University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

CSE Conference and Workshop Papers | Computer Science and Engineering, Department of

2009

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

Yunbo Wang
University of Nebraska at Lincoln, yunbow@cse.unl.edu

Mehmet C. Vuran
University of Nebraska at Lincoln, mcvuran@cse.unl.edu

Steve Goddard
University of Nebraska-Lincoln, goddard@cse.unl.edu

Follow this and additional works at: http://digitalcommons.unl.edu/cseconfwork

 Part of the Computer Sciences Commons

Wang, Yunbo; Vuran, Mehmet C.; and Goddard, Steve, "Cross-layer Analysis of the End-to-end Delay Distribution in Wireless Sensor Networks" (2009). *CSE Conference and Workshop Papers*. 144.
http://digitalcommons.unl.edu/cseconfwork/144

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Conference and Workshop Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

2009 30th IEEE Real-Time Systems Symposium

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

Yunbo Wang, Mehmet C. Vuran and Steve Goddard
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Email: {yunbow, mcvuran, goddard}@cse.unl.edu

***Abstract*—Emerging applications of wireless sensor networks (WSNs) require real-time quality of service (QoS) guarantees to be provided by the network. However, designing real-time scheduling and communication solutions for these networks is challenging since the characteristics of QoS metrics in WSNs are not well known yet. Due to the nature of wireless connectivity, it is infeasible to satisfy worst-case QoS requirements in WSNs. Instead, probabilistic QoS guarantees should be provided, which requires the definition of probabilistic QoS metrics. To provide an analytical tool for the development of real-time solutions, in this paper, the distribution of end-to-end delay in multi-hop WSNs is investigated. Accordingly, a comprehensive and accurate cross-layer analysis framework, which employs a stochastic queueing model in realistic channel environments, is developed. This framework captures the heterogeneity in WSNs in terms of channel quality, transmit power, queue length, and communication protocols. A case study with the TinyOS CSMA/CA MAC protocol is conducted to show how the developed framework can analytically predict the distribution of end-to-end delay. Testbed experiments are provided to validate the developed model. The cross-layer framework can be used to identify the relationships between network parameters and the distribution of end-to-end delay and accordingly, to design real-time solutions for WSNs. Our ongoing work suggests that this framework can be easily extended to model additional QoS metrics such as energy consumption distribution. To the best of our knowledge, this is the first work to investigate probabilistic QoS guarantees 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 [1]. Increasingly, a large number of WSN applications require real-time quality of service (QoS) guarantees [2]. 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 strict QoS guarantees in these multi-hop networks. Consequently, a probabilistic analysis of QoS guarantees 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], [4], [5], [6], [7]. However, complex and cross-layer interactions in multi-hop WSNs prevent complete characterization of the delay through only the mean and variance measures. Several efforts have been made to provide probabilistic *bounds* on delay. As an example, the concept of Network Calculus [8] has been extended to derive probabilistic bounds for delay through worst case analysis [9], [10]. However, because of the randomness in wireless communication and the low power nature of the communication links in WSNs, these worst case bounds cannot capture the stochastic behavior of end-to-end delay. Moreover, work on real-time queueing theory [11], [12] 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 [13], [14], [15], [16], [17] 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, we develop a comprehensive and accurate cross-layer analysis framework to *characterize the end-to-end delay distribution* in WSNs. 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.

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 our Markovian model is provided. The detailed derivation of the single-hop delay distribution is described in Section IV with a case study for the CSMA/CA MAC protocol in Section V. Derivation of the end-to-end delay distribution is then given in Section VI. Experimental results are provided in Section VII to validate the developed model. Finally, Section VIII concludes the paper.

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

1052-8725/09 $26.00 © 2009 IEEE
DOI 10.1109/RTSS.2009.27

IEEE computer society

II. Related Work

The problem of probabilistic QoS guarantees is not trivial and has attracted a large amount of research in recent years. The concept of Network Calculus [8] has been extended to support probabilistic delay *bounds* in [10], [9], [18], [19]. The network calculus and its probabilistic extensions are based on the min-plus algebra to provide traffic curves and service curves, which are deterministic (or statistic) 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 [11], [12] 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 [11] in that we use a stochastic queuing model for the analysis. In contrast, we do not focus on the real-time scheduling problem, which has been discussed intensively in the literature [11], [12], [20]. Rather, we aim to provide an analytical tool to help develop real-time scheduling and communication solutions.

