

# Cross-layer Analysis of the End-to-end Delay Distribution in Wireless Sensor Networks

Yunbo Wang    Mehmet C. Vuran    Steve Goddard

**Abstract**—Emerging applications of wireless sensor networks (WSNs) require real-time quality of service (QoS) guarantees to be provided by the network. Due to the non-deterministic impacts of the wireless channel and queuing mechanisms, probabilistic analysis of QoS is essential. One important metric of QoS in WSNs is the probability distribution of the end-to-end delay. Compared to other widely used delay performance metrics such as the mean delay, delay variance, and worst-case delay, the delay distribution can be used to obtain the probability to meet a specific deadline for QoS-based communication in WSNs.

To investigate the end-to-end delay distribution, in this paper, a comprehensive cross-layer analysis framework, which employs a stochastic queueing model in realistic channel environments, is developed. This framework is generic and can be parameterized for a wide variety of MAC protocols and routing protocols. Case studies with the CSMA/CA MAC protocol and an anycast protocol are conducted to illustrate how the developed framework can analytically predict the distribution of the end-to-end delay. Extensive testbed experiments and simulations are performed to validate the accuracy of the framework for both deterministic and random deployments. Moreover, the effects of various network parameters on the distribution of end-to-end delay are investigated through the developed framework. To the best of our knowledge, this is the first work that provides a generic, probabilistic cross-layer analysis of end-to-end delay in WSNs.

## I. INTRODUCTION

Wireless sensor networks (WSNs) have been utilized in many applications as both a connectivity infrastructure and a distributed data generation network due to their ubiquitous and flexible nature [6]. Increasingly, a large number of WSN applications require real-time quality of service (QoS) guarantees [5]. Such QoS requirements usually depend on two common parameters: timing and reliability. The resource constraints of WSNs, however, limit the extent to which these requirements can be guaranteed. Furthermore, the random effects of the wireless channel prohibits the development of deterministic QoS guarantees in these multi-hop networks. Consequently, a probabilistic analysis of QoS metrics is essential to address both timing and reliability requirements. In this work, we focus on the probability distribution of the end-to-end delay in WSNs. Characterization of the end-to-end delay distribution is fundamental for real-time communication applications with probabilistic QoS guarantees. Indeed, the cumulative distribution function (cdf) of the delay for a given deadline can be used as a probabilistic metric for reliability and timeliness.

Characterizing delay in distributed systems has been investigated in different contexts. Recent work has analyzed

the latency performance of WSNs in terms of its first order statistics, i.e., the mean and the variance [3, 8, 15]. However, complex and cross-layer interactions in multi-hop WSNs require a complete stochastic characterization of the delay. Several efforts have been made to provide probabilistic *bounds* on delay. As an example, the concept of Network Calculus [10] has been extended to derive probabilistic bounds for delay through worst case analysis [9, 12]. However, because of the randomness in wireless communication and the low power nature of the communication links in WSNs, worst case analysis cannot capture the stochastic behavior of end-to-end delay. Moreover, work on real-time queueing theory [19, 38] provides stochastic models for unreliable networks. However, these models consider heavy traffic rate, which is not applicable for WSNs. Recently, probabilistic analysis of delay has been performed for broadcast networks [7, 25, 28, 29, 31] considering several medium access control (MAC) protocols. While the channel contention has been adequately modeled in these studies, additional delay due to multi-hop communication, queuing delay, and wireless channel errors have not been captured. Capturing these cross-layer effects is imperative to completely characterize the delay distribution in WSNs.

Our goal is to provide a comprehensive analytical model for distribution of end-to-end delay in WSNs. Accordingly, the contributions of this paper are as follows: First, a comprehensive and accurate cross-layer analysis framework is developed to *characterize the end-to-end delay distribution* in WSNs for both deterministic and random deployments of nodes. Second, the *effects of heterogeneity in WSNs on latency is captured* in terms of channel quality, transmit power, queue length, and communication protocols. Third, the developed framework highlights the *relationships between network parameters and the delay distribution* in multi-hop WSNs. Using this framework, real-time scheduling, deployment, admission control, and communication solutions can be developed to provide probabilistic QoS guarantees. To the best of our knowledge, this is the first work that provides a probabilistic cross-layer analysis of end-to-end delay in WSNs.<sup>1</sup>

The remainder of this paper is organized as follows: Related work in this area is summarized in Section II. In Section III, the end-to-end delay distribution problem is formally defined, and an overview of the proposed Markovian model is provided. The detailed derivation of the single-hop delay distribution is described in Section IV, followed by the derivation of the end-to-end delay distribution in Section V. Then, case studies for the CSMA/CA MAC protocol and the anycast protocol are provided in Section VI and Section VII, respectively. Experimental results are provided in Section VIII to validate

The authors are with Department of Computer Science and Engineering, University of Nebraska-Lincoln, Email: {yunbow, mcvuran, goddard}@cse.unl.edu

Supported, in part, by grants from the National Science Foundation (0707975) and the Air Force Office of Scientific Research (FA9550-06-1-0375).

<sup>1</sup>A preliminary version of this work appeared in [34].

the developed model. Finally, Section IX concludes the paper.

## II. RELATED WORK

The problem of probabilistic QoS guarantees has attracted a large amount of research in recent years. The concept of Network Calculus [10] has been extended to support probabilistic delay *bounds* in [9, 12, 18, 30]. Network calculus and its probabilistic extensions are based on a min-plus algebra to provide traffic curves and service curves, which are deterministic (or statistical) bounds of traffic rate and service time, respectively. In these studies, the *worst case* performance bounds are analyzed. However, determining worst case bounds has limited applicability in WSNs for three reasons: First, because of the randomness in wireless communication and the low power nature of the communication links, worst case bounds do not exist in most practical scenarios. Second, the large variance in the end-to-end delay in WSNs results in loose bounds that cannot accurately characterize the delay distribution. Finally, most applications tolerate packet loss for a lower delay of higher priority packets since the efficiency of the system is improved. These motivate the need for *probabilistic delay analysis* rather than worst case bounds.

Moreover, work on real-time queueing theory [19, 38] combines real-time theory and queueing theory to provide stochastic models for unreliable networks. However, these models consider heavy traffic rate (usually saturation mode), which is not applicable for WSNs. Our approach in this paper is similar to real-time queueing theory [19] in that we use a stochastic queuing model for the analysis. In contrast, we do not focus on the scheduling problem in real-time systems, which has been discussed intensively in the literature [19, 21, 38]. Rather, we aim to provide an analytical tool to help develop communication solutions and real-time systems.

Recently, the delay distribution of MAC protocols has been analyzed in several studies for wireless networks and WSNs, in particular. The access delay of several MAC protocols has been investigated including IEEE 802.11b DCF protocol [2] in [7, 29, 31], IEEE 802.15.4 protocol in [27, 28], and TDMA protocols in [25]. However, in these studies, a broadcast network is considered, where each node can hear the transmission of each other. Moreover, in [7, 29, 31], saturated traffic is considered. Consequently, the multi-hop communication effects due to hidden node problems and the low traffic rate of WSNs cannot be captured.

The distribution of link layer retransmissions are modeled in [16]. While the distribution of the number of retransmissions is obtained, the transmission time is regarded as the same for each attempt. Hence, the resulting delay distribution model does not consider the uncertainty due to random backoffs of CSMA/CA protocols. In [37], the end-to-end delay distribution in a linear network is derived for homogeneous networks. However, this model assumes infinite queue length at each node, which may not be practical considering the resource constraints of sensor nodes. A probabilistic end-to-end delay and network lifetime analysis is given for WSNs performing data aggregation in [13], but with the assumption that packet transmission time is exponentially distributed. This assumption

is inaccurate for most of the MAC protocols commonly in use. Finally, in [11, 14, 26], empirical measurements are used to provide probabilistic estimations for end-to-end delay. These solutions exploit on-the-fly measurements but do not provide analytical results. It can be observed that completely and accurately characterizing end-to-end delay in WSNs is still an open problem.

A preliminary version of this work appeared in [34], which constitutes the first step in providing a comprehensive analytical model for distribution of the end-to-end delay in WSNs. In this paper, we extend our previous work in the following aspects. The analytical framework proposed in [34] characterizes the end-to-end delay distribution for a variety of MAC protocols including the TinyOS CSMA/CA MAC protocol. The model is extended to capture a wide variety of MAC protocols with duty cycle operation and a case study with an anycast protocol is included. Moreover, considering the various deployment methodologies in WSN applications, both deterministic and random network topologies are analyzed. Comprehensive testbed experiments and simulations are provided to validate the extended model. In the following, we present the extended analysis framework in more detail.

## III. PROBLEM DEFINITION AND SYSTEM MODEL

In our analysis, we consider a network composed of sensor nodes that are distributed in a 2-D field. Sensor nodes report their readings to a sink through a multi-hop route in the network. Two different types of network deployments are investigated.

- *Deterministic deployment*: Sensor nodes are located at deterministic locations. Each node is labeled by an index  $i$ , and is characterized by its input traffic rate,  $\lambda_i$ , queue length,  $M_i$ , and the maximum number of retransmission attempts,  $\chi_i$ .
- *Random deployment*: The locations for individual sensor nodes are located randomly according to a Poisson point process. In this case, nodes are identified according to their locations rather than deterministic indices. The input traffic rate, queue length, and the maximum number of retransmission attempts for a node located at  $\mathbf{x} = (x, y)$  is denoted as  $\lambda_{\mathbf{x}}$ ,  $M_{\mathbf{x}}$ , and  $\chi_{\mathbf{x}}$ , respectively.

Although any channel model can be used in our framework, in this paper, a log-normal fading channel model is considered for its accuracy [39]. Accordingly, for a given network with a certain MAC protocol and node parameters described above, we are interested in the following two problems:

1) What is the probability distribution function (pdf) of single-hop delay,  $f_{sh(i,j)}(t)$ , between two nodes  $i$  and  $j$  for deterministic deployment, and  $f_{sh(\mathbf{x}_1, \mathbf{x}_2)}(t)$ , between two nodes located at  $\mathbf{x}_1$  and  $\mathbf{x}_2$  for random deployment, for a new arriving packet?

2) Given the single-hop delay pdf, what is the end-to-end delay pdf,  $f_{e(i,s)}(t)$  between a node  $i$  and a sink  $s$  for deterministic deployment, and  $f_{e(\mathbf{x}, \mathbf{x}_s)}(t)$  between a node located at  $\mathbf{x}$  and a sink located at  $\mathbf{x}_s$  for random deployment?

We consider a heterogeneous network for this analysis, where the heterogeneity is defined in terms of channel conditions, the packet error rate,  $PER$ , traffic rate,  $\lambda$ , queue

length,  $M$ , maximum number of retransmission attempts,  $\chi$ , and transmission power,  $P^{tx}$ , with appropriate subscripts indicating the different values for different nodes. In the following, we provide an overview of our solutions for the two problems above and the detailed descriptions are deferred to Sections IV-V.

### A. Single-hop Delay Distribution

Each node is modeled according to a queuing model, which is characterized by its interarrival distribution and service process. More specifically, we model the traffic interarrival according to a geometric distribution as will be explained next. Furthermore, a Discrete Time Markov Process (DTMP) is used to model the service behavior. Therefore, the service time is Phase-Type (PH) distributed [24]. Considering a single processor at each node and a queue capacity of  $M$ , the resulting model is a discrete time Geom/PH/1/M queueing model.

1) *Inter-arrival time*: The geometric inter-arrival time is motivated by the following: In a typical multi-hop WSN, the input traffic at each node consists of two parts: *locally generated packets* and *relay packets*. Locally generated packets consist of the local information sampled by the sensors, whereas relay packets are received from the neighbors of the node. We are interested in finding the inter-arrival time of these packets at each node for our analysis.

