Wireless Pers Commun (2013) 72:987–1004
DOI 10.1007/s11277-013-1051-x

Discrete-time Markov Model for Wireless Link Burstiness Simulations

Yantao Li · Daniel Graham · Gang Zhou · Xin Qi · Shaojiang Deng · Di Xiao

Published online: 7 March 2013

Abstract Link burstiness can negatively affect the performance of wireless networking protocols, by causing an extra of 15 % transmission cost. It describes the underlying behavior of packet delivery and provides insights into tuning protocols to improve performance. In this paper, we propose a discrete-time Markov model to simulate the burstiness behavior of wireless links, which provides a novel approach for link burstiness studies. More specifically, we first present a discrete-time Markov model with the input of β value and the output of a sequence trace of burstiness traffic. Then we design an algorithm to simulate the Markov model, where the state transition represents the packet receptions or losses. Finally, we evaluate the proposed model in terms of distribution of link burstiness, accuracy and cost, and the results demonstrate that our model is able to accurately simulate the burstiness behavior.

Y. Li (✉)
College of Computer and Information Science,
Southwest University, Chongqing 400715, China
e-mail: yantaoli@foxmail.com

Y. Li · D. Graham · G. Zhou · X. Qi
Department of Computer Science, The College of William and Mary, Williamsburg, VA 23187, USA
e-mail: yantaoli@cs.wm.edu

D. Graham
e-mail: dggraham@cs.wm.edu

G. Zhou
e-mail: gzhou@cs.wm.edu

X. Qi
e-mail: xqi@cs.wm.edu

S. Deng · D. Xiao
College of Computer Science,
Chongqing University, Chongqing 400044, China
e-mail: sj_deng@cqu.edu.cn

D. Xiao
e-mail: xiaodi_cqu@hotmail.com

Keywords Simulation · Low power wireless networks · Discrete-time Markov model · Link burstiness

1 Introduction

Low power wireless networks are becoming increasingly popular in our daily life, such as ZigBee, Bluetooth and WiFi networks, which provide lots of convenient applications including smart healthcare [1], bluetooth headphones and receiving/sending emails. However, such wireless links are examined to be bursty [2]. Link burstiness is a physical property, which means that packet transmissions sporadically shift between good delivery to poor delivery.

Link burstiness can negatively affect protocol performance: on packet reception rates of MAC protocols in a short time scale and on packet retransmissions of routing protocols in a short-time period as well [3–8]. In addition, it impacts experimental results. The underlying causes of burstiness are wireless channel variation and interference [2]. Therefore, the study of wireless link burstiness behavior is able to provide insights into the correlation of continuous packet deliveries and improve the performance of newly designed wireless protocols as well.

There are a number of related works on quantifying burstiness behavior in wireless links. These investigations can be classified into analytical models [9–22] and empirical studies [2,23–33]. For analytical models, Mushkin et al. in [11] propose a two-state Markov chain, which presents burstiness by μ, effectively describing how bit errors are correlated. In [14], Khayam et al. present Markov-based stochastic chains to model channel behavior for both bit errors and packet losses. For empirical studies, Srinivasan et al. in [2]present an empirical study to quantify the extent and characteristics of bursty links by defining a metric β, which measures the burstiness of a wireless link. The factor β is calculated by comparing conditional probability distribution functions (CPDFs) of packet transmissions on a bursty link to that of an independent link. Alizai et al. [27] use short-term estimation of wireless links to accurately identify short-term stable periods of transmission on bursty links. Although the above works provide analytical models or empirical studies, this work is different in that we present a novel Markov model for simulating bursty traces based on a given β value, which can be used for investigating the burstiness of wireless links. Compared to the β approach [2] in empirical studies, which can measure the burstiness behavior of wireless links, this work is able to take this β metric as the input and generate bursty traces that conform to the supplied β value.

In this paper, we present a novel discrete-time Markov model with the input of β value and the output of burstiness traffic traces. In the discrete-time Markov model, a state represents the number of successful or failure transmissions while state transition probability represents the conditional probability of successful or failure transmissions. This Markov model is then simulated by an algorithm, where the state transition represents the packet receptions or losses. For performance evaluation of the Markov model, we repeat 100 simulations of $1,000,000$ packet transmissions with "1" representing packet receptions and "-1" representing packet losses. More specifically, we first illustrate the distribution of simulated burstiness of 100 packet transmissions extracted from the simulations to show that with the increasing input of β value the burstiness is closer to the ideal burstiness. Then we compare the calculated β values of simulated bursty traces with the corresponding input of β value, to show that simulated β values are closer to β values with standard deviations of less than 0.0187.

The main contributions of this work can be summarized as:

- We are among the first to present a novel discrete-time Markov model with the input of a metric value β to simulate wireless burstiness behavior.

- We design an algorithm to simulate the Markov model, where the state transition represents the packet receptions (1 s) or losses (−1 s).
- We evaluate our Markov model through 100-run simulations of 1,000,000 packet transmissions for each input β. The evaluation results demonstrate that our model can accurately simulate the burstiness behavior of wireless links.

The remainder of this paper is organized as follows: Section 2 presents related work for exploring the burstiness behavior of wireless links. In Sect. 3, we describe the preliminaries of the calculation of β value and the discrete-time Markov model. Then, we discuss how the Markov model is derived and simulated in Sect. 4. Next, we evaluate our derived Markov model in Sect. 5 and present conclusions in Sect. 6.

2 Related Work

There are a wide range of works capturing the burstiness behavior of wireless links. We classify these efforts into two categories: analytical models [9–22] and empirical studies [2,23–33].

