Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

An independent but not identically distributed bit error model for heavy-tailed wireless channels*

Jia LU[†1], Wei YANG[2], Jun-hui WANG[1], Bao-liang LI[1], Wen-hua DOU[1]

([1]*School of Computer, National University of Defense Technology, Changsha 410073, China*)

([2]*Navy Academy of Armament, Beijing 10036, China*)

[†]E-mail: lujia661@126.com

Received June 5, 2012; Revision accepted Nov. 18, 2012; Crosschecked Dec. 11, 2012

Abstract: The error patterns of a wireless channel can be represented by a binary sequence of ones (burst) and zeros (run), which is referred to as a trace. Recent surveys have shown that the run length distribution of a wireless channel is an intrinsically heavy-tailed distribution. Analytical models to characterize such features have to deal with the trade-off between complexity and accuracy. In this paper, we use an independent but not identically distributed (inid) stochastic process to characterize such channel behavior and show how to parameterize the inid bit error model on the basis of a trace. The proposed model has merely two parameters both having intuitive meanings and can be easily figured out from a trace. Compared with chaotic maps, the inid bit error model is simple for practical use but can still be deprived from heavy-tailed distribution in theory. Simulation results demonstrate that the inid model can match the trace, but with fewer parameters. We then propose an improvement on the inid model to capture the 'bursty' nature of channel errors, described by burst length distribution. Our theoretical analysis is supported by an experimental evaluation.

Key words: Trace, Heavy-tailed, Independent but not identically distributed (inid), Bit error model, Bursty
doi:10.1631/jzus.C1200175 **Document code:** A **CLC number:** TN911

1 Introduction

Bit transmission errors are inevitable in wireless communications. Analytical models of such errors are helpful for protocol design and performance evaluation in wireless networks. Generally, such analytical models are parameterized based on a trace, a binary sequence with a '1' denoting an incorrect reception bit, and a '0' denoting a correct reception bit. A sequence of error-free bits is called a run, and consecutive erroneous bits are called a burst. The ultimate goal of these analytical models is to generate a bit error sequence statistically similar to the trace. Therefore these analytical models are also called generative models.

Existing literature has proposed various analytical models. Each model has its own characteristics and emphasizes different aspects. The first and simplest model is the binary symmetric channel (BSC). The BSC model employs an independent Bernoulli experiment to determine the correctness of the bit. For all experiments, the fixed bit error probability p is used. The second category is based on Markov chains, including the well-known Gilbert-Elliot model (Gilbert, 1960), Fritchman models (SFMs) (Fritchman, 1967), and hidden Markov models (HMMs) (Garcia-Frias and Crespo, 1997). Some new approaches are proposed to im-

* Project supported by the National Natural Science Foundation of China (Nos. 61103010, 61103190, and 60803100), the National Basic Research Program (973) of China (No. 2012CB933500), and the High-Tech R&D Program (863) of China (No. 2012AA011001)

prove Markov-based models by proposing moderate increases in the model structure, such as the use of hierarchical models (Salih *et al.*, 2008; 2009) and the multi-level Markov model (M&M) (Kamthe *et al.*, 2009). Due to the computational complexity for parameter estimation, many models like the structured Markovian model (SMM) (Fernandes *et al.*, 2010) and Hamiltonian model (HM) (Qureshi *et al.*, 2011) are proposed to simplify the parameter estimation.

The third category is deterministic-process-based models, which stem from the second order statistics of fading envelope processes. These models are based on a properly parameterized and sampled deterministic process followed by a threshold detector and two parallel mappers. Although the deterministic-process-based models are ideal generative models in terms of providing realistic burst error statistics, they do not construct new error bursts in the process of generating error sequences (Wang and Xu, 2007).

The fourth category is based on the observation that the run length distribution of wireless channels is an intrinsically heavy-tailed distribution (Kopke *et al.*, 2003). Chaotic maps can be used to model such channel behavior by assuming that the wireless channel is in a 'bad' or 'good' state at time n. The chaotic maps model works as follows: in the good state all bits are correct; in the bad state all bits are erroneous. However, the chaotic maps model is complicated and the parameter estimation methods are difficult.