The inter-arrival time of the locally generated packets depends on the application requirements, with which the sensor data are generated. For *monitoring applications*, where nodes repeatedly poll their sensors, the generated data is periodic. Accordingly, the locally generated traffic can be modeled using a constant bit rate (CBR) model. For *event-based applications*, nodes send data only if a certain physical event of interest occurs, e.g., the temperature exceeds a given threshold. In this case, the generated data are often sporadic. Considering such physical events do not occur very frequently, the probability that the event occurs at any time is governed by a Poisson process, and the inter-arrival time is exponentially distributed. Since we employ a discrete time model, the Poisson process is equivalent to a Bernoulli process and the exponential distribution of inter-arrival time is equivalent to a geometric distribution [23]. Note that in some applications, the traffic generated for the physical event can be bursty. For tractability, the bursty traffic pattern is not considered in this paper, and is left for future work.

While the locally generated traffic mainly depends on the physical phenomena of interest and the application type, the relay traffic depends on the network parameters. Although characterization of the relay traffic is out of the scope of this paper, we approximate this distribution based on empirical measurements. Testbed experiments have been performed to estimate the distribution of the inter-arrival time of packets in a 10-hop chain network for both types of applications, i.e., monitoring and event-based for low and high traffic rates. In each experiment, each node uses the TinyOS CSMA/CA MAC protocol and generates packets according to either a CBR model (monitoring) or a Poisson process (event-based).

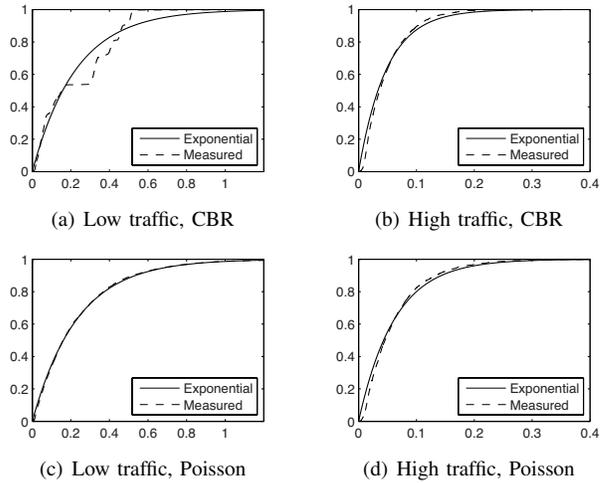


Fig. 1. The distribution of inter-arrival time for different types of traffic for a 10-hop chain. Low traffic: 0.4 packets/s and high traffic: 4 packets/s.

Each node transmits its generated packets and the received packets from its neighbors to the next node toward the end of the chain. The distribution of the inter-arrival time of the packets is recorded at the end of the chain. The empirical *cdf* of the inter-arrival time is shown in Fig. 1 along with an exponential distribution model for four cases<sup>2</sup>. The results reveal that except for the low periodic traffic case shown in Fig. 1(a), exponential distribution closely models the inter-arrival rate. Accordingly, in our discrete-time model, we consider that the inter-arrival time follows a geometric distribution, and define the traffic rate  $\lambda$  at a node to be the probability that a new locally generated packet or relay packet arrives during a time unit  $T^u$ .

2) *Service Time*: The service time of each node is Phase-Type (PH) distributed since the system is modeled according to a discrete-time Markov process (DTMP) with time unit,  $T^u$ . Since a Bernoulli arriving process is assumed for packets and the DTMP is used to describe the behavior of packet communication service, the system is essentially governed by a quasi-birth-death (QBD) process [24] and is modeled by a Geom/PH/1/M queue.

The communication system at each node is modeled as a discrete-time recurrent Markov chain,  $\{X_n\}$ . As shown in Fig. 2(a), this DTMC has a layered structure. Each layer  $i$  contains the part of the chain where there are  $i$  packets in the queue. The communication behaviors of each node are represented by transitions among states in  $\{X_n\}$ . Then, a second DTMC,  $\{Y_n\}$ , which is the absorbing variant of  $\{X_n\}$ , is used to obtain the single-hop delay distribution. The detailed explanation of these DTMCs is provided in Section IV.

### B. End-to-end Delay Distribution

With each hop modeled as a Geom/PH/1/M queue, the entire network is considered as a queueing network. Nodes are interrelated according to the traffic constraints. More specifically, the successfully transmitted traffic rate from one

<sup>2</sup>The exponential distributions shown in Fig. 1 are chosen such that their means are equal to the measured mean inter-arrival times.

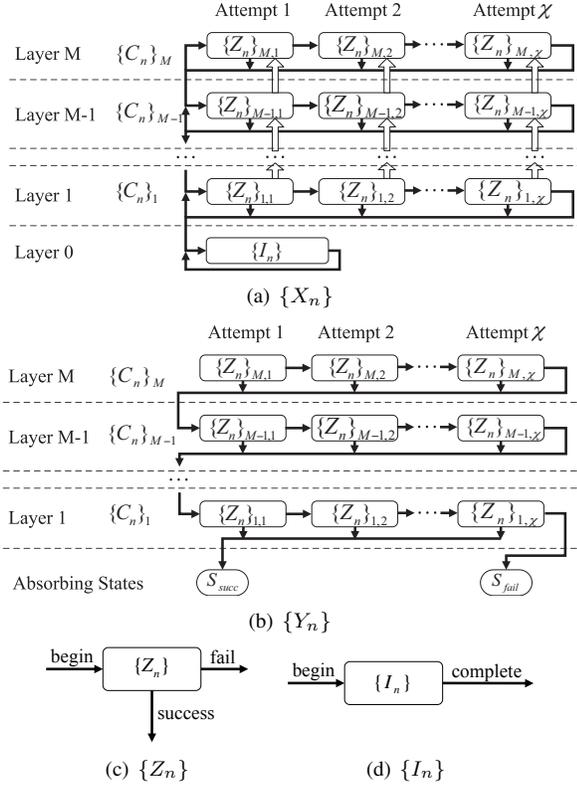


Fig. 2. The structures of Markov chains are shown in (a) for  $\{X_n\}$  and (b) for  $\{Y_n\}$ . The common structure of blocks  $\{Z_n\}$  and  $\{I_n\}$  are shown in (c) and (d).

node should be equal to the sum of the incoming relay traffic rate at each of the next-hop neighbors of the node.

The topology of the queueing network depends on the routing protocol used. In this paper, we focus on the class of routing protocols with which each node maintains a probabilistic routing table for its neighbors, e.g., Geographic routing protocols [4]. Nodes relay their packets to each of their neighbors according to a probability in their routing tables. By first calculating the relaying traffic and the single hop delay distribution for each pair of nodes, the end-to-end delay is obtained using an iterative procedure as will be explained in Section V.

#### IV. SINGLE-HOP DELAY DISTRIBUTION

The communication system at each node  $i$  is modeled by a DTMC  $\{X_n\}^i$  and its absorbing variant  $\{Y_n\}^i$ . For clarity, the index  $i$  is omitted in single-node contexts when there is no ambiguity. First,  $\{X_n\}$  is constructed to capture the equilibrium behavior of the communication. Then,  $\{Y_n\}$  is used to analyze the transient communication behavior after a specific packet arrives. The single-hop delay of the packet communication is then represented as the absorption time of  $\{Y_n\}$ . In the following, the construction of  $\{X_n\}$  and  $\{Y_n\}$  are described in detail and the single-hop delay distribution is derived according to Theorem 1 at the end of this section.

##### A. Constructing Markov Chain $\{X_n\}$

The DTMC,  $\{X_n\}$ , as shown in Fig. 2(a), is composed of  $M + 1$  layers, where each layer  $m$  ( $0 \leq m \leq M$ ) represents

the state where there are  $m$  packets in the queue and  $M$  is the queue capacity. These layers are of two different types, the *idle layer*,  $\{I_n\}$ , and the *communication layers*,  $\{C_n\}_m$ , each of which consists of one or more states. The states and the transitions among the states in each layer is determined by the protocols used by each node, and represent the operations conducted by the nodes according to the protocols. Case studies are presented in Section VI and VII for the TinyOS CSMA/CA protocol and an anycast protocol. The idle layer,  $\{I_n\}$ , ( $m = 0$ ) represents the *idle process*, during which the node does not have any packet to send, and waits for new packets. The communication layers,  $\{C_n\}_m$  ( $m > 0$ ), represent the *communication process* in which packets are transmitted. One or several transmission attempts are conducted, until either the packet is successfully transmitted, or the maximum number of transmission attempts,  $\chi$ , is exceeded. Accordingly, a layer  $m$  in  $\{X_n\}$  is denoted as  $\{C_n\}_m$ , and is composed of  $\chi$  blocks. The  $b$ -th block in layer  $m$  is denoted as  $\{Z_n\}_{m,b}$ <sup>3</sup>. As shown in Fig. 2(c), each block models a single transmission attempt. The structure of  $\{Z_n\}$  depends on the MAC protocol used. Packets are dropped if they arrive at a full queue or if all  $\chi$  transmission attempts fail. Consequently, the  $v$ -th state in layer  $m$  and transmission attempt  $b$  is denoted as  $S_{m,b,v}$ .

The traffic arriving at each node contains locally generated traffic and relay traffic. While locally generated traffic can arrive at any time, the relay traffic can only arrive when the node is listening. Therefore, the total traffic rate depends on the state of the process. The locally generated traffic rate and the relay traffic rate for a node are denoted as  $\lambda^l$  and  $\lambda^r$ , respectively. Therefore, in the states where the node is listening, the total traffic rate is  $\lambda^l + \lambda^r$ , and it is  $\lambda^l$  otherwise.

According to the MAC protocol employed,  $\{I_n\}$  and  $\{C_n\}$  are respectively parameterized by the following notations:

- $P_I$  and  $P_C$ : the transition probability matrix among the states in  $\{I_n\}$  and  $\{C_n\}$ .
- $\alpha_I$  and  $\alpha_C$ : the initial probability vector for  $\{I_n\}$  and  $\{C_n\}$ .
- $t_I^s$  and  $t_C^s$ : the probability vector from each state in  $\{I_n\}$  and  $\{C_n\}$  to complete the idle process and the transmission process successfully.
- $t_C^f$ : the probability vector from each state in  $\{C_n\}$  to complete the transmission process unsuccessfully.
- $\lambda_I$  and  $\lambda_C$ : the packet arrival probability vector for each state in  $\{I_n\}$  and  $\{C_n\}$ . Each element in the vector is the probability of a new packet arrival in a time unit when the process is in the corresponding state.

Each communication layer  $\{C_n\}$  consists of Markov chain blocks for each transmission attempt,  $\{Z_n\}$ , which is further characterized by the transition probability matrix  $P_Z$ , the initial probability vector  $\alpha_Z$ , the success probability vector  $t_Z^s$ , the failure probability vector  $t_Z^f$ , and packet arrival probability vector  $\lambda_Z$ .