2.1 Analytical Models

The authors of [9] provide the Gilbert model, which is a probabilistic model for simulating burst noise in data transmission channels. In this model, the two-state hidden Markov model is used to generate noise burstiness. The first state has zero probability of encountering an error whereas the other state has a certain fixed non-zero probability for transmission errors. In [10], the authors measure packet burstiness in wired LANs through packet trains by building packet arrival models, which can be extended and used for packet burstiness in wireless links. The authors of [11] present the Gilbert-Elliott model, which is a two-state Markov chain describing burstiness in terms of μ, effectively depicting how bit errors are correlated. In this model, the probability of successful transmissions varies based on the current state, where the good state corresponds to 100 % packet reception and the bad state corresponds to zero packet reception. The burstiness parameter μ is calculated by $\mu = 1 - p_{gb} - p_{bg}$, where p_{gb} is the probability of transition from the good state to the bad state while p_{bg} is the probability of transition from the bad state to the good state. Thus, μ captures the correlation between the current and the previous packet delivery events. In [12], the authors present a new class of Markovian-based channel models, called bipartite models, which allow the user to freely choose the desired model complexity and therefore model accuracy. The authors of Khayam and Radha [14] evaluate and propose Markov-based stochastic chains to model the 802.11b MAC-to-MAC channel behavior for both bit errors and packet losses. They introduce an Entropy Normalized Kullback-Leibler measure to evaluate the performance of existing and new bit error and packet loss models. In [16], the authors propose two novel generative methods to model the end-to-end error profile of radio channels described by long well-defined error bursts interleaved with long error-free intervals, so that researchers can design good error-control schemes for bursty channels. In [17], the authors propose a novel multilevel approach involving hidden Markov models (HMMs) and Mixtures of multivariate Bernoullis (MMBs) for modeling the long and short time scale behavior of wireless links using experimental data traces collected from multiple 802.15.4 testbeds. In addition, the models in [14] and [17] for simulating burstiness are based on the hidden state continuous time Markov chain models. These hidden state Markov models simulate burstiness by transitioning among a collection of unobserved (hidden) states and then sporadically transitioning to an observed (non-hidden) state. Transitions between hidden states are not recognized as events but serve to introduce random delay between actual events which are represented by transitions to non-hidden states. The authors of [18] propose to derive wireless simulation models from experimental traces of radio signal strength and apply the closest-fit pattern matching (CPM) algorithm, which is originally designed for modeling external interference, to model signal strength. They find that using CPM for signal strength improves simulation of packet burstiness, reducing the Kantorovich–Wasserstein (KW) distance of conditional packet delivery functions (CPDFs) by a factor of about 3 for intermediate links. In [20], the authors develop a generic analytical model to study the packet loss burstiness. In the model, they use the correlation length of packet loss rate as a metric to represent the packet loss burstiness mathematically, and they formulate the metric by investigating the correlations between packet losses. The authors of Bas and Ergen [22] show a way of accurately analyzing the performance of bulk data dissemination protocols in WSNs. This model can be applied to practical network topologies by use of the shortest propagation path, and is accurate by considering topological information, impact of contention, and impact of pipelining.

2.2 Empirical Studies

In [2], the authors present β factor to measure wireless link burstiness at the packet level. In [23], the authors propose the Markov-based trace analysis (MTA) algorithm, which extracts stationary components from a collected trace. This analytical channel model can more accurately represent characteristics, such as burstiness, statistical distribution of errors, and packet loss processes compared with traditional approaches. The authors of [24] use Allan deviation of delivery rates calculated over different time intervals to find the characteristic burst length of a link. The Allan deviation is expressed as $\sqrt{\frac{1}{2n}\sum_{i=2}^{n}(x_i - x_{i-1})^2}$ [25], where the sample x_i is the fraction of packets delivered in successive intervals of a particular length. The Allan deviation will be high for interval lengths near the characteristic burst length. At smaller intervals, adjacent samples will change slowly, and the Allan deviation will be low. At longer intervals, each sample will tend towards the long-term average, and the Allan deviation will also be small. In [26], the packet reception rate (PRR) analysis shows that links are bursty rather than constant, i.e., their reception quality varies greatly from the overall PRR at different times. The authors provide a possible explanation for burstiness using wavelet analysis of RSSI (received signal strength indicator) traces from a variety of wireless links. The authors of Alizai et al. [27] use short-term estimation of wireless links to accurately identify short-term stable periods of transmission on bursty links. The approach allows a routing protocol to forward packets over bursty links if they offer better routing progress than long-term stable links. The authors of Munir et al. [28] propose a scheduling algorithm that produces latency bounds of the real-time periodic streams and accounts for both link burstiness and interference. The solution is achieved through the definition of a new metric $Bmax$ based on a significant empirical evidence of 21 days and over 3,600,000 packet transmissions per link. The $Bmax$ characterizes links by the maximum burst length and by choosing a novel least-burst-route that minimizes the sum of worst case burst lengths over all links in the route. In [29], the authors describe a burst-link-aware routing protocol $BLAR$ for wireless sensor networks, which discusses the impact of bursty links on routing protocol performance, and exploits an efficient burstiness identification method (BIM) to identify the existence of the bursty links by passively measuring the RSSI value of packets received from neighbors. It also presents a timely and adaptive link quality estimator $EasiLQE$, which uses an error-based filter and takes account into the bursty links identification. The authors of Alizai et al.

[31] propose two metrics, *Expected Future Transmissions* (EFT) and MAC_3, for runtime estimation of bursty wireless links. Based on these two metrics, they introduce the Bursty Link Estimator (BLE), which accurately estimates bursty links in the network rendering them available for packet transmissions. In [33], the authors present Strawman, a contention resolution mechanism designed for low-power duty-cycled networks that experience traffic bursts, which efficiently resolves network contention, mitigates the hidden terminal problem, and has zero overhead unless activated to resolve data collisions.

Unlike the above analytical models or empirical studies, our work is based on a discrete-time Markov chain with the input of β value for simulating bursty traces of wireless links.

3 Preliminaries

In this section, we first explicitly describe the calculation of the metric β proposed in [2], which is the input of our discrete-time Markov model. Then we briefly introduce the structure of a discrete-time Markov model. Finally, we interpret the notations used in this paper.

3.1 Description of the Metric β

In [2], Srinivasan et al. present an empirical study to quantify the extent and characteristics of the burstiness of wireless links by defining a metric β, which is based on CPDFs that are derived from packet delivery traces [34]. CPDFs give the probability that the next packet will be successfully received after n consecutive packet successes or failures. CPDFs can describe the ideal bursty link, which has one long burst of either successes or failures, as well as the independent link, in which there is no correlation between packet deliveries. The metric β is calculated by comparing CPDFs of the traffic on a bursty link to that of an independent link. Adapted from [2], β calculation of an example link is illustrated in Fig. 1 and can be formulated as Eq. 1:

$$\beta = \frac{mean(i_{-n/2}, ..., i_{n/2}) - mean(e_{-n/2}, ..., e_{n/2})}{mean(i_{-n/2}, ..., i_{n/2})} \tag{1}$$

where the distances $i_{-n/2}$ through $i_{n/2}$ are the distances between the CPDF elements of the independent link and the CPDF elements of the ideal bursty link, while the distances $e_{-n/2}$