In addition, although simulation results of some existing models show heavy-tailed behaviors such as three layered hidden Markov models (3LHMMs) (Salih *et al.*, 2009) and deterministic-process-based models (Wang and Xu, 2007), these models lack theoretical proof of the heavy-tailed distribution. Therefore, a new error model is needed, which is simple for practical use and can capture heavy-tailed run length behaviors; the aim of the present paper is to fill this gap.

In this paper, we propose an independent but not identically distributed (inid) model which is simple for practical use but can still be derived from a heavy-tailed distribution in theory. To obtain natural properties of the wireless transmission errors, we set up an experimental wireless local area network (LAN) and capture all traces of the communications. The details of this setup are described in Section 2; it allows for the collection of a trace, a bit error sequence of ones and zeros, with a '1' denoting an error bit, and a '0' denoting a correctly received bit. Using this measurement setup, we investigate the complementary cumulative distribution function (CCDF) of the run length and discover that it behaves in a heavy-tailed manner which cannot be matched by simple error models (Kopke *et al.*, 2003).

Our main contribution is to show that an inid stochastic process with appropriate parameters is an appropriate representation of error processes in wireless channels, particularly when the wireless channel exhibits a heavy-tailed run length distribution. We present its parameterization process on the basis of a trace; we also explain in detail the intuitive meaning of these parameters. The generation algorithm is also provided to generate the simulated bit error sequences. To assess the suitability of the inid model, we compare the distribution functions of run lengths produced by the inid model and chaotic maps model with the trace. Theoretical analysis and simulation results show that the proposed model can properly describe the heavy-tail behavior of wireless channels.

2 Data collection and correlation analysis

2.1 Experimental setup and data collection

We set up an ad hoc wireless network environment that consists of three nodes, a transmitter node, a receiver node, and a monitoring node. For each node, we employ the TP-LINK WN650G wireless network card with Atheros chipsets, and the network card driver is MadWifi-0.9.4.

To collect traces, we modify ieee80211_monitor.c and ieee80211_proto.c contained in MadWifi-0.9.4 appropriately and set the wireless network card of the monitoring node in monitor mode. Concretely speaking, all received packets whose cyclic redundancy check (CRC) bits fail are dropped in the standard procedure referred to as ieee80211_monitor.c; we modify ieee80211_monitor.c to access an output function, ieee80211_dump_pkt() contained in ieee80211_proto.c and output these error packets.

All packets received in the monitoring node are marked with a serial number starting from zero. The monitoring node outputs only error packets and their corresponding serial numbers into a file errorTrace. We then write a program built with VC++6.0 to read these serial numbers and error packets from errorTrace and compute traces.

In the experiment, we fix the length of data packets and thus traces can be computed easily using serial numbers and the whole error packets. It is obvious that the serial number of the first error packet is the number of correct packets received by the monitoring node before the first error packet, and thus we output an all-zero sequence into the trace with a length of L_1, which is the product of the first serial number and the length of data packets. We then combine the first error packet and its corresponding correct packet by the bitwise '$\oplus$' to produce an error sequence and output it into the trace. Subsequently, we output an all-zero sequence into the trace with a length of L_2, which is the product of the length of data packets and the number of correct packets located between two consecutive error packets. The number of correct packets located between two consecutive error packets can be computed easily given serial numbers of these two consecutive error packets. For example, if the serial numbers are n_1 and n_2, the number of correct packets is $n_2 - n_1 - 1$. The same method is used repeatedly to produce the trace.

To simplify the calculation process, we use data packets with all data bytes set to 0x00. Owing to the scrambling procedure, it is not expected that the content of data packets has any significant impact on the experiment results (Han *et al.*, 2009). The traces were recorded in our laboratory and the distance between the transmitter node and the monitoring node was about 5 m. Experiments were conducted for different packet sizes in a variety of weather.

2.2 Correlation of time series based on traces

Given a trace, a run is defined as a string of consecutive zeros and a burst is defined as a string of consecutive ones. Therefore, a trace can be represented by the following time series:

$$l_1, e_1, l_2, e_2, \cdots, e_{n-1}, l_n, \tag{1}$$

where l_i is the run length and e_i is the burst length. Without loss of generality, we assume the first bit and the last bit contained in the trace are zeros.