Accordingly, the transition probability matrix among the states in a single layer  $\{C_n\}$  in  $\{X_n\}$  can be organized as

<sup>3</sup>In the following, we drop the indices  $m$  and  $b$ , where appropriate, to simplify the notation.

rows and columns of blocks:

$$\mathbf{P}_C = \begin{bmatrix} \mathbf{P}_Z & \mathbf{t}_Z^f \boldsymbol{\alpha}_Z & & \mathbf{0} \\ & \ddots & \ddots & \\ & & \mathbf{P}_Z & \mathbf{t}_Z^f \boldsymbol{\alpha}_Z \\ \mathbf{0} & & & \mathbf{P}_Z \end{bmatrix}, \quad (1)$$

where the number of  $\mathbf{P}_Z$  blocks in  $\mathbf{P}_C$  is equal to  $\chi$ , i.e, the maximum number of attempts for each packet transmission. Similarly, the initial probability vector,  $\boldsymbol{\alpha}_C$ , and the probability vectors,  $\mathbf{t}_C^s$  and  $\mathbf{t}_C^f$  to complete a layer in success and failure are respectively organized as

$$\boldsymbol{\alpha}_C = [\boldsymbol{\alpha}_Z \ \mathbf{0} \ \cdots \ \mathbf{0}] \quad (2)$$

$$\mathbf{t}_C^s = [\mathbf{t}_Z^s \ \mathbf{t}_Z^s \ \cdots \ \mathbf{t}_Z^s]^T \quad (3)$$

$$\mathbf{t}_C^f = [\mathbf{0} \ \mathbf{0} \ \cdots \ \mathbf{t}_Z^f]^T. \quad (4)$$

Note that since the idle layer does not have multiple attempts like the communication does, there is no similar organized internal pattern in the corresponding matrices and vectors for  $\{I_n\}$ . The states and the transitions related to  $\{I_n\}$  and  $\{Z_n\}$  depend on the MAC protocol employed. For now, we assume that these matrices are known and the case studies to obtain them for two different protocols are provided in Section VI and Section VII. The transition probability matrix,  $\mathbf{Q}_X$ , of the entire Markov chain  $\{X_n\}$  can then be found according to transitions between different states at each layer as explained next.

For layer  $m$ ,  $1 \leq m \leq M - 1$ , the queue is not full. Whenever a packet arrives, the process transits to a higher layer since the queue length increases. The probabilities of such transitions are governed by the probability matrix

$$\mathbf{A}_u = (\mathbf{1}\boldsymbol{\lambda}_C)^T \otimes \mathbf{P}_C, \quad (5)$$

where  $\mathbf{1}$  is a properly dimensioned matrix containing all 1's, and  $\otimes$  is the entry-wise product operator.  $\boldsymbol{\lambda}_C$  and  $\mathbf{P}_C$  are parameterized according to the MAC protocol. Note that element  $(v, v')$  in  $\mathbf{A}_u$  represents the transition probability from the  $v$ th state in previous layer to the  $v'$ th state in the upper layer, and other transition probability matrices in the following are defined the similar way. The transition probability matrix at the same level  $m$ ,  $1 \leq m \leq M - 1$ , is

$$\mathbf{A}_s = (\mathbf{1}\boldsymbol{\lambda}_C)^T \otimes (\mathbf{t}_C \boldsymbol{\alpha}_C) + (\mathbf{1} - \mathbf{1}\boldsymbol{\lambda}_C)^T \otimes \mathbf{P}_C, \quad (6)$$

where  $\mathbf{t}_C = \mathbf{t}_C^s + \mathbf{t}_C^f$  is the probability vector from each layer to complete the current communication process regardless of success or failure. The first term in (6) captures the case where a locally generated packet arrives at the same time unit in which a packet service is completed. The second term in (6) is for the case where neither service completion nor new packet arrival occurs during the time unit.

At layer  $m = M$ , the queue is full. Hence, new arriving packets are directly dropped. Therefore, the transition probability matrix in this layer is  $\mathbf{A}_u + \mathbf{A}_s$ .

When there is no packet arrival and the current packet service is completed, the Markov chain transits to one layer

below. The transition probability matrix from level  $m + 1$  to level  $m$ ,  $1 \leq m \leq M - 1$  is

$$\mathbf{A}_d = (\mathbf{1} - \mathbf{1}\boldsymbol{\lambda}_C)^T \otimes (\mathbf{t}_C \boldsymbol{\alpha}_C). \quad (7)$$

The transition probabilities are similar when the idle layer is involved as shown below:

$$\mathbf{A}_{u0} = \boldsymbol{\lambda}_I^T \boldsymbol{\alpha}_C, \quad (8)$$

$$\mathbf{A}_{d0} = (\mathbf{1} - \mathbf{1}\boldsymbol{\lambda}_C)^T \otimes \mathbf{t}_C \boldsymbol{\alpha}_I, \quad (9)$$

$$\mathbf{A}_{s0} = (\mathbf{1} - \mathbf{1}\boldsymbol{\lambda}_I)^T \otimes (\mathbf{P}_I + \mathbf{t}_I^s \boldsymbol{\alpha}_I). \quad (10)$$

When a new packet arrives while there is no packet in the system, the chain transits from the idle layer to layer 1 according to  $\mathbf{A}_{u0}$  in (8). When the service is completed for the only packet in the system and no new packet arrives, the chain transits from layer 1 to the idle layer according to  $\mathbf{A}_{d0}$  in (9). Finally, the transition probabilities with which the node stays in the idle layer are given in  $\mathbf{A}_{s0}$  in (10).

Using (5)-(10), the transition probability matrix  $\mathbf{Q}_X$  for the entire recurrent Markov chain  $\{X_n\}$ , can be constructed as follows:

$$\mathbf{Q}_X = \begin{matrix} & \text{layer} & 0 & 1 & 2 & \cdots & M \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \cdots \\ M \end{matrix} & & \begin{pmatrix} \mathbf{A}_{s0} & \mathbf{A}_{u0} & & & & & \mathbf{0} \\ \mathbf{A}_{d0} & \mathbf{A}_s & \mathbf{A}_u & & & & \\ & & \mathbf{A}_d & \ddots & \ddots & & \\ \cdots & & & \ddots & \mathbf{A}_s & \mathbf{A}_u & \\ \mathbf{0} & & & & \mathbf{A}_d & \mathbf{A}_s + \mathbf{A}_u \end{pmatrix} & \end{matrix}, \quad (11)$$

where each non-zero block corresponds to the transition probability among all layers. The duration of the time unit  $T^u$  is chosen to be small enough such that the probability of having two or more transitions in a single time unit is negligible. Therefore, it is only possible for  $\{X_n\}$  to have intra-layer transitions and inter-layer transitions to adjacent layers. Also note that the first row and column of blocks in  $\mathbf{Q}_X$  corresponds to the transition probabilities from and to the idle layer. Then, the equilibrium state probability vector,  $\boldsymbol{\pi}$ , for  $\{X_n\}$  is calculated by solving  $\boldsymbol{\pi}\mathbf{Q}_X = \boldsymbol{\pi}$  and  $\sum_i \pi_i = 1$ . The detailed solution to this equation system is documented in [34].

## B. Absorbing Time for $\{Y_n\}$

To obtain the distribution of single-hop delay for a packet, consider a particular packet that enters the system at time  $t = t_0$ . The single-hop delay of the packet is the time spent until it is transmitted or dropped. To derive the delay distribution, we use another DTMC,  $\{Y_n\}$ , as an absorbing variant of  $\{X_n\}$ . As shown in Fig. 2(b), in  $\{Y_n\}$ , the idle layer of  $\{X_n\}$  is replaced by two absorbing states  $S_{succ}$  and  $S_{fail}$ , corresponding to the two cases where the packet is successfully transmitted and dropped, respectively. In addition, all new packet arrivals are ignored since they do not interfere with the service time of the packet concerned. Thus, the state transitions occur only inside a layer or from layer  $m + 1$  to  $m$ . The steps to obtain  $\{Y_n\}$  from  $\{X_n\}$  is explained in the following.

Before the packet arrives, the system is in one of the states according to the equilibrium state probability vector,  $\boldsymbol{\pi}$ . After

the new packet arrives, if the queue is full, the packet is immediately dropped. The probability of queue full is

$$p_{qf} = \pi_M \mathbf{A}_u \mathbf{1}, \quad (12)$$

where  $\pi_M$  is the sub-vector in  $\pi$  corresponding to the  $M$ -th layer. Otherwise, the packet is inserted into the queue. The probability vector that the node is in a specific state after the new packet arrives is  $\pi' = \pi \mathbf{Q}_Y^{up}$ , where  $\mathbf{Q}_Y^{up}$  is the transition probability matrix of  $\{Y_n\}$  conditioned on the fact that the new packet arrives.  $\mathbf{Q}_Y^{up}$  is derived from  $\mathbf{Q}_X$  in (11) by replacing  $\lambda_I$  and  $\lambda_C$  with vectors of all 1's in (5)-(10) and replacing  $\mathbf{A}_s + \mathbf{A}_u$  with  $\mathbf{A}_s$ . Note that  $\mathbf{A}_u$  in the bottom-right block accounts for the transition that will cause a packet to drop because of a full queue. Then,  $\pi'$  is the initial probability vector for  $\{Y_n\}$ .

Accordingly, the transition probability matrix for  $\{Y_n\}$  is

$$\mathbf{Q}_Y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \mathbf{t}_Y^s & \mathbf{t}_Y^f & \mathbf{P}_Y \end{bmatrix}, \quad (13)$$

where the transition probabilities from and to the absorbing states  $S_{succ}$  and  $S_{fail}$  are listed in the first two rows and columns. The transition probability matrix among the transient states, i.e., all states except  $S_{succ}$  and  $S_{fail}$ , is given by

$$\mathbf{P}_Y = \begin{bmatrix} \mathbf{P}_C & & & \mathbf{0} \\ \mathbf{t}_C \alpha_C & \mathbf{P}_C & & \\ & \ddots & \ddots & \\ \mathbf{0} & & \mathbf{t}_C \alpha_C & \mathbf{P}_C \end{bmatrix}. \quad (14)$$

This is obtained from (11) by removing the first row and first column of blocks, and replacing  $\lambda_I$  and  $\lambda_C$  with vectors of all 0's in (5)-(10) for each remaining block. The transition probability vectors from each of the transient states to the absorbing states are

$$\mathbf{t}_Y^s = [\mathbf{t}_C^s \quad \mathbf{0} \quad \mathbf{0} \quad \cdots]^T, \quad \mathbf{t}_Y^f = [\mathbf{t}_C^f \quad \mathbf{0} \quad \mathbf{0} \quad \cdots]^T, \quad (15)$$

respectively, where  $\mathbf{t}_C^s$  and  $\mathbf{t}_C^f$  are given in (3) and (4). Finally, since a transition in  $\{Y_n\}$  takes a time unit  $T^u$ , the following important results are directly obtained:

**Theorem 1.** *The pmf of the number of time units,  $k$ , a packet should wait before being transmitted and dropped are*

$$f_K^s(k) = \alpha_Y \mathbf{P}_Y^{k-1} \mathbf{t}_Y^s, \quad f_K^f(k) = \alpha_Y \mathbf{P}_Y^{k-1} \mathbf{t}_Y^f, \quad (16)$$

respectively, where  $\alpha_Y = (\pi'_1, \pi'_2, \dots, \pi'_M)$ , i.e.,  $\pi'$  without the elements corresponding to the idle layer, and  $\mathbf{P}_Y^{k-1}$  represents the  $(k-1)$ -th power of  $\mathbf{P}_Y$ .

*Proof:* The theorem follows from [23, Ch. 9.5]. ■

The pmf of the number of time units a packet should wait, regardless of being transmitted and dropped, is obtained by adding  $f_K^s(k)$  and  $f_K^f(k)$ . Thus, the following corollary is directly obtained.