Recently, a large amount of studies have analyzed the delay distribution of MAC protocols for wireless networks and WSNs, in particular. The access delay of several MAC protocols has been investigated including IEEE 802.11b DCF protocol [21] in [13], [16], [17], IEEE 802.15.4 protocol in [15], and TDMA protocols in [14]. However, in these studies, a broadcast network is considered, where each node can hear the transmission of each other. Moreover, in [13], [16], [17], 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 [22]. While the distribution of the number of retransmissions is obtained, the transmission time is regarded 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 [23], the end-to-end delay distribution in a linear network is derived for homogeneous networks. However, this model assumes infinite queue lengths at each node, which may not be practical considering the resource constraints of sensor nodes. Finally, in [24] and [25], empirical measurements and estimations are used to route packets so that a probabilistic guarantee of delay is provided. These solutions exploit on-the-fly measurements but do not provide analytical results.

It can be observed that accurately characterizing end-to-end delay in WSNs is still an open problem. In the following, we provide a cross-layer analysis framework toward this goal.

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. Each node i is characterized by its input traffic rate, λ_i, queue length, M_i, the maximum number of retransmission attempts, N_{tx}, data rate, R_i, and a MAC protocol. Furthermore, a log-normal fading channel model is considered. Accordingly, we are interested in the following two problems:

1) For a given network with a certain MAC protocol and node parameters described above, what is the single-hop delay distribution, $f_{sh(i,j)}(t)$, between two nodes i and j for a new arriving packet?
2) Given the single-hop delay distribution, what is the end-to-end delay distribution, $f_{e(i,s)}(t)$ between a node i and a sink s in the network?

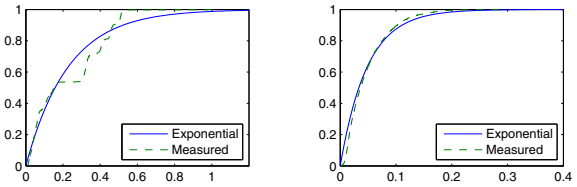
We also consider a heterogeneous network for this analysis, where the heterogeneity is defined in terms of channel conditions, (the packet error rate, PER), traffic rate λ_i, and transmission power P_{tx}. In this section, we discuss the intuition of our approach and provide an overview of our solutions for the two problems above. The detailed description of the framework is deferred to Sections IV-VI.

A. Single-hop Delay Distribution

To solve the aforementioned single hop delay problem, each node is modeled according to a queueing 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 [26]. 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 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. Accordingly, input traffic rate is $\lambda = \lambda_l + \lambda_r$. Both of these types of traffic should be transmitted to another neighbor on the path to the sink. 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 [27]. Note



(a) Low traffic, CBR (b) High traffic, CBR

(c) Low traffic, Poisson (d) High traffic, Poisson

Fig. 1. The distribution of inter-arrival time for different types of traffics. A 10-hop chain is used. Every node in average generates 0.4 packets per second in (a) and (c), and generates 4 packets per second in (b) and (d). Packets are generated in CBR model in (a) and (b), and Poisson process in (c) and (d).

that in some applications, the traffic generated for the physical event can be bursty. For simplicity this 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. While 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). 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 Figure 1 along with an exponential distribution model for four cases.[1] 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 λ at some 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 transmission service, the system is essentially governed by a Quasi-Birth-Death (QBD) process [26] 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), $\{X_n\}$ is composed of M layers, where each layer m ($1 \leq m \leq M$) represents the state where there are m packets in the queue and M is the queue capacity. Whenever, there is no packet in the queue, the system is in the *Idle* state, denoted as Layer 0 in Fig. 2(a). Otherwise, i.e., $m > 0$, the system tries to transmit the packet through several transmission attempts until either the packet is successfully transmitted or the maximum number of transmission attempts, N_{tx}, is exceeded. Accordingly, layer m, in $\{X_n\}$ is composed of N_{tx} blocks, denoted as $\{Z_n\}$. As shown in Fig. 2(c), each block, $\{Z_n\}$, models a single transmission attempt and depends on the MAC protocol employed in the system. Packets are dropped if they arrive at a full queue or if all N_{tx} transmission attempts fail. Then, the equilibrium state distribution of the system can be derived using $\{X_n\}$.

While $\{X_n\}$ can be directly used to derive the *expected* single-hop delay of a packet, the *distribution* of single-hop delay can be found by focusing on a particular packet that enters the system at time $t = t_0$. Then, the service time of the packet is the time spent until being transmitted or dropped. To derive the delay distribution, we use a second Markov chain, $\{Y_n\}$, as an absorbing version of $\{X_n\}$. As shown in Fig. 2(b), in $\{Y_n\}$, the idle state 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.