When looking at the time series, we want to find out whether there is a degree of correlation in the time series, which can be described by a phase plot of successive burst lengths (e_{i-1} vs. e_i), run lengths (l_{i-1} vs. l_i), and run-burst lengths (l_i vs. e_i).

Different traces have different phase plots of successive burst lengths (e_{i-1} vs. e_i), run lengths (l_{i-1} vs. l_i), and run-burst lengths (l_i vs. e_i). However, they have some characteristics in common. We give three typical phase plots of successive burst lengths, run lengths, and run-burst lengths in Figs. 1–3, respectively.

A typical phase plot of successive burst lengths is illustrated in Fig. 1 using e_{i-1} as the coordinate axis X, e_i as the coordinate axis Y, and the number of (X, Y) as the coordinate axis Z. From Fig. 1, we see some artifacts that we do not expect: The first artifact is that a short burst is quite often followed by bursts of various lengths and most of them are also short bursts. The second artifact is that a long burst is followed by a very short burst. The third artifact is that the probability that two consecutive bursts are both short bursts, short-short bursts, is greater than short-long bursts and long-short bursts. In addition, it is obvious that the probability that two consecutive bursts are both long bursts, long-long bursts, is equal to zero.

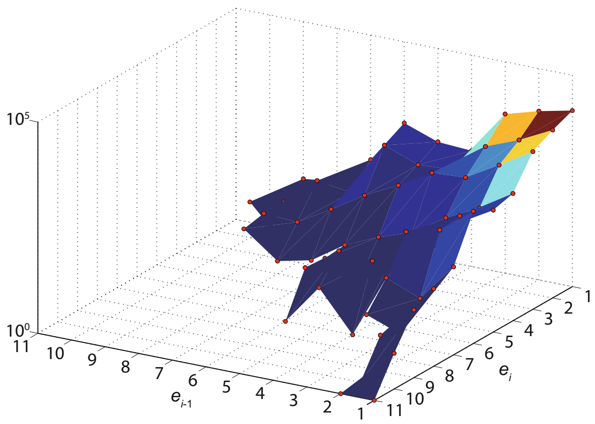

Fig. 1 Phase plot of successive burst lengths e_{i-1} vs. e_i, showing large proportions of short-short bursts, etc.

A typical phase plot of successive run lengths is illustrated by Fig. 2 using l_{i-1} as coordinate axis X and l_i as coordinate axis Y, which shows some deterministic structure, such as vertical and horizontal lines, etc. To be precise, given a short run with its length less than 10, it is very probable that the length of the next run is greater than 10, and vice versa.

Fig. 3 gives an example of the phase plot of successive run-burst lengths using l_i as the coordinate axis X and e_i as the coordinate axis Y. Fig. 3 shows that long runs are always followed by short bursts. For example, if the run length is greater than 100, the length of its corresponding burst is almost less than 7. Specifically, if the run length is greater than 10^6, the length of its corresponding burst is less than 4.

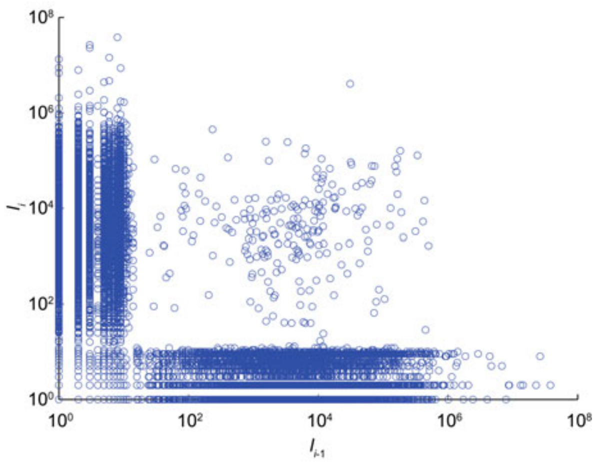

Fig. 2 Phase plot of successive run lengths l_{i-1} vs. l_i, showing deterministic structure: vertical and horizontal lines, etc.