**Corollary 1.** *The pmf of single-hop delay, measured by the number of time units of  $T^u$ , is given by*

$$f_K(k) = \alpha_Y \mathbf{P}_Y^{k-1} \mathbf{t}_Y. \quad (17)$$

Using this model, the probability that the packet is eventually delivered in success can also be found, and is given by the following corollary:

**Corollary 2.** *The delivery rate of a new arriving packet is*

$$p_{deli} = \sum_{k=1}^{+\infty} f_K^s(k) = \alpha_Y (\mathbf{I} - \mathbf{P}_Y)^{-1} \mathbf{t}_Y^s. \quad (18)$$

Of interest, the first two moments of the successful single-hop delay, which are widely used as the performance metrics in WSN applications, can also be derived.

**Corollary 3.** *The mean and variance of single-hop delay for a new arriving packet are given by*

$$\mu_K = \alpha_Y (\mathbf{I} - \mathbf{P}_Y)^{-2} \mathbf{t}_Y^s / p_{deli}, \quad (19)$$

$$\sigma_K^2 = \frac{\alpha_Y (2(\mathbf{I} - \mathbf{P}_Y)^{-3} - (\mathbf{I} - \mathbf{P}_Y)^{-2}) \mathbf{t}_Y^s}{p_{deli}} - \mu_K^2. \quad (20)$$

The derivations are straightforward and are not included. Next, we derive the end-to-end delay distribution based on the single-hop delay distribution analysis in this section.

## V. END-TO-END DELAY DISTRIBUTION

The end-to-end delay distribution depends on the topology of the network and the routing protocol used. Two types of node deployments are considered in this paper: deterministic deployment and random deployment. For both deployments, we focus on the steady state behavior of the routing protocol. Accordingly, a node forwards a particular packet to any of its neighbor nodes with a certain probability, which does not change rapidly over time. These protocols comprise the majority of routing protocols in WSNs [6]. It is also assumed that in-network processing, such as data aggregation, is not employed.

### A. Deterministic Deployment

In a network with deterministic deployment, each node has a deterministic location and the forwarding probabilities among nodes is determined with the knowledge of the locations. A typical network setup for common applications is considered, where a single sink is used and the routing protocol produces no closed loops in the routing paths (i.e., packets are never routed by a node more than once). In such a case, the network is viewed as a directed acyclic graph (DAG). Without loss of generality, this graph can be topologically sorted so that a node with a larger index never transmits a packet to a node with smaller index. In a network with  $N$  nodes, the index for the sink  $s$  is  $N$ .

Suppose in each time unit of  $T^u$ , each node  $i$  generates a local traffic of  $\lambda_i^l$  to the sink. Each packet is routed using a relay  $k \in \mathbb{N}_i$  with probability  $p_{i,k}^{fw}$ , where  $\mathbb{N}_i$  is the set of potential relays from  $i$  to the sink. Thus,  $\sum_{k \in \mathbb{N}_i} p_{i,k}^{fw} = 1$ ,  $\forall i$ . We first calculate the average relay traffic  $\bar{\lambda}_i^r$  in each time unit from node  $i$  by solving the following equation system for every node:

$$\bar{\lambda}_i^r = \sum_{j=1}^{i-1} (\bar{\lambda}_j^r + \lambda_j^l) p_{j,i}^{fw} p_{deli,j,i}, \quad \forall i, \quad (21)$$

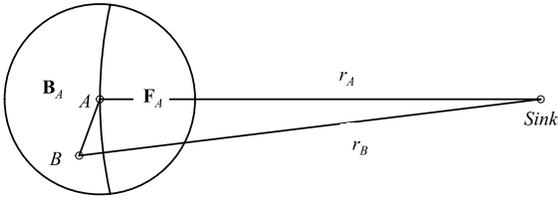


Fig. 3. The feasible region,  $\mathbb{F}_A$ , and the infeasible region,  $\mathbb{B}_A$ , of node  $A$ .

and  $\bar{\lambda}_1^r = 0$ , where  $p_{deli,j,i}$  is the probability that a packet is successfully delivered from node  $j$  to  $i$ , as defined in (18). Then, since each node cannot receive packets in transmission and sleeping states, the relay traffic rate in the states, in which the node is capable to receive packets, is

$$\lambda_i^r = \bar{\lambda}_i^r / \pi_i^{listen}, \quad (22)$$

where  $\pi_i^{listen}$  is the probability that  $i$  is in any state in which the node can receive packets, and is the sum of the probabilities corresponding to all such states in  $\pi_i$ . Whether or not the node can receive packets in a state is determined by the protocol. Accordingly, the input traffic rate vectors  $\lambda_I$  and  $\lambda_C$  of a node  $i$  can be found according to Section IV. Then,  $\lambda_I$  and  $\lambda_C$  are used in (5)-(10) to determine the single-hop delay distribution,  $f_{sh(i,j)}(t)$ , between a pair of nodes  $i$  and  $j$  as discussed in Section IV.

Finally, the end-to-end delay distribution is given as

$$f_{e(i,s)}(t) = \sum_{k=i+1}^{N-1} f_{sh(i,k)}(t) * f_{e(k,s)}(t) p_{i,k}^{fw} + f_{sh(i,N)}(t) p_{i,N}^{fw}, \quad (23)$$

where  $(*)$  is the convolution operator. Our numerical experiments show that it takes less than 2 minutes to obtain the end-to-end delay distribution between two nodes in a network consisting of 16 nodes with TinyOS CSMA/CA MAC protocol. This calculation time is affordable for protocol analysis.

## B. Random Deployment

For the random deployment, the nodes are located in the network according to a Poisson point process with density  $\rho$ . Due to this randomness, the location for each node is stochastic. Therefore, geographic routing protocols [4] are often used due to their scalability and adaptability to the random geographic locations of the nodes. In such protocols, instead of the routing probability  $p_{i,j}^{fw}$  between any pair of nodes  $i$  and  $j$ , the routing probability between any pair of locations  $\mathbf{x}_A$  and  $\mathbf{x}_B$ ,  $p_{\mathbf{x}_A, \mathbf{x}_B}^{fw}$  can be determined.

A common scenario is also considered for the random deployment, where the nodes in the network generate homogeneous amount of local traffic to a sink. Moreover, each node forwards packets to the neighboring nodes within its *feasible region*,  $\mathbb{F}_A$ , i.e., the region in which nodes are closer to the sink, but are still in the transmission range, as shown in Fig. 3. Assume that the sink is located at the center of a circular plane with a radius  $R$ . In this scenario, the end-to-end delay analysis can take advantage of the symmetry of the topology as explained next.

The entire circular plane is discretized into concentric rings indexed by their distance to the sink,  $r$ . Each node senses the physical events, and generates packets with traffic rate  $\lambda^l$ . By symmetry, the relay traffic  $\lambda_r^r$  is the same for all nodes in the same ring  $r$ . In the following analysis, we assume a polar coordinate system with the sink located at the origin.

For a node  $A$  located at  $\mathbf{x}_A = (r_A, \theta_A)$ , the relay traffic arrives from any node  $B$  in the *infeasible region*  $\mathbb{B}_A = \mathbb{C}_A \setminus \mathbb{F}_A$ , where  $\mathbb{C}_A$  is the communication range of  $A$ , as shown in Fig. 3. i.e.,  $\mathbb{B}_A$  is the region in which nodes are farther to the sink but are still in the transmission range. To derive the relay traffic rate for  $A$  and other nodes in ring  $r_A$ , consider the small area  $(r_A : r_A + \Delta r, \theta : \theta + \Delta \theta)$  around node  $A$  located at  $(r_A, \theta)$ . Similar to the deterministic deployment, the relay traffic rate  $\lambda_{r_A}^r$  is given by

$$\lambda_{r_A}^r = \bar{\lambda}_{r_A}^r / \pi_{r_A}^{listen},$$

$$\bar{\lambda}_{r_A}^r = \frac{\int_{\mathbb{B}_A} \rho (\bar{\lambda}_{\mathbf{x}_B}^r + \lambda^l) p_{\mathbf{x}_B, \mathbf{x}_A}^{fw} p_{deli, \mathbf{x}_B, \mathbf{x}_A} d\mathbf{x}_B}{\rho \Delta r \Delta \theta r_A}, \quad (24)$$

where  $\rho$  is the network density of the Poisson node distribution,  $p_{\mathbf{x}_B, \mathbf{x}_A}^{fw}$  and  $p_{deli, \mathbf{x}_B, \mathbf{x}_A}$  are similarly defined as  $p_{m,i}^{fw}$  and  $p_{deli, m, i}$  in (21), except that the nodes are indexed by their locations. Finally,  $p_{\mathbf{x}_B, \mathbf{x}_A}^{fw}$  in (24) is the routing protocol-specific probability that the node at  $\mathbf{x}_B$  transmit packets to a node at  $\mathbf{x}_A$ . A case study for the anycast protocol will be provided in Section VII to show how this probability is obtained.

According to (24), the traffic rate of node  $A$  at each state is determined. Accordingly, the input traffic rate vectors  $\lambda_I$  and  $\lambda_C$  of node  $A$  can be found according to Section IV. Then, the equilibrium state probability for the DTMC  $\{X_n\}$ ,  $\pi_{r_A}$  is obtained. Note that in (24), the traffic rate for nodes in ring  $r_A$  depends on the traffic rate and delivery rate for nodes in their infeasible region. Therefore, the single-hop delay distribution is obtained first for nodes in the outmost ring, and then, for nodes in the inner rings in the decreasing order of the ring radius.

By symmetry, the end-to-end delay distribution to the sink is the same for all nodes with a same distance  $r_A$  to the sink, and is obtained by

$$f_{e(r_A)}(t) = \int_{\mathbb{F}_A} p_{\mathbf{x}_A, \mathbf{x}_B}^{fw} f_{sh(r_A)} * f_{e(r_B)}(t) d\mathbf{x}_B. \quad (25)$$

The end-to-end delay distribution is found in the ascending order of the distance to the sink.

Next, in Section VI, the TinyOS CSMA protocol is used as a case study to show how the DTMCs, specifically, the single transmission attempt block  $\{Z_n\}$ , are constructed, and how the end-to-end delay distribution is obtained, in a deterministic deployed network. Likewise, another case study of the anycast protocol is provided in Section VII to illustrate the end-to-end delay analysis in a randomly deployed network.

## VI. CASE STUDY: TINYOS CSMA/CA PROTOCOL

In this section, we illustrate how single-hop delay distribution can be obtained for a particular MAC protocol in a deterministically deployed network. We use the TinyOS

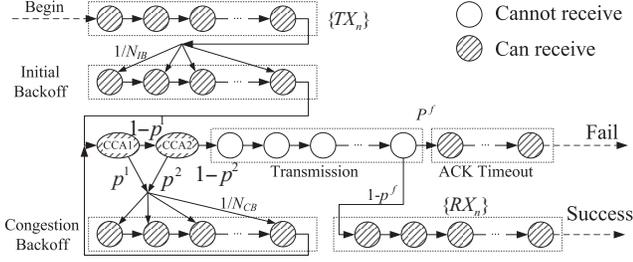


Fig. 4. Markov chain structure for each attempt for TinyOS CSMA protocol.  $N_{IB}$  and  $N_{CB}$  are the number of states representing the initial backoff and congestion backoff, respectively. The subscript  $i$  for node  $i$  is omitted in the figure.

default CSMA/CA protocol [32], which is widely adopted by applications due to the popularity of TinyOS. Similar to the IEEE 802.15.4 protocol [1], a two-slot Clear Channel Assessment (CCA) is conducted before transmitting a packet. As discussed in Section II, there exist several studies that characterize the CSMA/CA protocol in a broadcast network. In this section, we refer to the framework in [28] for our analysis. Since multi-hop traffic and the hidden node problem are not considered in [28], we extend this analysis to the multi-hop case. Note that our aim in this section is not to propose yet another analysis of the CSMA/CA protocol. Instead, we illustrate how the existing models of MAC protocols can be extended through our framework to model the *end-to-end* delay distribution.