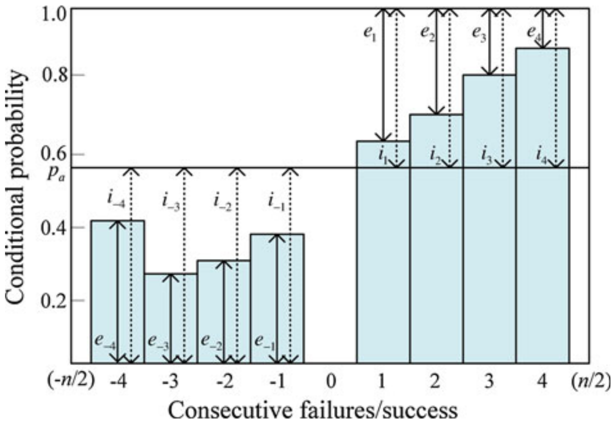


Fig. 1 β calculation of an example link

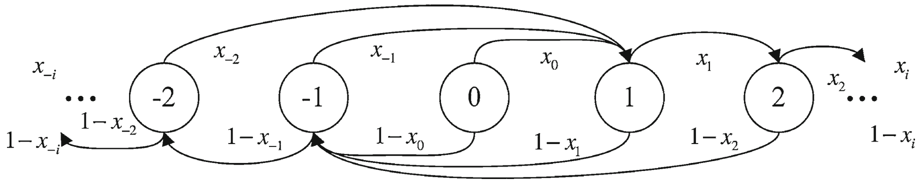


Fig. 2 An example of a discrete-time Markov Model

through $e_{n/2}$ are the distances between the CPDF elements of the corresponding example link and the CPDF elements of the ideal bursty link. Note that the subscripts of i or e are symmetric, thus we express the subscripts in an intuitive way, such as $-\frac{n}{2}$ and $\frac{n}{2}$. Moreover, n is the total number of packet transmissions including successes and failures. When n is an odd number, we use $\frac{n}{2}$ to denote $\lfloor\frac{n}{2}\rfloor$ and $-\frac{n}{2}$ to denote $-\lfloor\frac{n}{2}\rfloor$, respectively. Therefore, from Eq. 1, obviously a perfectly bursty link has a $\beta = 1$, while a link with independent deliveries has a $\beta = 0$. In addition, Fig. 1 provides an additional insight into how these CPDFs are used to calculate the metric β, where the x-axis represents the number of consecutive failures or successes that precedes a successful transmission and the y-axis represents the conditional probability of a successful transmission. Furthermore, p_a represents the average packet delivery ratio of n packet transmissions. For conditional probability, we explain it through an example. Let us consider a synthetic bursty trace $T : \{F, F, S, S, S, F, F, F, S, S, S, S\}$, where S represents a successfully delivered packet and F represents a failed or dropped packet. Assume that we calculate the conditional probability of a successful transmission under two consecutive failure transmissions. Notice that in the trace T we have two instances of two consecutive failures: $\{F, F, S\}$ and $\{F, F, F\}$, which means that the sample space is 2. Also notice that there is only one final successful transmission in the two instances. Therefore, the probability of successful transmission given that two failure transmissions occurred is the number of successful instances divided by the sample space.

3.2 Introduction of a Discrete-time Markov Model

As described in [35], a stochastic process $\{x(t), t \in T'\}$ is a Markov process if the future state of the process only depends on the current state of the process and not on its past history. Formally, a discrete-time Markov chain $\{x_t, t \in T'\}$ is a stochastic process whose state space is a finite or countably infinite set with index set $T' = \{0, 1, 2, ...\}$, which obeys $Pr[X_{k+1} = x_{k+1}|X_0 = x_0, ..., X_k = x_k] = Pr[X_{k+1} = x_{k+1}|X_k = x_k]$. The conditional probabilities $Pr[X_{k+1} = j|X_k = i]$ are called the transition probabilities of the Markov chain.

We use this discrete-time Markov chain to model the burstiness behavior in wireless networks. We plot an example of a discrete-time Markov model in Fig. 2. As depicted in Fig. 2, each circle denotes a state, which represents the number of successful or failure transmissions. For example, state -2 represents two consecutive failure transmissions while state 3 indicates three consecutive successful transmissions. Moreover, x_i denotes the transition probability from the state of $|i|$ consecutive successful transmissions (for $i > 0$) or failure transmissions (for $i < 0$) to the state of a successful transmission. In contrast, $1 - x_i$ denotes the transition probability from the state of $|i|$ consecutive successful transmissions (for $i > 0$) or failure transmissions (for $i < 0$) to the state of a failure transmission. The transition probability between states represents the conditional probability given a failure or successful transmission. These state transition probabilities are the same to the CPDFs used to calculate the metric β. In other words, the state transitions are equivalent to the height of

Table 1 Notations and their semantic meanings

Notations	Meanings
β	A metric to measure link bursty behavior
i_j, I_j	The distance between the jth CPDF element of the independent link and that of the ideal bursty link
e_j	The distance between the jth CPDF element of the example link and that of the ideal bursty link
n	The total number of packet transmissions including transmissions of failures and successes
x_i	The transition probability from the state of i consecutive transmissions to a successful transmission state
$x_+(x_-)$	The transition probability from the state of consecutive successful (failure) transmissions to a successful transmission state
$X_+(X_-)$	The transition probability sequences

the bar illustrated in Fig. 2. We introduce the Markov model in the following: at the beginning of the simulation, it starts from state 0; then, it transitions to state 1 with the probability x_0, which indicates that the packet is successfully delivered; if it does not transition to state 1, it automatically transitions to state -1 with transition probability $1 - x_0$, which indicates that the packet is not successfully delivered; after that, it will transition between non-zero states with different transition probabilities. For example, supposing that it is currently in state -2, with the probability x_{-2}, it transitions to state 1, which indicates that the packet is delivered after two consecutive failures. Otherwise, it dynamically transitions to state -3 with probability $1 - x_{-3}$, which means the packet transmission fails after three consecutive failures. We simplify this model in Sect. 4, which is the discrete-time Markov model used in our algorithm.

3.3 Notations

In this section, we list the notations used in this paper in Table 1.

4 The Discrete-Time Markov Model

In this section, we first derive the discrete-time Markov model from the calculation of β, and then we design an algorithm to simulate the Markov model with the input of the β value and the output of bursty traces.

4.1 Derivation of the Markov Model

In order to relate the properties of the Markov chain to the variables in Eq. 2, we attempt to derive a discrete-time Markov model that can generate bursty traffic.