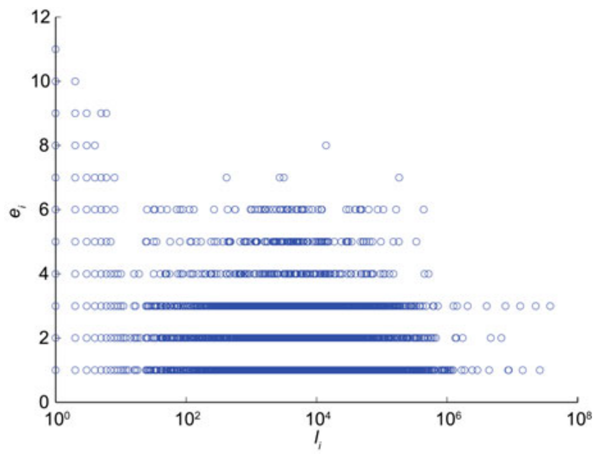

Fig. 3 Phase plot of successive run-burst lengths l_i vs. e_i showing that long runs are followed by short bursts

Strictly speaking, we should find or develop a model that is capable of expressing such correlation behaviors of successive burst lengths (e_{i-1} vs. e_i), run lengths (l_{i-1} vs. l_i), and run-burst lengths (l_i vs. e_i). Due to the ambiguity and complexity of such correlation behaviors, we consider these effects to be spurious and do not attempt to model them explicitly (Kopke *et al.*, 2003). In this paper, we focus only on the random distribution of the length of both runs and bursts, which are important measures of how errors models match wireless channels.

3 Inid error model

Given a trace, the set of all runs is available; we compute empirical CCDF of run length distribution by counting how many observed runs are greater than a particular run length. The length CCDF of a typical run is illustrated by Fig. 4 using a double logarithmic plot. This form is present in about two-thirds of our traces' run length distributions. A BSC model with bit error probability $q = 0.0011$ is provided to match the run length distribution of the trace.

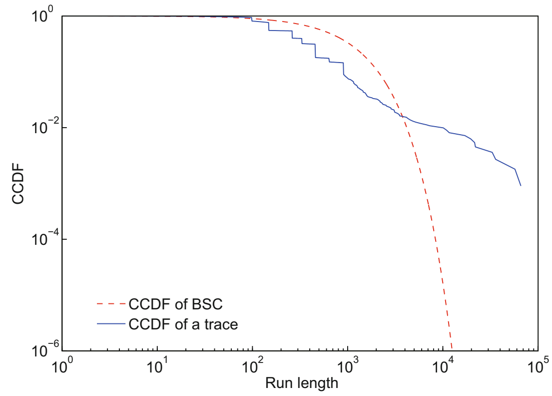

Fig. 4 Comparison of the trace's complementary cumulative distribution function (CCDF) with binary symmetric channel (BSC). The CCDFs of other simple error models can be found in Kopke *et al.* (2003)

In Fig. 4, the run length distribution of the trace shows two approximative straight lines with apparently different slopes. For values of L less than about 100 bits, the slope of the straight line is almost 0. The slope of another straight line is much less than 0 when L is greater than 100. In the proposed model, we employ two distinct models to describe both lines.

3.1 Motivation and basic ideas

From Fig. 4, it is obvious that when the run length is less than a threshold of length $L_{\text{threshold}}$, e.g., 100, the CCDF is describable using a BSC model. Therefore, in the proposed error model, an independent Bernoulli experiment, with fixed bit error probability q, is performed to determine the correctness of the bit when $L < L_{\text{threshold}}$.

However, when the run length is greater than $L_{\text{threshold}}$, the BSC model cannot match it and a CCDF that follows a power law is required to describe this type of behavior. One candidate model for describing such heavy-tailed behavior while requiring only one parameter is the inid stochastic process.