#### A. Markov Process Overview

With the TinyOS CSMA/CA protocol, nodes can start transmission at any time when a packet arrives. Therefore, the idle layer  $\{I_n\}$  contains only one state, denoted here as  $S_{idle}$ . The elements in  $\mathbf{P}_I$ ,  $\boldsymbol{\alpha}_I$ ,  $\mathbf{t}_I^s$  and  $\boldsymbol{\lambda}_I$  are easily determined:

$$\mathbf{P}_I = \{0\}, \quad \boldsymbol{\alpha}_I = \{1\}, \quad \mathbf{t}_I^s = \{1\}, \quad \boldsymbol{\lambda}_I = \{\lambda^r + \lambda^l\}. \quad (26)$$

Moreover, the Markov chain,  $\{Z_n\}$ , that models each transmission attempt is depicted in Fig. 4. Before each transmission, the packet in the queue is transferred from the microcontroller to the transceiver. The time needed for such transfer differs for various transceivers but is not negligible. Our experiments with TelosB nodes suggest that the durations of loading time before and after radio transmission are constant and are approximately 1.7 ms and 2.0 ms, respectively. Therefore, the data transfer delay is modeled by two additional state chains with a length corresponding to the transfer duration. These chains are the first and the last part of  $\{Z_n\}$ , denoted as  $\{TX_n\}$  and  $\{RX_n\}$  in Fig. 4.

After the packet is transferred to the transceiver, a random *initial backoff* is conducted to arbitrate with other nodes. Then, the two-slot CCA is performed, which is followed by the packet *transmission* if both CCAs result in a clear channel. If the channel is busy, a random *congestion backoff* is conducted and the channel is sensed again. After the transmission is completed, the node waits for the acknowledgment from the receiver until ACK timeout.

#### B. Constructing the DTMC $\{X_n\}$

For each transmission attempt, the corresponding block of the Markov chain is depicted in Fig. 4, which is characterized by three variables in the chain:  $p_i^1$  and  $p_i^2$  are respectively the probabilities that the node senses the channel busy in the first and second CCA, and  $p_i^f$  is the probability that a transmission attempt fails due to either channel noise or collisions. For the derivations of their values, we first define the *collision area*,  $\mathbb{C}_i$ , of a node  $i$  as the area in which all the neighbors interfere with node  $i$ . For two communicating nodes  $i$  and  $j$ , both nodes reside in the intersection of the collision areas of these nodes, i.e.,  $\{i, j\} \in \mathbb{C}_{i,j}$ <sup>4</sup>, where  $\mathbb{C}_{i,j} = \mathbb{C}_i \cap \mathbb{C}_j$ . Moreover, the collision area of  $i$  that is not in  $\mathbb{C}_{i,j}$  is defined as  $\mathbb{H}_{i,j} = \mathbb{C}_i \setminus \mathbb{C}_{i,j}$ , which is the *hidden node area* of  $i$  with respect to  $j$ . Essentially, nodes that reside in  $\mathbb{H}_{i,j}$  cannot be heard by  $j$ . Similarly, the hidden node area of  $j$  w.r.t.  $i$  is denoted as  $\mathbb{H}_{j,i}$ . The size of these areas  $|\mathbb{C}_{i,j}|$ ,  $|\mathbb{H}_{i,j}|$ , and  $|\mathbb{H}_{j,i}|$  can easily be obtained according to the distance between  $i$  and  $j$  and their respective interference ranges. Accordingly, the number of nodes in these areas are the product of their respective sizes and the network density  $\rho$ .

$$p_{\mathbb{C}_i}^{nc} = 1 - \frac{\sum_{k \in \mathbb{C}_i} p_k^w}{p_{send, \mathbb{C}_i}}. \quad (27)$$

Then, the values of  $p_i^1$ ,  $p_i^2$  and  $p_i^f$  for each node  $i$  are found by solving the following set of equations.

$$p_i^1 = p_{\mathbb{C}_i}^{send} L_{TX} + p_i^{ack} L_{ACK}, \quad (28)$$

$$p_i^2 = \left[ 1 - \frac{2 - p_{\mathbb{C}_i}^c}{2 - p_{\mathbb{C}_i}^c + \frac{1}{1 - \prod_{k \in \mathbb{C}_i} (1 - \phi_k)}} \right] \left( 1 - \prod_{k \in \mathbb{C}_i} (1 - \phi_k) \right) + \frac{1 - p_{\mathbb{C}_i}^c}{2 - p_{\mathbb{C}_i}^c + \frac{1}{\prod_{k \in \mathbb{C}_i} (1 - \phi_k)}}, \quad (29)$$

$$p_{i,j}^f = 1 - \frac{p_i^w (1 - p_{i,j}^{collhid}) (1 - PER_{i,j})}{\phi_i (1 - p_i^1) (1 - p_i^2)}, \quad (30)$$

where  $p_{\mathbb{C}_i}^{send}$  is the probability with which *at least one* node  $k \in \mathbb{C}_i$  begins a transmission,  $L_{TX}$  and  $L_{ACK}$  are respectively the duration of a data packet transmission and an ACK transmission in terms of time units,  $p_i^{ack}$  is the probability that an ACK packet is transmitted by at least one node in  $\mathbb{C}_i$  during a time unit,  $p_{\mathbb{C}_i}^c$  is the probability that given a transmission was going on, a collision is observed on the channel in a given time unit,  $p_i^w$  is the probability that only node  $i$  starts to transmit a packet in a given time unit, and  $p_{i,j}^{collhid}$  is the probability of collision due to hidden terminal transmissions. They are obtained by (25), (27), (28), (30), and (31) in [34]. Moreover,  $\phi_i$  is the probability that node  $i$  is in the first CCA state and is given in  $\boldsymbol{\pi}_i$ , the stationary probability vector for node  $i$ ;  $p_i^1$  and  $p_i^2$  are the probability that the node senses the first and the second CCA busy, respectively. Note that since heterogeneous network traffic is considered,  $\phi_i$  may be different for different nodes. Finally,  $PER_{i,j}$  is the packet error rate dependent on

<sup>4</sup>With a slight abuse of notation, in the following,  $i \in \mathbb{C}$  is used to indicate node  $i$  resides in area  $\mathbb{C}$ .

channel noise, which depends on the transmission distance, transmission power, random multi-path and shadowing effects. In our model, we define the expected packet reception rate for a pair of nodes according to the log-normal fading model in [39].

Also note that  $p_{i,j}^f$  is averaged among all destinations,  $j$ , as the approximation of  $p_i^f$  for each node  $i$ . As suggested in (30), the value of  $p_{i,j}^f$  depends on the channel conditions and the collision probability. Considering a channel-aware routing protocol is employed,  $p_{i,j}^f$  does not vary significantly for different node pairs and such approximation is acceptable. Accordingly, for a given node  $i$ , the failure probability for each transmission attempt,  $p_i^f$ , is the same for all packets in the queue.

The three probability values,  $p_i^1$ ,  $p_i^2$ , and  $p_i^f$  are then used to construct the Markov chain,  $\{Z_n\}$ , as follows. First, the states in  $\{Z_n\}$  are given an index in an arbitrary order. Then, the following matrices and vectors are determined:

- The  $(v, v')$ -th element in  $\mathbf{P}_Z$  is equal to the transition probability from state  $v$  to  $v'$ .
- The element in  $\alpha_Z$  corresponding to the first state in  $\{TX_n\}$  (as shown in Fig. 4) is 1. Other elements are 0's.
- The element in  $t_Z^s$  corresponding to the last state in  $\{RX_n\}$  is 1, and the element in  $t_Z^f$  corresponding to the last state for ACK timeout is 1. Other elements are 0's.
- The elements in  $\lambda_Z$  corresponding to the transmission states are  $\lambda^l$  and other elements are  $\lambda^l + \lambda^r$ .

Each of these values depends on each other as well as  $\phi_i$ , which is the probability that the node  $i$  is in the first CCA state. Note that  $\phi_i$ ,  $p_i^1$  and  $p_i^2$  cannot be determined without the knowledge of  $\pi_i$ , which can only be obtained after constructing the Markov chain as explained in Section IV. Consequently, an iterative procedure is used to find these parameters. First, initial guesses of  $\phi_j$ ,  $p_i^1$  and  $p_i^2$ , which are set to all 0's in our evaluation, are used to construct the Markov chains for each node. Based on the Markov chains,  $\pi_i$  is calculated. Then, values for  $\phi_i$ ,  $p_i^1$  and  $p_i^2$  are updated accordingly to the knowledge of  $\pi_i$ . The calculation of  $\phi_i$ ,  $p_i^1$ ,  $p_i^2$ , and  $\pi_i$  is conducted iteratively, until the difference of the value for any variable between two iterations is negligible.

After  $\{Z_n\}$  is constructed, the entire DTMC  $\{X_n\}$  is obtained according to Section IV. The single-hop delay distribution is then derived by Theorem 1. Finally, the end-to-end delay distribution is found according to (23). The results are described in Section VIII-A.

## VII. CASE STUDY: ANYCAST PROTOCOL

In this section, the approach for computing single-hop and end-to-end delay distributions is illustrated for an anycast protocol. The anycast technique has been widely adopted in WSNs recently [17, 22, 33]. Since there is no dominantly used anycast protocol, in this paper, we model the representative protocol described below. This case study is used to show how the single-hop and the end-to-end delay analysis in Section IV and Section V can be applied to protocols with *duty cycle* operations for a randomly deployed network. Other

anycast protocols, and more generally, other duty cycle-based protocols, can be modeled using similar approaches.

For the random deployment of nodes, the topology model in Section V-B is considered, and node-specific variables are indexed by the ring radius  $r$ . In the following analysis, when there is no ambiguity, the subscript  $r$  in ring-specific variables is omitted.

In the anycast protocol, sensor nodes report their readings to the sink, located at the center of the circular plane, through multi-hop routes in the network. The nodes (excluding the sink) turn off their radio periodically to save energy. We assume that the waking period in a sleep-wake cycle  $T^c$  for each node is  $T^w$ , and the sleeping period is  $T^{sl}$ . When a node  $A$  has a packet to send, it starts to repeatedly transmit RTS beacon packets based on a CSMA/CA manner, i.e., through carrier sense and random backoff mechanisms. When a node  $B$  in the transmission range is awake and hears the packet, it checks for the following criteria: (1) node  $B$  is closer to the sink than  $A$ , and (2) the signal-to-noise ratio (SNR) of the received RTS packet,  $\psi$ , is greater than some predefined threshold  $\psi^{th}$ . If both criteria are met, node  $B$  sends a CTS packet. Node  $A$  then chooses the first node that sent a CTS packet as the next-hop node and transmits the data packet to it. Successful data packet transmissions are acknowledged by the receiver, otherwise the sender retransmits the data packet until successful, or the maximum number of transmission attempts,  $\chi$ , is reached.

To reduce the waiting time for the packets spent in the queue and balance the energy consumption in the network, in the protocol, each node responds to beacon packets only when it does not have packets to send. Considering the sink is awake all the time, if a node closer than a distance threshold  $r^{th}$  to the sink transmits beacons, it is assumed that no node except the sink will respond. Here  $r^{th}$  is chosen such that a high SNR is almost always guaranteed. Moreover, nodes go to sleep when they finish transmitting all packets in the queue. As a result, compared to non-transmitting nodes, the active period is shorter. In cases where transmission energy consumption is significantly higher than listening, this helps balancing energy consumption among nodes.