Thus, we begin by rewriting Eq. 1 in a different way, expressed in Eq. 2 as:

$$\beta = \frac{\frac{1}{n}\sum_{i=-n/2}^{n/2} I_i - e_i}{\frac{1}{n}\sum_{i=-n/2}^{n/2} I_i} \tag{2}$$

where n denotes the total number of packet transmissions including transmissions of failures or successes, which is associated to x-axis in Fig. 1, and $i \in [-\frac{n}{2}, -1] \cup [1, \frac{n}{2}]$ indicates the

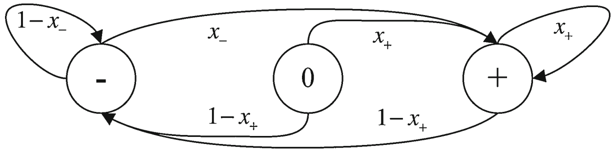


Fig. 3 The simplified discrete-time Markov Model

number of consecutive failure transmissions (for $i < 0$) or consecutive successful transmissions (for $i > 0$). For example, $i = -2$ represents two consecutive failure transmissions, while $i = 3$ indicates three consecutive successful transitions. For $i = 0$, we define it as "no packet transmission", which is the initial state of the Markov model. Furthermore, i is valued in a symmetric range $[-\frac{n}{2}, \frac{n}{2}]$ in our Markov model, where we use $-\frac{n}{2}$ to denote $-\lfloor\frac{n}{2}\rfloor$ and $\frac{n}{2}$ to denote $\lfloor\frac{n}{2}\rfloor$, respectively, when n is an odd number. In addition, I_i is equivalent to i_i in Eq. 1.

Then, we deduce the mean value of I ($\bar{I} = \frac{1}{n}\sum_{i=-n/2}^{n/2} I_i$) to its simplest form. After reviewing Fig. 1, we notice that all the positive instances share the same I value and the same is true for negative instances. Therefore, we separate all the values $I_i (i \in [-\frac{n}{2}, -1] \cup [1, \frac{n}{2}])$ to values $I_i^- (i \in [-\frac{n}{2}, -1])$ and $I_i^+ (i \in [1, \frac{n}{2}])$. Then, with the two substitutions, $\bar{I}$ can be rewritten as:

$$\bar{I} = \frac{1}{n}\sum_{i=-n/2}^{n/2} I_i = \frac{1}{n}\left\{\sum_{i=-n/2}^{-1} I_i^- + \sum_{i=1}^{n/2} I_i^+\right\} \tag{3}$$

As illustrated in Fig. 1, I_i^- is assigned the value of p_a, and naturally I_i^+ will be assigned the value of $1 - p_a$. Then, with these substitutions, $\bar{I}$ can be further rewritten as:

$$\bar{I} = \frac{1}{n}\sum_{i=-n/2}^{n/2} I_i = \frac{1}{n}\left\{\sum_{i=-n/2}^{-1} I_i^- + \sum_{i=1}^{n/2} I_i^+\right\} = \frac{1}{n}\sum_{i=-n/2}^{n/2} (p_a + (1 - p_a)) = \frac{1}{2} \tag{4}$$

We replace all instances of $\frac{1}{n}\sum_{i=-n/2}^{n/2} I_i$ in Eq. 2 by the constant $\frac{1}{2}$ according to Eq. 4. Thus, with the substitution, Eq. 2 can be rewritten as:

$$\beta = \frac{\frac{1}{2} - \frac{1}{n}\sum_{i=-n/2}^{n/2} e_i}{\frac{1}{2}} \tag{5}$$

We simply transform Eqs. 5 to 6 for further analysis:

$$\frac{1}{n}\sum_{i=-n/2}^{n/2} e_i = \frac{1-\beta}{2} \tag{6}$$

Next, we attempt to use Eq. 6 to derive the state transitions of a simplified descrete-time Markov model, which is illustrated in Fig. 3. Thus, we first introduce the markov model used in this paper and then explore the relationship between the e_i terms in Fig. 1 and transition probabilities in Fig. 3.

As illustrated in Fig. 3, the circle with number "0" denotes state 0, which is the initial state and represents no transmission. The circle with symbol "−" denotes all the negative states, which represent the consecutive failure transmissions, while the circle with symbol "+" denotes all the positive states, which represent the consecutive successful transmissions.

Moreover, x_+ denotes the transition probability from the state of consecutive successful transmissions to the state of another successful transmission and $1 - x_+$ represents the transition probability from the state of consecutive successful transmissions to the state of a failure transmission. In contrast, x_- denotes the transition probability from the state of consecutive failure transmissions to the state of a successful transmission and $1 - x_-$ represents the transition probability from the state of consecutive failure transmissions to the state of another failure transmission. In the Markov model, it starts from state 0, and with the probability x_+ it transitions to positive state. Otherwise, it transitions to negative state with transition probability $1 - x_+$. If it is currently in negative state, it transitions to positive state with probability x_- or it automatically transitions to negative state with probability $1 - x_-$. On the other hand, if it is currently in positive state, it transitions to positive state with probability x_+ or it automatically transitions to negative states with probability $1 - x_+$. Therefore, except the initial state, there are only two states: the positive state and the negative state. Thus, there are four state transitions: from the positive state to the positive state, from the positive to the negative, form the negative to the negative and from the negative to the positive and the corresponding transition probabilities are: x_+, $1 - x_+$, $1 - x_-$ and x_-.

We explore the relationship between e_i and transition probability by defining a new state transition probability x_i, which represents the probability of a successful transition out of state i to a higher state, where a higher state represents a state of a successful transmission from the current state. We express x_i in terms of e_i as:

$$x_i = \begin{cases} e_i & i < 0 \\ 1 - e_i & i > 0 \end{cases} \tag{7}$$

Since this value x_i only describes the probability of transition to higher states, we also need to describe the probability for transition to a lower state, where a lower state represents a state of a failure transmission from the current state. From current state, there are only two possible state transitions: to a higher state or a lower state. Therefore, the state transition probability to lower states can be calculated as: $1 - x_i$. Remember that we intent to end up with a model in terms of β, which means that we must find some way of deriving these x_i values in terms of a single value β. We achieve this by making two assumptions: (i) the values of x_i are all the same for all the negative states and the same for all the positive states; (ii) the markov chain is symmetric and contains an equal number of positive and negative states. With these assumptions, we only have to derive two values of x_i: one for positive state denoted as $x_{+|i|}$ and the other for negative state denoted as $x_{-|i|}$. Accordingly, as expressed in Eq. 7, we also separate the values of e_i into two components: one component representing the values of e_i for the negative states denoted as $e_{-|i|}$ and the other representing the values of e_i for the positive states denoted as $e_{+|i|}$. Furthermore, as illustrated in Fig. 1, the increment of indexes of e_i from $-\lfloor \frac{2}{n} \rfloor$ to -1 means that there is a successful delivery after $|i|$ consecutive failure transmissions, which can be explained as that the states currently transition to higher states. Therefore, the values of $e_{-|i|}$ are equivalent to $x_{-|i|}$, which is consistent to the Markov model in Fig. 2. On the other hand, the increment of indexes of e_i from 1 to $\lfloor \frac{n}{2} \rfloor$ means that there is a successful delivery after i consecutive successful transmissions, which can be explained as that the states currently transition to higher states as well. Thus, the values of $e_{+|i|}$ are equivalent to $1 - x_{+|i|}$. We can now use these equivalents to express e_i in the left part of Eq. 6 in terms of $x_{+|i|}$ and $x_{-|i|}$ as:

$$\frac{1}{n} \sum_{i=-n/2}^{n/2} e_i = \frac{1}{2} \left\{ \frac{2}{n} \sum_{i=-n/2}^{-1} x_{-|i|} + \frac{2}{n} \sum_{i=1}^{n/2} (1 - x_{+|i|}) \right\} \tag{8}$$