Given an inid stochastic process X, $X_i \in \{0, 1\}$ and $P(X_i = 1)=p_i$ $(i = 1, 2, \cdots)$, the CCDF of run length L can be computed by

$$\text{CCDF}(L) = \prod_{i=1}^{L}(1 - p_i). \quad (2)$$

On the other hand, if the CCDF of run length L for X is a heavy-tailed distribution, CCDF(L) follows a power law distribution:

$$\text{CCDF}(L) = kL^{-\alpha}, \quad (3)$$

where k is a positive constant. Together with Eq. (2), we have

$$p_L = 1 - \frac{L^{-\alpha}}{(L-1)^{-\alpha}}. \quad (4)$$

Therefore, the proposed error model is made up of a BSC model and an inid stochastic process. Its run length CCDF can be computed as follows when the run length $L \leq L_{\text{threshold}}$:

$$\text{CCDF}(L) = (1 - q)^L. \quad (5)$$

If the run length L is greater than $L_{\text{threshold}}$, the run length CCDF can be computed as follows:

$$\text{CCDF}(L) = (1-q)^{L_{\text{threshold}}} \prod_{i=L_{\text{threshold}}+1}^{L} (1-p_i). \quad (6)$$

In Fig. 5, we give an example of the CCDF for the BSC model with bit error probability $q = 0.0011$ and a power law distribution with $\alpha = 1.35$ and $k = 495$ using a double logarithmic plot. The proposed model with model parameters $\alpha = 1.35$ and $L_{\text{threshold}} = 100$ is plotted. Fig. 5 shows that the proposed model can match a heavy-tailed distribution perfectly.

From Fig. 5, it is obvious that the bit error probability of the inid error model, p_L, is greater than $q = 0.0011$ when $L \in [L_{\text{threshold}}, L_B]$, which makes the CCDF of the proposed model decrease faster than BSC's. When the run length L is greater than L_B, p_L is less than q, which makes the CCDF of the proposed model decrease extremely slowly. When the run length L is equal to L_B, p_L is equal to q. Together with Eq. (4), we have

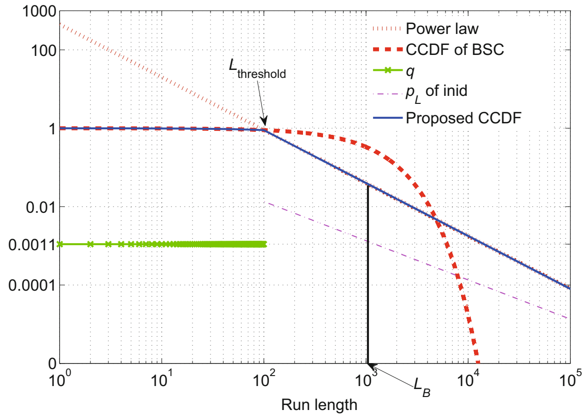

Fig. 5 Theoretical analysis of the proposed error model

$$L_B = \frac{1}{1 - (1 - q)^{1/\alpha}}. \quad (7)$$

The variable bit error probabilities of the inid error model, which follow a power law distribution, give rise to a heavy-tailed behavior for the proposed model.

3.2 Model parameterization

As described in Section 3.1, when the run length L is less than $L_{\text{threshold}}$, an independent Bernoulli experiment with a fixed bit error probability q is performed to determine the correctness of the bit; when the run length L is greater than $L_{\text{threshold}}$, an independent Bernoulli experiment with variable bit error probability p_L is performed.

Various values of p_L can be computed by a single parameter α (Eq. (4)). In addition, given a trace-based run length CCDF, $L_{\text{threshold}}$ and q follow Eq. (5). Therefore, actually the inid model has merely two parameters $\{L_{\text{threshold}}, \alpha\}$ or $\{q, \alpha\}$. Since $L_{\text{threshold}}$ has intuitive meanings in the double-logarithmic plot, we take $\{L_{\text{threshold}}, \alpha\}$ as model parameters.

The first step for the parameterization of the proposed error model is to plot run length CCDF based on traces. The run length CCDF in the double-logarithmic plot has a characteristic form: from a certain point, it follows a power law distribution and can be depicted by a straight line in the double-logarithmic plot. Its start point and slope are equal to the model parameters $L_{\text{threshold}}$ and α, respectively (Fig. 5).