We first show the DTMC  $\{X_n\}$  for the protocol. Then, the protocol-specific parameters for the generic analysis in Section IV, including the relay traffic rate at each state, and the transition probabilities for  $\{X_n\}$  are derived. The single-hop delay distribution for each pair of nodes is obtained after these parameters are known. Finally, the end-to-end delay distribution from each node to the sink is provided.

### A. Markov Process Overview

The anycast protocol is modeled according to a DTMC  $\{X_n\}$  discussed in Section IV. The structures of the idle layer  $\{I_n\}$  and one of the communication layers  $\{C_n\}$  for this protocol are shown in Fig. 5(a) and Fig. 5(b), respectively. The process enters  $\{I_n\}$  periodically when there is no packet to send.  $\{I_n\}$  consists of a group of sleeping states and listening states. During the listening states, the node listens to the channel. Thus, both locally generated packets and relay packets can

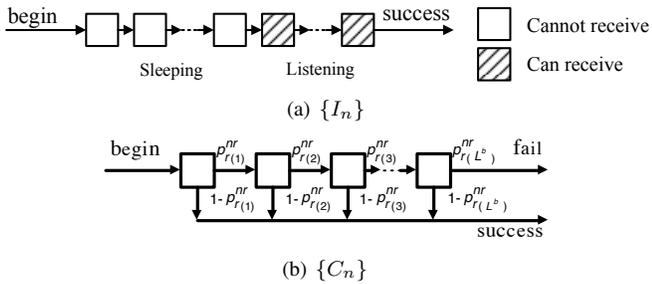


Fig. 5. The Markov chain structure of (a) the communication process,  $\{C_n\}$ , and (b) the idle process,  $\{I_n\}$ , for the anycast protocol.

arrive. During the sleeping states, however, the node turns off its transceiver and therefore only local packets can arrive. The number of states in  $\{I_n\}$  is  $L^c = T^{sl}/T^u + T^w/T^u = T^c/T^u$ , where  $T^u$  is the unit time. A large  $T^u$  can reduce the number of states in the DTMC, thus reducing computation cost for the model, but at the cost of reducing the granularity and accuracy of the result.<sup>5</sup> When a local or relay packet arrives, the node terminates the idle process and begins the first layer of communication process.

In each communication layer, which consists of a  $\{C_n\}$  block, the node keeps transmitting beacon packets. The number of states in  $\{C_n\}$  is  $L^b = T^b/T^u$ , where  $T^b$  is the beacon timeout. If a node receives RTS responses from other nodes, it starts transmitting the data packet to the first responding node. Retransmissions are conducted in case of a transmission failure. Since only neighbor nodes that receive the beacon packets with a high SNR will respond, a high quality wireless channel is guaranteed. Moreover, in most WSN applications, the traffic rate is low, and the chance of packet collision with other nodes is small. Therefore, data packets are transmitted successfully in limited number of (re)transmission attempts, which takes negligible time compared to the sleeping cycle  $T^c$  (usually longer than 10sec). Thus,  $\{C_n\}$  only contains transmission states. When the first RTS packet is received, the transmission terminates in a success. When the beacon transmission times out, the packet is dropped, and the transmission terminates in a failure. In either way, the node enters the lower layer. Note that the beacon timeout  $T^b$  is usually chosen equal to or longer than the cycle  $T^c$ . This is to ensure that each neighbor node can receive the beacon messages within their duty cycle period. The entire beacon transmission process before packet delivery or timeout is regarded as a single transmission attempt. Thus, each communication layer  $\{C_n\}$  contains only one block of  $\{Z_n\}$ .

### B. Constructing the DTMC $\{X_n\}$

Unlike the TinyOS CSMA protocol with which there is only a single state in  $\{I_n\}$ , to capture the duty cycle operation in the anycast protocol, a series of states are needed. The transition probabilities in  $\{I_n\}$  and  $\{C_n\}$  are 1's when not noted. The transition probabilities  $p_{r_A}^{nr(i)}$ , and the traffic rate  $\lambda_I$ ,  $\lambda_C$  are explained in the following.

In the  $i$ th time unit in  $\{C_n\}$ , a node  $A$  in ring  $r_A$  has a probability of  $p_{r_A}^{nr(i)}$  of not receiving any CTS response, and

<sup>5</sup>Recall in Section IV that it is assumed only one packet may arrive in a time unit. This is accurate only when  $T^u$  is chosen small.

enters the next state. If in all  $N_m$  states, node  $A$  receives no CTS response, the transmission fails and the packet is dropped. On the other hand, if in any of the states, a CTS response is received, the node transmits the packet and the transmission succeeds. The probability  $p_{r_A}^{nr(i)}$  is the conditional probability that given the transmissions in the previous  $i - 1$  states failed, the transmissions in the  $i$ th state still fails. For simplicity the hidden terminals are ignored. Hidden terminal effects in high density networks can be easily captured by the model as shown in Section VI. Therefore,

$$p_{r_A}^{nr(1)} = p_{r_A(1\sim 1)}^{nr}$$

$$p_{r_A}^{nr(i)} = p_{r_A(1\sim i)}^{nr} / p_{r_A(1\sim i-1)}^{nr}, \quad 2 \leq i \leq L_r^b \quad (31)$$

where  $p_{r_A(1\sim i)}^{nr}$  is the probability that during all first  $i$  states in  $\{C_n\}$ , beacon transmission fails, since no CTS packet is received in these states. Therefore,

$$p_{r_A(1\sim i)}^{nr} = \prod_{\mathbf{x}_B \in \mathbb{F}_A} \left( 1 - \rho \Delta r_B \Delta \theta_B p_{r_B(i)}^{ol} p_{A,B}^{SNR} \right), \quad (32)$$

where node  $(\mathbf{x})_B = (r_B, \theta_B)$  is any location in the feasible region of  $A$ ;  $\rho$  is the node density;  $p_{r_B(i)}^{ol}$  is the probability that the waking period of a node  $B$ , which is located  $r_B$  away from the sink, overlaps with the first  $i$  beacon transmission time units of  $A$ , and is given by (28) in [35]. Moreover,  $p_{A,B}^{SNR}$  is the probability that a packet, which is transmitted by node  $A$  and received by a node located at  $\mathbf{x}_B$ , has an SNR higher than some predefined threshold  $\psi^{th}$ . It is obtained by (10) in [39]. Then,  $p_{r_A(1\sim i)}^{nr}$  in (32) can be determined, and  $p_{r_A}^{nr(i)}$  in (31) is obtained using (32).

Next, the traffic rate at each state,  $\lambda_I$  and  $\lambda_C$ , are discussed. The arriving traffic at  $A$  contains locally generated and relay traffic. In sleeping states, the traffic arrival rate is  $\lambda_{r_A}^l$ . In listening states, the traffic rate is  $\lambda_{r_A}^r + \lambda_{r_A}^l$ . Finally, in the beacon transmission states, since nodes are assumed not to respond to any relay packets, the traffic rate is  $\lambda_{r_A}^l$ . The value of  $\lambda_{r_A}^r$ , is obtained according to (24). The protocol-specific probabilities,  $p_{\mathbf{x}_B, \mathbf{x}_A}^{fw}$  are derived in the following.

The probability that a node  $A$  located at  $\mathbf{x}_B$  forwards a packet to a node  $B$  located at  $\mathbf{x}_A$ , among all possible forward targets,  $p_{\mathbf{x}_B, \mathbf{x}_A}^{fw}$ , is proportional to the probability that  $A$  is available when  $B$  transmits a beacon, and is normalized on the total availabilities for all possible nodes. The availability is defined as

$$p_{\mathbf{x}_B, \mathbf{x}_A}^{avail} = p_{r_A}^{wake} p_{\mathbf{x}_B, \mathbf{x}_A}^{SNR}, \quad (33)$$

where  $p_{r_A}^{wake} = \sum_{j=1}^{L^w} \pi_{r_A}^{W_j}$  is the probability that node  $A$  is awake, and  $\pi_{r_A}^{W_j}$  is the equilibrium probability that  $A$  is in the  $j$ th waking state in  $\{X_n\}$ . Then,  $p_{\mathbf{x}_B, \mathbf{x}_A}^{fw}$  in (24) is given by

$$p_{\mathbf{x}_B, \mathbf{x}_A}^{fw} = \frac{p_{\mathbf{x}_B, \mathbf{x}_A}^{avail}}{\int_{\mathbb{F}_B} p_{\mathbf{x}_B, \mathbf{x}_C}^{avail} d\mathbf{x}_C}, \quad (34)$$

where  $\mathbf{x}_C = (r_C, \theta_C)$ , is any location in  $\mathbb{F}_B$ .

Accordingly,  $\{I_n\}$  and  $\{C_n\}$  (containing only one block of  $\{Z_n\}$ ) are characterized by:

- The  $(v, v')$ -th element in  $\mathbf{P}_I$  and  $\mathbf{P}_C$  is the transition probability from state  $v$  to  $v'$  shown in Fig. 5.

- The element in  $\alpha_I$  and  $\alpha_C$  is 1 for states pointed by a “begin” arrow. Other elements are 0’s.
- The element in  $t_I^s$ ,  $t_C^s$ , and  $t_C^f$  is set according to the probability attached to the arrows pointing to “success” and “fail”, respectively.
- The elements in  $\lambda_I$  corresponding to the sleeping states are set to  $\lambda^l$ . Other elements in  $\lambda_I$  and  $\lambda_C$  are set to  $\lambda^l + \lambda^r$ .

Consequently, the single-hop delay distribution and end-to-end delay distribution for each ring are respectively obtained according to (17) and (25).

In the following section, we use empirical evaluations to validate the analytical model for both protocols.

### VIII. ANALYTICAL RESULTS AND EMPIRICAL VALIDATIONS

The end-to-end delay distribution model has been evaluated using MATLAB to determine the single-hop and multi-hop delay distributions for the TinyOS CSMA/CA MAC protocol (Section VI) and the anycast protocol (Section VII). The computing environment is a PC with a Xeon 5150 CPU working at 2.66GHz and 2G RAM. Moreover, empirical experiments and TOSSIM based simulations [20] have been conducted on our WSN testbed to validate the results. The simulations are conducted in the same PC environment. For the empirical validations, Crossbow TelosB motes with a data rate of 250 kbps are used. The packet size is  $l_p = 39$  bytes. Each node  $i$  generates local traffic to be sent to sink  $s$  according to a Poisson distribution with rate  $\lambda_{(i,s)}^l$ . Our experiments with the TelosB motes suggest that it requires on the average 1.7 ms to transfer each packet from the MCU to the RF transceiver and 2.0 ms vice versa. The transmission power is set to -15dBm for all the experiments unless otherwise stated.

In the experiments, the single-hop delay and end-to-end delay are measured as follows: When the source node generates a packet, it simultaneously sends an electric pulse to the destination node through a pair of wires. The destination node starts a timer when it receives the pulse, and waits for the packet. When the packet is received by the destination node, the duration after the reception of the pulse is recorded as the packet delay. This eliminates the need for synchronization among all the nodes. Next, we present the evaluation results for TinyOS CSMA/CA protocol and the Anycast protocol in Section VIII-A and Section VIII-B, respectively.

#### A. Experiments for TinyOS CSMA/CA MAC protocol

1) *Single-hop Delay Distribution*: First, the single-hop delay distribution of the TinyOS CSMA/CA protocol is evaluated according to the derivations in Section VI. The time unit is set to  $T^u = 320\mu s$ . The maximum initial backoff and congestion backoff durations are set to 9.77 ms and 2.44 ms, respectively. For the evaluations, a single hop network is considered where the delay distribution is found for a node under the contention from neighbor nodes. Three different network configurations are considered for the evaluations.