Before the packet arrives, i.e., at $t = t_0^-$, the system is in one of the states according to the equilibrium state probability vector of $\{X_n\}$. Using this vector, the initial state probability vector for $\{Y_n\}$ is found. To derive the single-hop delay distribution, the absorption time of the Markov chain $\{Y_n\}$ is considered. More specifically, the *pmf* of the number of transitions, K, needed for $\{Y_n\}$ to absorb can be obtained by the phase-type distribution [26]:

$$f_K(k) = \boldsymbol{\alpha}_Y \boldsymbol{P}_Y^{k-1} \boldsymbol{t}_Y, \tag{1}$$

where $\boldsymbol{\alpha}_Y$ is the initial state probability vector of $\{Y_n\}$, $\boldsymbol{P}_Y$ is the transition probability matrix among the transient states, and $\boldsymbol{t}_Y$ is the transition probability matrix from the transient states to the absorbing states. Accordingly, the distribution of the single-hop delay is given by

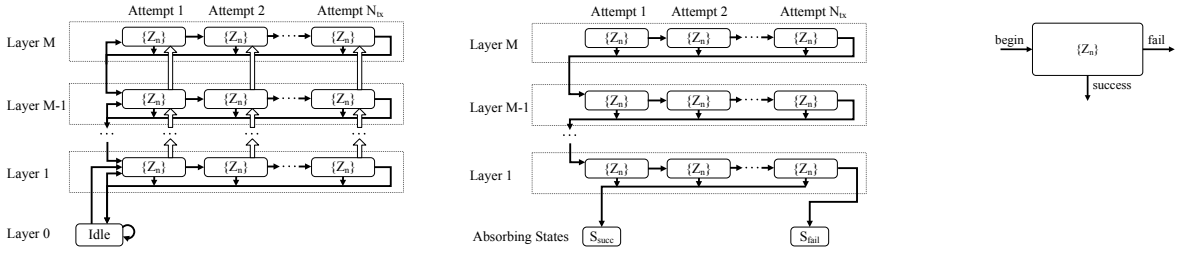
$$f_{sh}(t) = f_K\left(\lfloor \frac{t}{T_u} \rfloor\right). \tag{2}$$

In Section IV, we provide the details about the construction of these two Markov chains, based on which the single-hop delay distribution is obtained. The TinyOS CSMA MAC protocol is then analyzed as a special case in Section V.

B. End-to-end Delay Distribution

With each hop modeled as a Geom/PH/1/M queue, the entire network is considered a queueing network. Nodes are interrelated according to the traffic constraints. More specifically, the successfully transmitted traffic rate from one node should be equal to the sum of the incoming relay traffic rate at each of the next-hop neighbors of the node.

[1]The exponential distribution shown in the figures are chosen such that their mean equals the measured inter-arrival times.



(a) $\{X_n\}$ (b) $\{Y_n\}$ (c) $\{Z_n\}$

Fig. 2. The structures of Markov chains $\{X_n\}$ and $\{Y_n\}$ are shown in (a) and (b). They consist of blocks of $\{Z_n\}$, whose structure is determined by the MAC protocol.

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 [28]. 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 (36) as will be explained in Section VI.

According to the overview of the framework discussed in this section, next, we provide the details of the framework. We derive the single-hop delay distribution in Section IV and a case study for the CSMA/CA protocol is presented in Section V. In Section VI, given the single-hop delay distribution, the derivation of the end-to-end delay distribution is discussed. The analytical solutions are validated through experiments in Section VII.

IV. Single-hop Delay Distribution

In this section, we introduce our general approach for the single-hop delay distribution. We are interested in the distribution of the absorption time, K, for $\{Y_n\}$ as expressed in (1). The transition matrices, $\boldsymbol{P}_Y$ and $\boldsymbol{t}_Y$, in (1) are constructed with the knowledge of the traffic rate, λ, and the MAC protocol operation. Moreover, $\boldsymbol{\alpha}_Y$ in (1) is found according to the equilibrium behavior of the node, which is governed by the Markov chain, $\{X_n\}$, as will be explained in the following.

At any time, a typical WSN node is either in a packet transmission attempt or an idle state depending on the MAC protocol. To make the model tractable, the receiving time for the packets are ignored. This is motivated by the fact that the receiving time for a packet is already included in the transmitting time and is usually much smaller than the service time of the packet, which includes the channel access delay in addition to packet transmission.

For the derivation of $\{Y_n\}$, first $\{X_n\}$ is generated. The difference between $\{X_n\}$ and $\{Y_n\}$ lies in the inter-layer transitions. For $\{X_n\}$, an idle state in layer 0 is considered as shown in Fig. 2(a). A node enters the idle state when it has no packets to transmit. When there is a new arriving packet, the Markov chain transits to the layer above and when the service for a packet is complete (successfully transmitted or dropped), the chain transits to a lower layer. This is actually a description of the equilibrium behavior of the node handling packet transmissions. On the other hand, in $\{Y_n\}$, there is no idle state. Instead, the absorbing chain starts when a packet arrives, and absorbs when that particular packet is served. Note that in a queue without priority, new packets that arrive after the absorbing chain starts do not interfere with the waiting time of the packet. Therefore, in $\{Y_n\}$, there are no transitions to higher layers.