To estimate the parameter α, we use the relationship between the trace's CCDF and α: $P_r[L > l] = kl^{-\alpha}$. Taking its logarithm shows that it is possible to estimate the parameter α from the slope of the least squares straight line fitted to the CCDF in the double-logarithmic plot:

$$\log P_r[L > l] = \log k - \alpha \log l. \quad (8)$$

We use simple linear regression to fit the run length CCDF and estimate model parameters α and k. Consequently, the different bit error probabilities p_L can be calculated by Eq. (4), which contains the only parameter α.

After obtaining an estimate of α and k, we plot $P_r[L > l] = kl^{-\alpha}$ in the double-logarithmic plot. There are many intersections of $P_r[L > l]$ and the run length CCDF. Generally, the horizontal coordinate of the first intersection of both curves is the estimate of $L_{\text{threshold}}$.

In the proposed model, when the run length L is less than $L_{\text{threshold}}$, an independent Bernoulli experiment is performed for each bit to determine the correctness of the bit. The fixed bit error probability q can be calculated as follows:

$$q = 1 - \text{CCDF}(L_{\text{threshold}})^{\frac{1}{L_{\text{threshold}}}}. \quad (9)$$

3.3 Generation algorithm

Once the model is parameterized, it can be used to generate the bit error sequence. The generating procedure, referred to as Algorithm 1, is initialized by an independent Bernoulli experiment with the bit error probability q. If the Bernoulli experiment returns 0, it produces a correct bit. Otherwise, an error bit is produced.

We record the current run length L. If the run length L is less than $L_{\text{threshold}}$, the same bit error probability q, given by Eq. (10), is used by an independent Bernoulli experiment to determine the correctness of the bit. If the run length L is greater than $L_{\text{threshold}}$, an independent Bernoulli experiment is performed with the bit error probability p_L, given by Eq. (4).

Algorithm 1 Generating a bit error sequence of maxLen bits using the inid model

Require: $0 < \alpha < 2, L_{\text{threshold}} > 0, q > 0$

$L \leftarrow 0$;
$i \leftarrow 0$;
while i <maxLen **do**
  if $L < L_{\text{threshold}}$ **then**
    Perform an independent Bernoulli experiment with bit error q to determine the correctness of the bit b_i;
  else
    $p_L \leftarrow 1 - \frac{L^{-\alpha}}{(L-1)^{-\alpha}}$;
    Perform an independent Bernoulli experiment with bit error p_L to determine the correctness of the bit b_i;
  end if
  if $b_i == 0$ **then**
    $L \leftarrow L + 1$;
  else
    $L \leftarrow 0$;
  end if
  $i \leftarrow i + 1$;
  if i ==maxLen **then**
    exit;
  end if
end while

4 Improved inid error model

As highlighted in the previous section, the proposed inid error model can match a heavy-tailed run length distribution perfectly. However, the proposed inid error model cannot capture the 'bursty' nature of channel errors observed in low level measurements. For example, the inid model can hardly produce a string like (01^t0) with $t > 0$ since the initial bit error probability q is very small.

We enhance the inid error model and employ Bayes' theorem to approximate such burst-error statistics behavior with a certain degree of accuracy. Since the maximum burst length is a small value such as 34 in Kopke *et al.* (2003) and 11 in this paper, it is possible to compute the probability that the next bit is 1 given a fixed burst. For example, given a burst with m bits, the conditional probability that the next bit is 1 $p(1|01^m)$ is calculated (Kandhway *et al.*, 2008):

$$p(1|01^m) = \frac{N(01^{m+1})}{N(01^m)} = \frac{N(01^{m+1})}{N(01^{m+1}) + N(01^m0)}, \quad (10)$$

where $N(01^m)$ is the number of the string '01^m' contained in the trace, which can be easily counted. Table 1 shows these probabilities calculated using the trace.