In the first configuration, a node continuously transmits locally generated packets to a receiver node with a data rate of

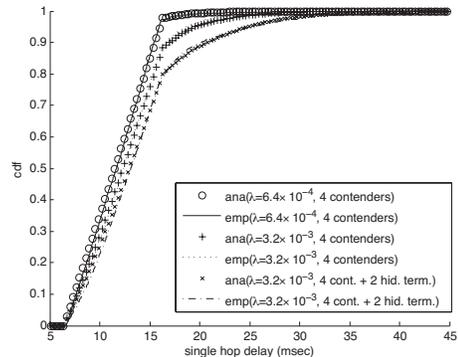


Fig. 6. The *cdf* of the single hop delay of the CSMA/CA protocol. Both empirical (emp) and analytical (ana) results are shown.

2 packets per second. This corresponds to  $\lambda^l = 6.4 \times 10^{-4}$  in the analytical model. Four other nodes are used to transmit packets at the same rate to create background traffic for contention. In the second case, the packet rate for all 5 nodes is increased to 10 packets per second. For the third case, two additional nodes with the same packet generation rate are used, but are placed so that they act as hidden terminals for the transmitting node. The single hop delay for 5,000 packets is recorded for each experiment.

The results of both analytical and empirical validations are shown in Fig. 6 for the *cdf* of the delay. The results show that a higher traffic rate increases hop delay, which is also captured by our model. In addition, the two hidden nodes introduced in the third case cause heavy contention and further increase the hop delay. It can be observed that the analytical model accurately captures the effects of hidden nodes. For all cases, the analytical model has less than 2% of error compared to the empirical evaluations.

In our computing environment with a Xeon 5150 CPU working at 2.66GHz and 2G RAM, the calculations run for less than 10 seconds for a typical hop with 6 neighbors, with  $\chi = 3$  and  $M = 5$  for all nodes, while empirical experiments take more than 5 minutes to obtain enough data samples for meaningful delay distribution estimation.

2) *End-to-end Delay Distribution*: To validate the model for multi-hop networks and illustrate the effects of network parameters in WSNs, two sets of experiments have been performed. First, a network consisting of 25 TelosB nodes are used. The nodes are placed in a  $5 \times 5$  grid, as illustrated in Fig. 7(a). Nodes shown as light-colored boxes only relay packets while the 8 dark-colored nodes also generate packets according to a Poisson process. The transmit power for every node is -25 dBm. The generated traffic rate for the 8 nodes,  $\lambda^l$ , the queue length,  $M$ , and the maximum number of transmission attempts,  $\chi$  are varied to reveal the relationships between each of the parameters and the end-to-end delay distribution. End-to-end delay is measured for approximately 3,000 packets for each configuration.

The results are shown in Fig. 7(d). As can be observed, the *cdf* of the analytical model match well with the empirical results with an error less than 5%. The slight difference in these results is partially due to the inaccurate collision

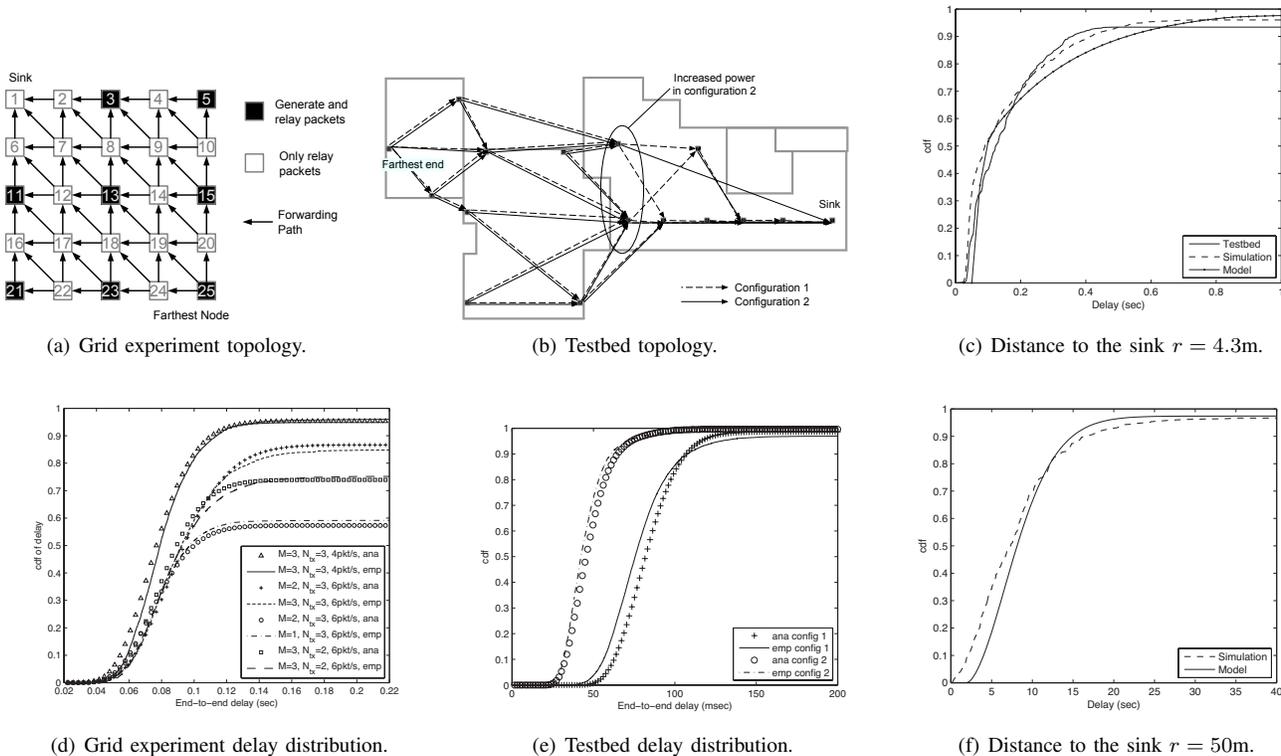


Fig. 7. The topology and the end-to-end delay distribution for the multi-hop grid experiments (a, d) and the testbed experiments (b, e). The end-to-end delay distribution of the Anycast protocol are also shown in (c) and (f).

models, since the collision range in practice is not an arbitrary area for each node and a transitional area exists around the boundary [39]. The results suggest that heavier traffic leads to a longer end-to-end delay and a lower reliability as can be observed from the asymptotic value of the *cdf*. In addition, by reducing the queue length,  $M$ , and the maximum number of transmission attempts,  $\chi$ , the reliability decreases. However, when a low delivery rate (e.g., less than 50%) is sufficient, a lower  $M$  or  $\chi$  does not largely affect the delay performance. More specifically, the average waiting time can be reduced by decreasing the queue capacity and the chance of collisions is decreased since less retransmissions are allowed. This fact is useful when designing applications with nodes having limited memory space.

Experiments are also performed in a realistic indoor environment. A multi-hop network of 16 TelosB nodes is located in three rooms as shown in Fig. 7(b). Two different network configurations are used to illustrate the effects of topology changes. In both configurations, each node generates Poisson traffic of 2 packets per second and the packets are forwarded to the sink as shown in Fig. 7(b). A geographical routing protocol is used to determine the forwarding routes based on the distance between each node and the sink. In the first configuration, every node transmits packets with a power of -15 dBm and the routes are shown in dashed lines. In the second configuration, *two nodes* are selected to transmit packets with an increased power of -7dBm. Therefore, they can directly reach the sink. The routes for the second case are shown in Fig. 7(b) as solid lines. The *cdfs* of the results are shown in Fig. 7(e). Accordingly, increasing transmit power in only

two nodes significantly impacts the end-to-end delay as also captured by the analytical evaluations.

### B. Experiments for Anycast Protocol

We first show that the analytical results of the end-to-end delay distribution are validated by the simulation and the testbed experiments. The anycast protocol described in Section VII is implemented in TinyOS 2.0. Our testbed consists of 25 Crossbow TelosB motes. The nodes are randomly placed in a circular area of radius  $R = 4.5\text{m}$ . Thus, the density is roughly  $\rho = 0.39$ . The data packet size is  $l_p = 39$  bytes, whereas the beacon message and the CTS response message have the same size of  $l_m = 22$  bytes. Each node generates the same amount of local traffic to be sent to the sink according to a Bernoulli process with average rate  $\lambda^l = 0.001$  in each time unit  $T^u = 0.01\text{sec}$ , which equals to 0.1 packet per second. The operating cycle of each node is  $T^c = 1$  sec, during which the wake period is  $T^w = 0.5\text{sec}$ , thus the duty cycle is  $\eta = 0.5$ . Moreover, the beacon transmission timeout is  $T^b = 1\text{sec}$ . The transmission power is set to -15dBm for all the nodes. The threshold radius  $r^{th}$  is set to 2.7m, within which all nodes only transmit packets to the sink. The SNR threshold is set to  $\psi^{th} = 10\text{dB}$ . The simulation is performed on the same topology. Both the simulation and the testbed experiment have been run for 2.5 hours and the end-to-end delay distribution for a node at distance  $r = 4.3\text{m}$  is recorded.

The results are compared with analytical prediction from the model, as shown in Fig. 7(c). It can be observed that the analytical results agree well with both the simulation result and the testbed experiment result, and the error is less than

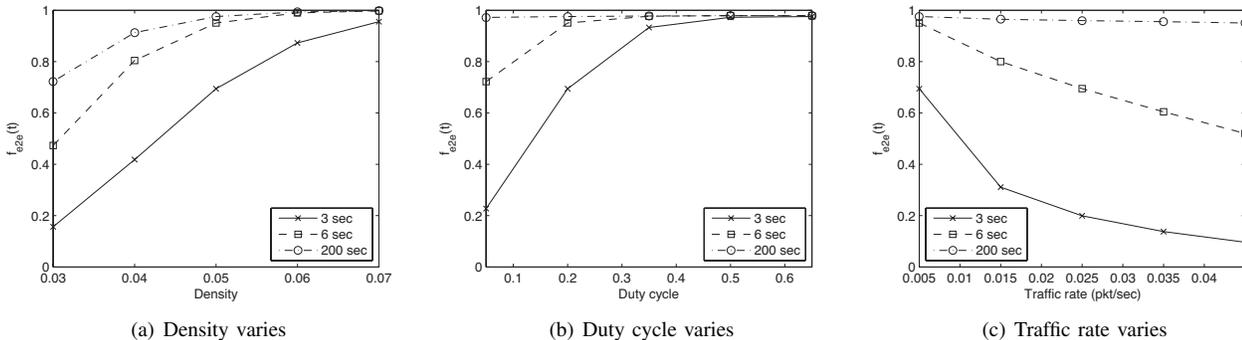


Fig. 8. The relationship between network parameters and the delivery probability within 3 sec, 6 sec, and 200 sec with the Anycast protocol.

10%. Therefore, the simulation is used in the following to validate our model in a larger space and time scales, and for more randomly generated topologies.

In the second set of evaluations, the network radius is set to 50m, the transmission power is increased to -10dBm. Accordingly, the threshold distance is changed to  $r^{th} = 10m$ . Moreover, the network density is  $\rho = 0.1$ . Durations  $T^c$ ,  $T^w$ , and  $T^b$  are 10sec, 5sec and 10sec, respectively, and the traffic rate is 0.01pkt/sec. Other parameters are left unchanged. 20 different topologies are randomly generated according to a Poisson distribution with the same density. Each topology is simulated for 1 hour. The end-to-end delay distribution from all nodes with a distance of 50m to the sink are measured. The result is shown in Fig. 7(f), along with the analytical results. It can be observed that the analytical result is also within 10% of the simulation result.