Algorithm 1: The generation of transition probability

Input: β value
Output: The sequences of transition probabilities of X_+ and X_-

$X_+(MaxIterationNumber) \leftarrow [0, 0, ..., 0]$
$X_-(MaxIterationNumber) \leftarrow [0, 0, ..., 0]$
for $i = 1$ ***to*** $MaxIterationNumber$ **do**
  if $\beta < x_+ < 1$ *and* $0 < x_- < 1$ **then**
    $x_+ \leftarrow rand()$
    $x_- \leftarrow -\beta + x_+$
  end
  $X_+(i) \leftarrow x_+$
  $X_-(i) \leftarrow x_-$
end
return X_+ *and* X_-

Recalling the assumptions that all the values of $x_{-|i|}$ are the same and all the values of $x_{+|i|}$ are the same as well, we use x_- to denote $x_{-|i|}$ and use x_+ to denote $x_{+|i|}$, which are consistent with the Markov model in Fig. 3. Then, with the substitutions, Eq. 8 can be rewritten as:

$$\frac{1}{n}\sum_{i=-n/2}^{n/2} e_i = \frac{1}{2}\left\{\frac{2}{n}\frac{n}{2}x_- + \frac{2}{n}\frac{n}{2}(1-x_+)\right\} = \frac{1}{2}\left(x_- - x_+ + 1\right) \tag{9}$$

By substituting Eq. 9 into Eq. 6, we obtain:

$$\frac{1}{2}\left(x_- - x_+ + 1\right) = \frac{1-\beta}{2} \tag{10}$$

Finally, we obtain the positive and negative state transitions of Markov model with the input of β as:

$$x_- = -\beta + x_+ \tag{11}$$

where x_+ denotes the transition probability from positives states to higher states, while x_- denotes transition probability from negative states to higher states.

4.2 Simulation of the Markov Model

In this section, we present how to generate a bursty trace based on the deduced Markov model with the input of β. We design an algorithm to simulate the model, which is composed of Algorithms 1 and 2.

With Algorithm 1, we generate sequences of transition probabilities (x_+ and x_-) by running the Markov model $MaxIterationNumber$ times, where $MaxIterationNumber$ represents the maximum iteration numbers, with the input of β value. Then, the transition probability sequences X_+ and X_- are used as the inputs of Algorithm 2 to generate bursty traces, which are represented by 1 or -1. "1" represents a transition to a positive state from the current state since there is a successful transmission, while "-1" represents a transition to a negative state since there is a failure transmission. Within Algorithm 2, we start with the state $start$, where we compare the randomly generated value r with a decimal value 0.5. If $r < 0.5$ the current state is updated to 1; otherwise it is updated to -1. Then, for the remaining $MaxIterationNumber - 1$ iterations, the state transitions between negative and positive states: if the current state is positive and satisfies $r < 1 - x_+$, it transitions to the

Algorithm 2: Bursty traffic simulation through state transition

Input: The sequences of transition probabilities of X_+ and X_-
Output: The sequence of bursty traffic behavior B

```
start ← true
state ← 0
B(MaxIterationNumber) ← [0, 0, ..., 0]
for i = 1 to MaxIterationNumber do
    r ← rand()
    if start then
        if r < 0.5 then
            state ← 1
        end
        else
            state ← −1
        end
        start ← false
    end
    else
        if state==1 then
            if r < 1 − x+ then
                state ← −1
            end
        end
        else
            if r < x− then
                state ← +1
            end
        end
    end
    B(i) ← state
end
return B
```

negative state. Otherwise, if the current state is negative and satisfies $r < x_-$, it transitions to the positive state. These state transitions are stored in a sequence B, which is the trace of packet receptions (1s) and losses (-1s) for describing the distribution of a bursty link. For example, a bursty traffic $B = 11 - 1 - 1 - 11$ represents two consecutive successful transmissions, three consecutive failure transmissions and then one successful transmission.

Therefore, through Algorithms 1 and 2, we simulate a bursty traffic with a length of *MaxIterationNumber*.

5 Performance Evaluation

In this section, we evaluate the discrete-time Markov model in terms of distribution of link burstiness, accuracy, statistical analysis, and cost analysis.

5.1 Distribution of Link Burstiness

In this section, we illustrate the distribution of simulated burstiness of wireless links generated by the discrete-time Markov model. For this purpose, we conduct the following experiment:

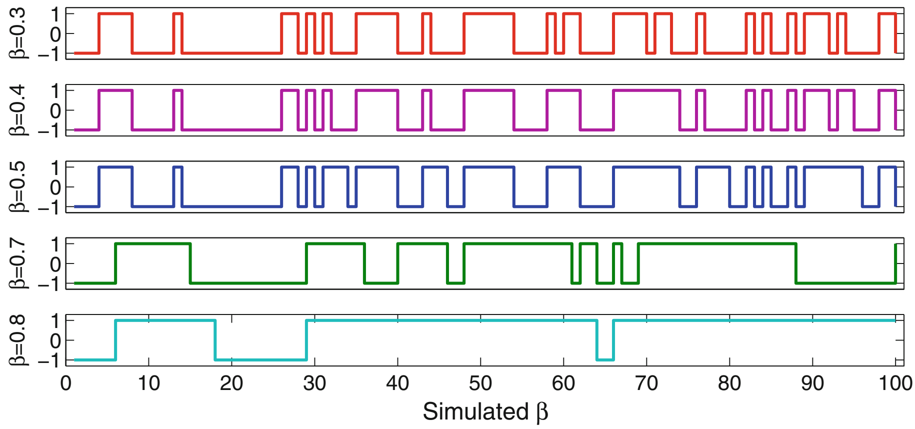


Fig. 4 Link burstiness simulation with inputs of β values

with the input of β value, we call Algorithms 1 and 2 with $MaxIterationNumber = 100$ and obtain the outputs of a sequence of simulated bursty traffic represented by B with 100 elements. In the experiment, we assign input values as: $\beta = \{0.3, 0.4, 0.5, 0.7, 0.8\}$ and the corresponding simulated bursty sequences are illustrated in Fig. 4, where the x-axis represents the sequence of a simulated bursty traffic (packet receptions (1 s) or losses (−1 s)), while the y-axis indicates the corresponding β values. Recall that a link with independent deliveries has a $\beta = 0$, while a perfectly bursty link has a $\beta = 1$ [2]. As illustrated in Fig. 4, with the increase of β values the burstiness of the simulated links are closer to ideal bursty links. More specifically, comparing the sequence distribution of $\beta = 0.3$ with $\beta = 0.8$, obviously, the distribution of $\beta = 0.8$ shows more burstiness after the 30th packet transmission. On the other hand, the distribution of $\beta = 0.4$ shows more burstiness than that of $\beta = 0.3$ from the 66th to 75th packet transmissions.