Table 1 Conditional probabilities that the next bit is 1 given a fixed burst length

Probability	Value	Probability	Value
$p(1\|01^1)$	0.4803	$p(1\|01^2)$	0.4558
$p(1\|01^3)$	0.3670	$p(1\|01^4)$	0.6908
$p(1\|01^5)$	0.6530	$p(1\|01^6)$	0.1393
$p(1\|01^7)$	0.4371	$p(1\|01^8)$	0.4173
$p(1\|01^9)$	0.1724	$p(1\|01^{10})$	0.2000
$p(1\|01^{11})$	0	$p(1\|01^{12})$	0

The generation algorithm of the improved inid error model is given in Algorithm 2. We use m to denote the burst length and if the burst length m is greater than 0, an independent Bernoulli experiment is performed with the bit error probability $p(1|01^m)$, given in Table 1.

Algorithm 2 Generating a bit error sequence of maxLen bits using the improved inid model

Require: $0 < \alpha < 2, L_{\text{threshold}} > 0, q > 0$
```
m ← 0;
L ← 0;
i ← 0;
while i <maxLen do
    if m > 0 then
        Perform an independent Bernoulli experiment
        with bit error p(1|01^m) to determine the cor-
        rectness of the bit b_i;
    else if L < L_threshold then
        Perform an independent Bernoulli experiment
        with bit error q to determine the correctness of
        the bit b_i;
    else
        p_L ← 1 − L^(−α) / (L−1)^(−α);
        Perform an independent Bernoulli experiment
        with bit error p_L to determine the correctness
        of the bit b_i;
    end if
    if b_i == 0 then
        m ← 0;
        L ← L + 1;
    else
        L ← 0;
        m ← m + 1;
    end if
    i ← i + 1;
    if i ==maxLen then
        exit;
    end if
end while
```

5 Comparison and discussion

After generating the bit error sequence using the proposed model, we count the run length CCDF and the burst length CCDF and compare them with the trace's CCDF. For comparison, the chaotic maps model is implemented. The parameters we employ are given in Table 2. The procedure for determining these parameters from a given trace is described in Kopke *et al.* (2003).

Table 2 Parameters for the chaotic map*

u_{g}	z_{g}	ϵ_{g}	u_{b}	z_{b}	ϵ_{b}
0.005	1.74	1.0^{-15}	5.83	1.67	0.0046

* Kopke *et al.* (2003)

From Fig. 6, it is obvious that both the inid model and the improved inid model can provide a good fit for the heavy-tailed run length distribution. Compared to chaotic maps (Kopke *et al.*, 2003), the proposed models need only two parameters and both have a clear intuitive meaning and can be calculated easily.

We also compare the burst length distribution. Fig. 7 demonstrates the almost perfect match of the burst length distribution achieved by the improved inid model. Since almost all of the burst lengths are equal to 1 for the inid model, we omit its burst length distribution. The chaotic maps model can be used to capture such behavior, but as depicted in Fig. 7, it cannot do it very well.

