "A New Class of Packet- and Bit-Level Models for Wireless Channels," in *Proc. IEEE PIMRC*, 2002, accepted for publication.

[15] Heinz Georg Schuster, *Deterministisches Chaos: Eine Einführung*, VCH, Weinheim, 1994.

[16] Andreas Ahrens, "A new digital radio-channel model suitable for the evaluation and simulation of channel effects," in *Colloqium Speech Coding Algorithms for Radio Channels*, London, April 2000, IEE Electronics and Communications, pp. 2/1–2/15, Ref. No. 2000/030.

[17] Benoît B. Mandelbrot, "Self-similarity and conditional stationarity," In *Multifractals and 1/f Noise* [19], pp. 166–205, originally appeared in: IEEE Transactions on Communications Technology: COM-13, 1965, 71–90.

[18] E. Costamagna, A. Fanni, L. Favalli, and P. Gamba, "Experiments in modeling the parameters of chaos equation models for mobile radio channels," in *IEEE-APS Conference on Antennas and Propagation for Wireless Communication*, 2000, pp. 103–106.

[19] Benoît B. Mandelbrot, *Multifractals and 1/f Noise*, vol. N of *Selecta*, Springer, New York, Berlin, 1999.