5.2 Accuracy Evaluation

In this section, we evaluate the accuracy of simulated link burstiness by comparing the calculated β values of simulated bursty links with the inputted β values. For this end, we conduct the following experiment: first, with the input of $\beta = 0.2$, we call Algorithms 1 and 2 with $MaxIterationNumber = 1,000,000$ and generate a 1,000,000-element B, and then we gradually increase β to 0.89 with an increment of 0.01. Second, we repeat the experiment 100 times. Finally, we calculate the corresponding β values for simulated B. Therefore, for each input of β assigned values from 0.2 to 0.89 with the increment of 0.01, there are 100 calculated β values of simulated bursty links. For easy understanding, we term calculated β value of simulated bursty links as simulated β or "$\bar{\beta}$", in terms of β.

In Fig. 5, we plot the scattergraph to show the distribution of simulated β values, where "$y = x$" line represents the inputs of β values, while *dots* represent the distribution of simulated β values. As illustrated in Fig. 5, simulated β values distribute closely around "$y = x$" line, which demonstrates that the simulated β values are very close to the input values β. That is to say, our Markov model can simulate the good characteristic of burstiness of wireless links. Furthermore, with the increase of β values, *dots* are closer to "$y = x$" line.

In Fig. 6, we plot the cumulative distribution function (CDF) of simulated β values for the Markov model with inputs of $\beta = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$. The figure demonstrates

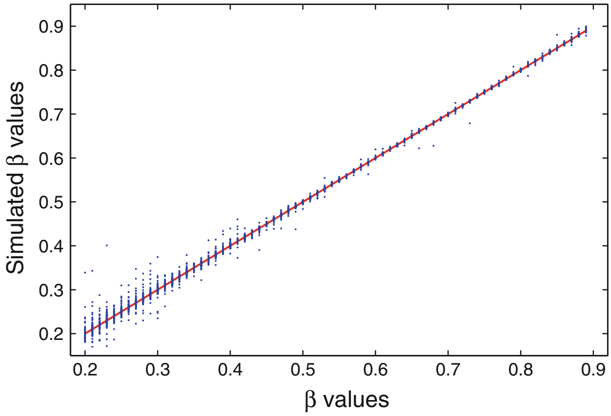


Fig. 5 Scattergraph of distributions of simulated β values

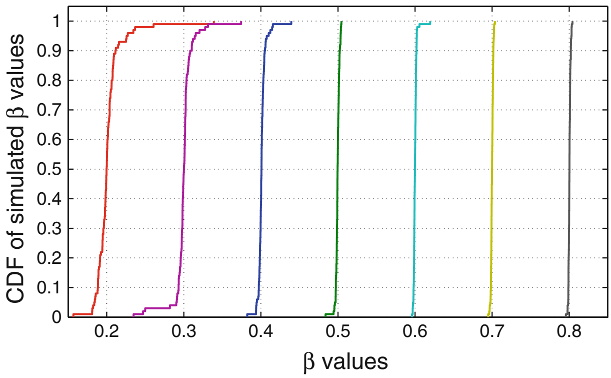


Fig. 6 CDF of simulated β values of bursty links

that the simulated β values are very close to corresponding β values. For $\beta = 0.2$, more than 20 % simulated β values are equal to 0.2, while for $\beta = 0.8$ more than 80 % simulated β values are equal to 0.8. In addition, the same trend with the scattergraph is that more percentage of simulated β values are equal to corresponding β values as β value increases.

In Fig. 7, we quantify the absolute errors of simulated β values, which are calculated by: $|\text{mean}(\text{simulated}\beta\text{value}) - \beta \text{ value}|$. As the figure illustrated, the largest absolute error is 0.0034 while the smallest one is zero, and with the increase of the β value, the absolute errors decrease. That is to say, the simulated β value is extremely close to β values, which demonstrates that our Markov model can simulate excellent burstiness of wireless links.

Therefore, based on the above analysis, the proposed Markov model is able to accurately simulate the burstiness of wireless links.

5.3 Statistical Analysis

In order to capture the statistical characteristics of our discrete-time Markov model, we redo the above experiment and compute the statistics: mean values $\bar{\beta}'$,average errors$\bar{\delta}$, standard

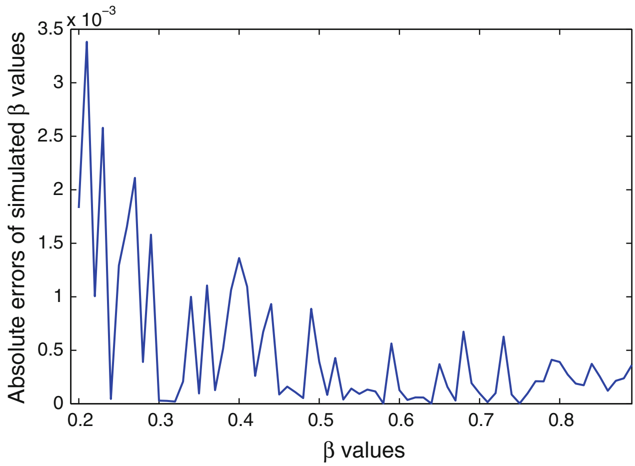


Fig. 7 Absolute errors of simulated β values

Table 2 Statistics of simulated burtiness of wireless links (β')

β	$\bar{\beta}'$	$\bar{\delta}$	σ	$\bar{E}$ (%)
0.2	0.2018	0.0096	0.0187	4.77
0.3	0.3000	0.0068	0.0142	2.27
0.4	0.4014	0.0031	0.0056	0.77
0.5	0.4996	0.0017	0.0025	0.34
0.6	0.6001	0.0014	0.0025	0.22
0.7	0.6999	0.0011	0.0014	0.15
0.8	0.8004	0.0010	0.0013	0.12

deviation σ, and average relative errors$\bar{E}$. They are defined as: $\bar{\beta}' = \frac{1}{N}\sum_{i=1}^{N}\beta_i'$, $\bar{\delta} = \frac{1}{N}\sum_{i=1}^{N}|\beta_i' - \bar{\beta}'|$, $\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\beta_i' - \bar{\beta}')^2}$, $\bar{E} = (\bar{\delta}/\bar{\beta}') \times 100\,\%$, where N=100.