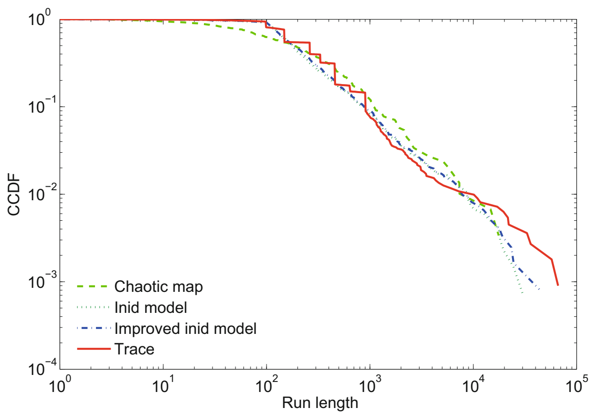

Fig. 6 Comparison of the run length and CCDF

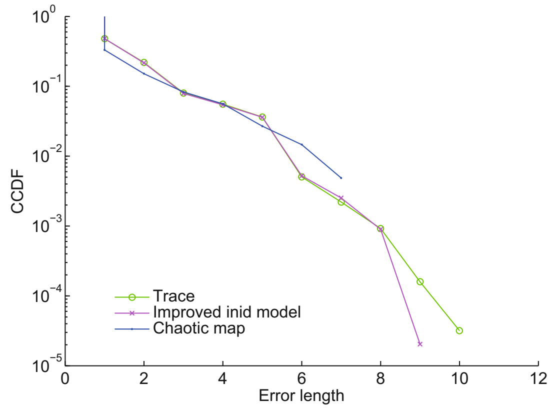

Fig. 7 Comparison of the error length and CCDF

6 Conclusions

This paper proposes an inid error model that is quite simple but maintains a certain degree of accuracy to characterize the heavy-tailed run length distribution in wireless communications. Along with the model, parameterization methods and generation algorithms are provided for practical use. Simulation results show that the proposed models match the actual trace and are much less complex than the classical chaotic maps model .

To match burst length distribution, we propose an improved inid error model that employs Bayes' theorem to approximate such burst-error statistics behaviors like (01^t0) with $t > 0$. Simulation results show that the improved inid error model is qualified to describe the random distribution of the lengths of both runs and bursts. However, the Bayes' theorem based method provides a weak expandability. Therefore, the extensibility is still one issue for future work.

References

Fernandes, V.S., Pinto, E.L., Grivet, M., 2010. A Novel Structured Markovian Model for Burst-Error Channels. Int. Conf. on Wireless and Mobile Communications, p.11-15. [doi:10.1109/ICWMC.2010.13]

Fritchman, B.D., 1967. A binary channel characterization using partitioned Markov chains. *IEEE Trans. Inf. Theory*, **13**(2):221-227. [doi:10.1109/TIT.1967.1053975]

Garcia-Frias, J., Crespo, P.M., 1997. Hidden Markov models for burst error characterization in indoor radio channels. *IEEE Trans. Veh. Technol.*, **46**(4):1006-1020. [doi:10.1109/25.653074]

Gilbert, E.N., 1960. Capacity of a burst-noise channel. *Bell Syst. Tech. J.*, **39**:1253-1265.

Han, B., Ji, L., Lee, S., Bhattacharjee, B., Miller, R.R., 2009. All Bits are not Equal? A Study of IEEE 802.11 Communication Bit Errors. IEEE Int. Conf. on Computer Communications, p.1602-1610. [doi:10.1109/INFCOM.2009.5062078]

Kamthe, A., Carreira-Perpian, M.A., Cerpa, A.E., 2009. M&M: Multi-level Markov Model for Wireless Link Simulations. Proc. 7th ACM Conf. on Embedded Networked Sensor Systems, p.57-70. [doi:10.1145/1644038.1644045]

Kandhway, K., Anand, A., Bhattacharjee, R., 2008. A Generative Model for Burst Error Characterization in a Fading Channel. Int. Technical Conf. of IEEE Region 10, the Asia Pacific Region, p.1-5. [doi:10.1109/TENCON.2008.4766506]

Kopke, A.K., Willig, A., Karl, H., 2003. Chaotic Maps as Parsimonious Bit Error Models of Wireless Channels. 22nd IEEE Annual Joint Conf. on Computer and Communications, p.513-523. [doi:10.1109/INFCOM.2003.1208702]

Qureshi, H.K., Ahmad, J.J., Khayam, S.A., Rakocevic, V., Rajarajan, M., 2011. Graph-theoretic complexity reduction for Markovian wireless channel models. *Wirel. Pers. Commun.*, **58**(4):831-849. [doi:10.1007/s11277-009-9908-8]

Salih, O.S., Wang, C.X., Laurenson, D.I., 2008. Double Embedded Processes Based Hidden Markov Models for Binary Digital Wireless Channels. IEEE Int. Symp. on Wireless Communication Systems, p.219-223. [doi:10.1109/ISWCS.2008.4726050]

Salih, O.S., Wang, C.X., Laurenson, D.I., 2009. Three Layered Hidden Markov Models for Binary Digital Wireless Channels. IEEE Int. Conf. on Communications, p.1-5. [doi:10.1109/ICC.2009.5199522]

Wang, C.X., Xu, W., 2007. A new class of generative models for burst error characterization in digital wireless channels. *IEEE Trans. Commun.*, **55**(3):453-462. [doi:10.1109/TCOMM.2007.892447]