Next, using the end-to-end delay distribution modeled in (17), we investigate the relationships between the probability of achieving a given end-to-end delay and various network parameters. In each of the following evaluations, the network density  $\rho$ , the duty cycle  $\eta$  for all nodes, and the traffic rate  $\lambda^l$  for all nodes are varied, respectively. The default values for these parameters are 0.02, 0.2, and 0.005pkt/sec. Other parameters are kept unchanged from the previous experiment. The network radius is  $R = 50m$ .

The probability that the end-to-end delay of a node at distance  $r = 50m$  is smaller than 3 sec, 6 sec, and 200 sec are shown in Fig. 8. The results in Fig. 8(a) reveal that when the network density increases, the probability of delivering packets from the edge to the sink also increases. This is because a network with a higher density tends to have more available relaying nodes at any time. Similarly, as shown in Fig. 8(b), when the duty cycle increases, nodes have more waking time to relay packets, thus the probability of delivering packets is increased. Finally, Fig. 8(c) suggests that increasing the traffic rate increases the queuing delay and decreases the probability that nodes are ready to relay packets. Therefore, the probability of delivering packets is smaller as traffic rate increases. It is important to note that given enough time, e.g., 200 sec, the delivery probability does not change much when the duty cycle or the traffic rate varies as shown in Fig. 8(b) and 8(c). However, in Fig. 8(a), the delivery probability after 200 sec changes greatly when the network density changes. This is because lower duty cycle and higher traffic rate prolong

the packet waiting time. Given enough time, there are still enough nodes to relay the packets. On the other hand, a low network density reduces the number of relaying nodes. Therefore, eventually more packets are lost due to timeout in a low density network.

For any network setup in the experiments above, the calculation for the end-to-end delay distribution during any given duration takes less than 2 minute. On the other hand, the TOSSIM-based simulations determine the delay distribution in the same order of actual time. For example, for a simulated duration of 2 hours, the simulation takes roughly 30 mins. Thus, our analytical approach provides insights significantly faster with the same accuracy.

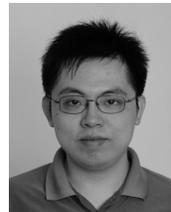
## IX. CONCLUSIONS

Providing QoS guarantees in wireless sensor networks (WSNs) necessitate a probabilistic approach, where the queuing delay and the effects of wireless channel errors are captured. In this paper, an end-to-end analysis of the communication delay is provided. A Markov process based on birth-death problem is used to model the communication process in a multi-hop network. The developed model is validated by extensive testbed experiments through several network configurations and parameters. The results show that the developed framework accurately models the distribution of the end-to-end delay and captures the heterogeneous effects of multi-hop WSNs. The developed framework can be used to guide the development of QoS-based scheduling and communication solutions for WSNs. Based on the framework, models are also developed for event detection delay distributions in WSNs [36].

## REFERENCES

- [1] "IEEE Std 802.15.4: IEEE standard for wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)," Oct 2003.
- [2] "IEEE Std 802.11: IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications," Jun 2007.
- [3] T. Abdelzaher, S. Prabh, and R. Kiran, "On real-time capacity limits of multihop wireless sensor networks," in *Proc. of IEEE RTSS 2004*, pp. 359–370, Dec 2004.
- [4] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, Sept 2005.
- [5] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, Mar 2007.

- [6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks (Elsevier) Journal*, vol. 38, no. 4, pp. 393–422, Mar 2002.
- [7] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, Mar 2000.
- [8] N. Bisnik and A. Abouzeid, "Queueing network models for delay analysis of multihop wireless ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 1, pp. 79–97, Jan 2009.
- [9] A. Burchard, J. Liebeherr, and S. Patek, "A min-plus calculus for end-to-end statistical service guarantees," *IEEE Trans. on Information Theory*, vol. 52, no. 9, pp. 4105–4114, Sep 2006.
- [10] R. Cruz, "A calculus for network delay. i. network elements in isolation," *IEEE Trans. on Information Theory*, vol. 37, no. 1, pp. 114–131, Jan 1991.
- [11] E. Felemban, C.-G. Lee, E. Ekici, R. Boder, and S. Vural, "Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks," in *Proc. of IEEE INFOCOM 2005*, vol. 4, pp. 2646–2657 vol. 4, Mar 2005.
- [12] M. Fidler, "An end-to-end probabilistic network calculus with moment generating functions," in *Proc. of IEEE IWQoS*, pp. 261–270, New Haven, CT, Jun 2006.
- [13] L. Galluccio and S. Palazzo, "End-to-End delay and network lifetime analysis in a wireless sensor network performing data aggregation," in *Proc. of IEEE GLOBECOM 2009*, pp. 146–151, Honolulu, HI, Nov 2009.
- [14] K. Gopalan, T.-C. Chiueh, and Y.-J. Lin, "Probabilistic delay guarantees using delay distribution measurement," in *Proc. of ACM MULTIMEDIA 2004*, pp. 900–907, New York, NY, Oct 2004.
- [15] G. R. Gupta and N. B. Shroff, "Delay analysis for multi-hop wireless networks," in *Proc. of IEEE INFOCOM 2009*, pp. 412–421, Rio de Janeiro, Brazil, Apr 2009.
- [16] T. Issariyakul and E. Hossain, "Analysis of end-to-end performance in a multi-hop wireless network for different hop-level ARQ policies," in *Proc. of IEEE GLOBECOM 2004*, vol. 5, pp. 3022–3026 Vol.5, Nov 2004.
- [17] J. Kim, X. Lin, and N. Shroff, "Optimal anycast technique for delay-sensitive energy-constrained asynchronous wireless sensor networks," in *Proc. of IEEE INFOCOM 2009*, pp. 412–421, Rio de Janeiro, Brazil, Apr 2009.
- [18] A. Koubaa, M. Alves, and E. Tovar, "Modeling and worst-case dimensioning of cluster-tree wireless sensor networks," in *Proc. of IEEE RTSS 2006*, pp. 412–421, Dec 2006.
- [19] J. Lehoczky, "Real-time queueing network theory," in *Proc. of IEEE RTSS 1997*, pp. 58–67, Dec 1997.
- [20] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: accurate and scalable simulation of entire tinys applications," in *Proc. of ACM SenSys 2003*, pp. 126–137, Los Angeles, CA, Nov 2003.
- [21] H. Li, P. Shenoy, and K. Ramamritham, "Scheduling messages with deadlines in multi-hop real-time sensor networks," in *Proc. of IEEE RTAS 2005*, pp. 415–425, Mar 2005.
- [22] S. Liu, K.-W. Fan, and P. Sinha, "CMAC: An energy efficient mac layer protocol using convergent packet forwarding for wireless sensor networks," in *Proc. of SECON 2007*, pp. 11–20, Jun 2007.
- [23] R. Nelson, *Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling*. Springer-Verlag New York, Inc., 1995.
- [24] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: an Algorithmic Approach*. Dover Publications Inc., 1981.
- [25] M. Neuts, J. Guo, M. Zukerman, and H. L. Vu, "The waiting time distribution for a TDMA model with a finite buffer and state-dependent service," *IEEE Trans. on Communications*, vol. 53, no. 9, pp. 1522–1533, Sep 2005.
- [26] R. Oliver and G. Fohler, "Probabilistic estimation of end-to-end path latency in wireless sensor networks," in *Proc. of IEEE MASS 2009*, pp. 423–431, Macau, China, Oct 2009.
- [27] P. Park, P. Di Marco, P. Soldati, C. Fischione, and K. Johansson, "A generalized markov chain model for effective analysis of slotted IEEE 802.15.4," in *Proc. of IEEE MASS 2009*, pp. 130–139, Oct 2009.
- [28] S. Pollin, M. Ergen, S. Ergen, B. Bougard, F. Catthoor, A. Bahai, and P. Varaiya, "Performance analysis of slotted carrier sense IEEE 802.15.4 acknowledged uplink transmissions," in *Proc. of IEEE WCNC 2008*, pp. 1559–1564, Mar 2008.
- [29] T. Sakurai and H. Vu, "MAC access delay of IEEE 802.11 DCF," *IEEE Trans. on Wireless Communications*, vol. 6, no. 5, pp. 1702–1710, May 2007.
- [30] J. Schmitt, F. Zdarsky, and L. Thiele, "A comprehensive worst-case calculus for wireless sensor networks with in-network processing," in *RTSS 2007*, pp. 193–202, Dec 2007.
- [31] O. Tickoo and B. Sikdar, "Modeling queueing and channel access delay in unsaturated IEEE 802.11 random access MAC based wireless networks," *IEEE/ACM Trans. Networks*, vol. 16, no. 4, pp. 878–891, Aug 2008.
- [32] "TinyOS," web, <http://webs.cs.berkeley.edu/tos/>.
- [33] M. C. Vuran and I. F. Akyildiz, "XLP: A cross layer protocol for efficient communication in wireless sensor networks," *IEEE Trans. on Mobile Computing*, vol. 9, no. 11, pp. 1578–1591, Nov 2010.
- [34] Y. Wang, M. C. Vuran, and S. Goddard, "Cross-layer analysis of the end-to-end delay distribution in wireless sensor networks," in *Proc. of IEEE RTSS 2009*, Washington, DC, Dec 2009.
- [35] —, "Stochastic analysis of energy consumption in wireless sensor networks," in *Proc. SECON 2010*, Boston, MA, Jun 2010.
- [36] —, "Analysis of event detection delay in wireless sensor networks," in *Proc. of IEEE INFOCOM 2011*, Shanghai, China, Apr 2011.
- [37] M. Xie and M. Haenggi, "Towards an end-to-end delay analysis of wireless multihop networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 849–861, July 2009.
- [38] S.-N. Yeung and J. Lehoczky, "End-to-end delay analysis for real-time networks," in *Proc. of IEEE RTSS 2001*, pp. 299–309, Dec 2001.
- [39] M. Zúniga and B. Krishnamachari, "An analysis of unreliability and asymmetry in low-power wireless links," *ACM Trans. on Sensor Networks*, vol. 3, no. 2, Jun 2007.



**Yunbo Wang** (M'07) received the B.S. degree and M.S. degree in electric science and technology from University of Science and Technology of China in 2003 and 2006, respectively. He is now a Ph.D. student in computer science at University of Nebraska-Lincoln, under the supervision of Prof. Steve Goddard and Prof. Mehmet C. Vuran. His current research interests include system modeling and performance analysis of wireless sensor networks and embedded systems.



**Mehmet C. Vuran** (M'98) received his B.Sc. degree in Electrical and Electronics Engineering from Bilkent University, Ankara, Turkey, in 2002. He received his M.S. and Ph. D. degrees in Electrical and Computer Engineering from the Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, in 2004 and 2007, respectively, under the guidance of Prof. Ian F. Akyildiz. Currently, he is an Assistant Professor in the Department of Computer Science and Engineering at the University of Nebraska-Lincoln. He is the recipient of an NSF CAREER award in 2010. His current research interests include cross-layer design and correlation-based communication for wireless sensor networks, underground sensor networks, and cognitive radio networks.



**Steve Goddard** (M'86-SM'89-F'96) is a Professor and Chair in the Department of Computer Science & Engineering at the University of Nebraska-Lincoln. He received the B.A. degree in computer science and mathematics from the University of Minnesota (1985). He received the M.S. and Ph.D. degrees in computer science from the University of North Carolina at Chapel Hill (1995, 1998). His research interests are cyber-physical, embedded, real-time and distributed systems, and wireless sensor networks, with an emphasis in real-time, rate-based scheduling

theory.