As we can see in Table 2, the mean values $\bar{\beta}'$ are very close to the input values β. More specifically, with the increase of β, the average errors $\bar{\delta}$, standard deviation σ, and average relative errors $\bar{E}$ gradually decrease. Therefore, the Markov model can statistically simulate the burstiness of wireless links.

5.4 Cost Analysis

In this section, we analyze the cost of the discrete-time Markov model in terms of computational complexity, space complexity, and execution time. As described in Algorithms 1 and 2, we iterate them $MaxIterationNumber$ times to generate bursty behavior. Therefore, the computational complexity of the model is $\mathcal{O}(MaxIterationNumber)$. In experiments, we implement both algorithms on Visual C++ on a laptop with Intel Centrino 2.4GHz processor and 1.0 GB memory.As expressed in the C++ code, both algorithms only take 14.8 KB space on disc. Based on this platform, the experiment of distribution of link burstiness takes only 25 s while that of accuracy evaluation spends less than 2 min.

6 Conclusions

To the best of our knowledge, there is no existing model that allows researchers to simulate burstiness behavior by simply inputting a β value. In order to address this problem, we propose a discrete-time Markov model to simulate the burstiness behavior of wireless links, which allows researchers to simulate different bursty conditions and evaluate protocols that cope with busty links. More specifically, we first present a discrete-time Markov model with the input of β value and the output of a trace of burstiness traffic. Then we design an algorithm to simulate the Markov model, where the state transition represents the packet receptions or losses. Finally, we evaluate the model with 100-run simulations of 1,000,000 packet transmissions for each input of β, and the results demonstrate that our proposed model is able to accurately simulate the burstiness behavior, with the standard deviation of simulated β and β of less than 0.0187.

Acknowledgments This work was supported in part by US National Science Foundation (Grant nos. ECCS-0901437, CNS-0916994), the National Natural Science Foundation of China (Grant nos. 61173178, 61070246, 61003247), the Natural Science Foundation Project of CQ CSTC (Grant No. 2011jjjq40001), the Program for New Century Excellent Talents in University of China (Grant nos. NCET-09-0838, NCET-08-0603), the Fundamental Research Funds for the Central Universities (Grant nos. CDJZR10180002, CDJZR10180003, CDJXS12180004), and the Natural Science Foundation Project of CQ CSTC (Grant nos. 2010BB2047, 2010BB2210, 2009BB2211).

References

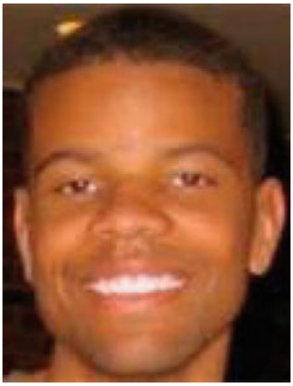
1. Huang, H., Sun, Y., Yang, Q., Zhang, F., Zhang, X., Liu, Y., Ren, J., & Sierra, F. (2010). Integrating neuromuscular and cyber systems for neural control of artificial legs. In *IEEE/ACM First International Conference on Cyber-Physical Systems* (ICCPS'10), pp. 129–138, April, 2010.
2. Srinivasan, K., Kazandjieva, M. A., Agarwal, S., & Levis, P. (2008). The β-factor: Measuring wireless link burstiness. In *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems* (SenSys'08), pp. 29–41, November 2008.
3. Woo, A., Tong, T., & Culler, D. (2003). Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems* (SenSys'03) November 2003.
4. Karner, W., Nemethova, O., Svoboda, P., & Rupp, M. (2007). Link error analysis and modeling for video streaming cross-layer design in mobile communication networks. *ETRI Journal*, *29*(5), 25–36.
5. Gandikota, V. R., Tamma, B. R., & Murthy, C. S. R. (2008). Adaptive FEC-based packet loss resilience scheme for supporting voice communication over ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, *7*(10), 1184–1199.
6. Becher, A., Landsiedel, O., Kunz, G., & Wehrle, K. (2008). Towards short-term wireless link quality estimation. In *Proceedings of the Fifth Workshop on Embedded Networked Sensors* (HotEmNets'08) pp. 1–5, June 2008.
7. Vutukuru, M., Balakrishnan, H., & Jamieson, K. (2009). Cross-layer wireless bit rate adaptation. In *Proceedings of the 2009 Conference on Applications, Technologies, Architectures, and Protocols for Computer, Communication* (SIGCOMM'09) August 2009.
8. Aggarwal, R., Schniter, P., & Koksal, C. (2009). Rate adaptation via link-layer feedback for goodput maximization over a time-varying channel. *IEEE Transactions on Wireless Communications*, *8*(8), 4276–4285.
9. Gilbert, E. N. (1960). Capacity of a burst-noise channel. *Bell System Technical Journal*, *39*, 1253–1266.
10. Jain, R., & Routhier, S. (1986). Packet trains-measurements and a new model for computer network traffic. *IEEE Journal on Selected Areas in Communications*, *4*(6), 986–995.
11. Mushkin, M., & Bar-David, I. (1989). Capacity and coding for the Gilbert-Elliott channels. *IEEE Transactions on Information Theory*, *35*, 1277–1290.
12. Willig, A. (2002). A new class of packet- and bit-level models for wireless channels. In *Proceedings of the 13th IEEE International Symposium On Personal, Indoor and Mobile Radio, Communications* (PIMRC'02), pp. 2434–2440, September 2002.

13. Kopke, A., Willig, A., & Karl, H. (2003). Chaotic maps as parsimonious bit error models of wireless channels. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM'03), pp. 513–523, March–April, 2003.
14. Khayam, S. A., & Radha, H. (2003). Markov-based modeling of wireless local area networks. In *Proceedings of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems* (MSWiM '03), pp. 100–107, September, 2003.
15. Wang, C. X., & Patzold, M. (2004). A generative deterministic model for digital mobile fading channels. *IEEE Communications Letters*, *8*(4), 223–225.
16. Zhu, W., & Garcia-Frias, J. (2004). Stochastic context-free grammars and gidden Markov models for modeling of Bursty channels. *IEEE Transactions on Vehicular Technology*, *53*(3), 666–676.
17. Kamthe, A., Carreira-Perpinan, M. A., & Cerpa, A. E. (2009). M&M: Multi-level Markov model for wireless link simulations. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems* (SenSys'09), pp. 57–70, November, 2009.
18. Rusak, T., & Levis, P. (2008). Investigating a physically-based signal power model for robust low power wireless link simulation. In *Proceedings of the 11th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems* (MSWiM '08), pp. 37–46, October 2008.
19. Cai, L., Shen, X., & Mark, J. W. (2009). *Multimedia Services in Wireless Internet: Modeling and Analysis*. New York: Wiley.
20. Liu, F., Fan, Y. Shen, X., Lin, C., & Zeng, R. (2010). An Analytical Model to Study the Packet Loss Burstiness over Wireless Channels. In *Proceedings of IEEE Global, Communications Society* (Globecom'10), pp. 1–5, December 2010.
21. Dong, W., Chen, C., Liu, X., Teng, G., Bu, J., & Liu, Y. (2011). Bulk data dissemination in wireless sensor networks: Modeling and analysis. *Computer Networks*, *56*(11), 2664–2676.
22. Bas, C. U., & Ergen, S. C. (2012). Spatio-temporal characteristics of link quality in wireless sensor networks. In *Proceedings of 2012 IEEE, Wireless Communications and Networking Conference* (WCNC'12), pp. 1152–1157, April, 2012.
23. Konrad, A., Zhao, B., Joseph, A., & Ludwig, R. (2003). A Markov-based channel model algorithm for wireless networks. *Wireless Networks*, *9*, 189–199.
24. Aguayo, D., Bicket, J., Biswas, S., Judd, G., & Morris, R. (2004). Link-level measurements from an 802.11b mesh network. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for, Computer Communications* (SIGCOMM'04), pp. 121–132, August–September, 2004.
25. Allan, D. W. (1987). Time and frequency (time domain) characterization, estimation and prediction of precision clocks and oscillators. *IEEE Transactions on UFFC*, *34*(6), 647–654.
26. Rusak, T., & Levis, P. (2009). Burstiness and scaling in the structure of low-power wireless links. *ACM SIGMOBILE Mobile Computing and Communications Review*, *13*(1), 60–64.
27. Alizai, M. H., Landsiedel, O., Link, J. A. B., Gotz, S., & Wehrle, K. (2009). Bursty traffic over bursty links. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems* (SenSys'09), pp. 71–84, November 2009.
28. Munir, S., Lin, S., Hoque, E., Nirjon, S. M. S., Stankovic, J. A., & Whitehouse, K. (2010). Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks* (IPSN '10), pp. 303–314, April 2010.
29. Huang, T., Li, D., Zhang, Z., & Cui, L. (2010). Design of a burst-link-aware routing protocol in wireless sensor networks. In *IET 2010 International Conference on, Wireless Sensor Network* (IET-WSN'10), pp. 374–379, November 2010.
30. Liu, Y., He, Y., Li, M., Wang, J., Liu, K., & Mo, L. (2011). Does wireless sensor network scale? A Measurement study on GreenOrbs. In *Proceedings of the 30th IEEE International Conference on, Computer Communications* (INFOCOM'11) pp. 873–881, March 2011.
31. Alizai, M. H., Wirtz, H., Kunz, G., Grap, B., & Wehrle, K. (2011). Efficient online estimation of bursty wireless links. In *Proceedings of 2011 IEEE Symposium on Computers and, Communications* (ISCC'11), pp. 191–198, June–July 2011.
32. Luo, J., Xu, X., & Zhang, Q. (2011). Understanding link feature of wireless sensor networks in outdoor space: A measurement study. In *Proceedings of 2011 IEEE Global Telecommunications Conference* (GLOBECOM'11), pp. 1–5, December 2011.
33. Österlind, F., Mottola, L., Voigt, T., Tsiftes, N., & Dunkels, A. (2012). Strawman: Resolving collisions in bursty low-power wireless networks. In *Proceedings of the 11th International Conference on Information Processing in Sensor Networks* (IPSN'12) pp. 161–172, April 2012.

34. Lee, H., Cerpa, A., & Levis, P. (2007). Improving wireless simulation through noise modeling. In *Proceedings of the 6th ACM/IEEE International Conference on Information Processing in Sensor Networks* (IPSN '07), pp. 21–30, April 2007.
35. Mieghem, P. V. (2006). *Performance Analysis of Communications Networks and Systems*. Cambridge: Cambridge University Press.

Author Biographies

Yantao Li is an Assistant Professor in the College of Computer and Information Science in Southwest University, China. He received his Ph.D. degree from the College of Computer Science at Chongqing University in December 2012. From September 2010 to September 2012, he was a visiting scholar supported by China Scholarship Council working with Professor Gang Zhou at Department of Computer Science in The College of William and Mary, USA. His research area includes wireless communication and networking, sensor networks, and hash function construction.



Daniel Graham is a Ph.D. candidate in Department of Computer Science at The College of William and Mary. His research is with wireless sensor networks and software evolution.

Gang Zhou is an Assistant Professor in the Computer Science Department at the College of William and Mary. He received his Ph.D. degree from the University of Virginia in 2007 under Professor John A. Stankovic. He has published more than 40 wireless communication and sensor networking papers in prestigious conferences and journals. The total citations of his papers are more than 2400 according to Google Scholar, among which the MobiSys'04 paper has been cited about 600 times. Also, he has 8 papers each of which attracts more than 100 citations since 2004. Dr. Zhou served program vice chair, doctor colloquium panelist, or technical program committee positions for 45 academic conferences, including ACM SenSys, IEEE INFOCOM, IEEE RTSS, etc. He is a reviewer for 30 journals and 50 conferences. Dr. Zhou served as NSF, NIH, and GENI proposal review panelists multiple times during 2008 2012. He received an award for his outstanding service to the IEEE Instrumentation and Measurement Society in 2008. He is also a recipient of the Best Paper Award of IEEE ICNP 2010.

Xin Qi received the B.Sc. degrees in Computer Science from Nanjing University, China, in 2007 and the M.E. Degree from LIAMA, a joint lab between Chinese Academy of Science and INRIA, in 2010, respectively. He has been working in declarative networking. He is currently pursuing Ph.D. degree in the Department of Computer Science, The College of William and Mary. His research interests are mainly in wireless networking and pervasive computing.

Shaojiang Deng is a Professor in the College of Computer Science at Chongqing University, China. He received his Ph.D. degree from the College of Computer Science at Chongqing University in 2005. He was with the Institute of Applied Computer Science, Dresden University of Technology, Germany, as a visiting scholar in 2007. His research interest focuses on digital chaotic cryptography, information security and hash function construction.

Di Xiao is a Professor in the College of Computer Science at Chongqing University, China. He received his Ph.D. degree from the College of Computer Science at Chongqing University in 2005. From 2008 to 2009, He was with the College of Computing Science, New Jersey Institute of Technology, USA, as a visiting scholar. His research interest lies in information security, multimedia security, information authentication and hash function construction.