In the following, we will first discuss the construction of the Markov chains $\{X_n\}$ and $\{Y_n\}$. Then, the equilibrium state probability vector $\boldsymbol{\pi}$ is obtained for $\{X_n\}$, which is then used to derive the initial probability $\boldsymbol{\alpha}_Y$ for $\{Y_n\}$. Finally the *pmf* of absorbing time, K, for $\{Y_n\}$ is obtained.

A. Constructing Markov chain $\{X_n\}$

As shown in Figure 2, Markov chains $\{X_n\}$ and $\{Y_n\}$ are divided into layers and each layer is further divided into $\{Z_n\}$ blocks, which represents the process of a single transmission attempt. The Markov chain, $\{Z_n\}$, is characterized by the following:

- $\boldsymbol{P}_Z$, the transition probability matrix among the states in $\{Z_n\}$,
- $\boldsymbol{\alpha}_Z$, the initial probability vector for $\{Z_n\}$, and
- $\boldsymbol{t}_Z^s$ and $\boldsymbol{t}_Z^f$, the probability vector from each state in $\{Z_n\}$ to complete the transmission attempt successfully or unsuccessfully, respectively.

The states and the transitions related to $\{Z_n\}$ depend on the MAC protocol employed. For now, we assume that these matrices are known and a case study to obtain them for the TinyOS CSMA/CA protocol is provided in Section V. Accordingly, the transition probability matrix among the states in a single layer in $\{X_n\}$ is

$$\boldsymbol{P}_L = \begin{bmatrix} \boldsymbol{P}_Z & \boldsymbol{t}_Z^f\boldsymbol{\alpha}_Z & & \boldsymbol{0} \\ & \ddots & \ddots & \\ & & \boldsymbol{P}_Z & \boldsymbol{t}_Z^f\boldsymbol{\alpha}_Z \\ \boldsymbol{0} & & & \boldsymbol{P}_Z \end{bmatrix}, \tag{3}$$

where the number of $\boldsymbol{P}_Z$ blocks in $\boldsymbol{P}_L$ is equal to N_{tx}, i.e, the maximum number of attempts for each packet transmission. Moreover, the initial probability vector, $\boldsymbol{\alpha}_L$, and the probability vectors, $\boldsymbol{t}_L^s$ and $\boldsymbol{t}_L^f$ to complete a layer in success and failure are

$$\boldsymbol{\alpha}_L = \begin{bmatrix} \boldsymbol{\alpha}_Z & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix} \tag{4}$$

$$\boldsymbol{t}_L^s = \begin{bmatrix} \boldsymbol{t}_Z^s & \boldsymbol{t}_Z^s & \cdots & \boldsymbol{t}_Z^s \end{bmatrix}^{\mathrm{T}} \tag{5}$$

$$\boldsymbol{t}_L^f = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{t}_Z^f \end{bmatrix}^{\mathrm{T}} \tag{6}$$

respectively, where there are N_{tx} blocks in each vector, corresponding to each of the transmission attempts in the layer.

The transition probability matrix, $\boldsymbol{Q}_X$, 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

$$\boldsymbol{A}_u = \lambda \boldsymbol{P}_L, \tag{7}$$

where λ is the traffic rate in each time unit. The transition probability matrix at the same level m, $1 \leq m \leq M-1$, is

$$\boldsymbol{A}_s = \lambda \boldsymbol{t}_L \boldsymbol{\alpha}_L + (1-\lambda)\boldsymbol{P}_L, \tag{8}$$

where $\boldsymbol{t}_L = \boldsymbol{t}_L^s + \boldsymbol{t}_L^f$ is the completion probability vector in each layer regardless of success or failure. In (8), the first term captures the case where a locally generated packet arrives at the same time unit that a packet service is completed. The second term in (8) is for the case where neither service completion nor new packet arrivals occur during the time unit. In both cases, the Markov chain stays at the same level m, $1 \leq m \leq M-1$.

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

When there is no packet arrival and the current packet service is completed at the current time unit, 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

$$\boldsymbol{A}_d = (1-\lambda)\boldsymbol{t}_L \boldsymbol{\alpha}_L. \tag{9}$$

The transition probabilities are similar when the idle state is considered as shown below:

$$\boldsymbol{A}_{u0} = \lambda \boldsymbol{\alpha}_L,\ \boldsymbol{A}_{d0} = (1-\lambda)\boldsymbol{t}_L,\ A_{s0} = 1-\lambda, \tag{10}$$

where λ is the input packet rate, $\boldsymbol{\alpha}_L$ is given in (4), and $\boldsymbol{t}_L = \boldsymbol{t}_L^s + \boldsymbol{t}_L^f$, which are given in (5) and (6), respectively. When the node is idle and a new packet arrives, the Markov chain transits from the idle state to layer 1 according to $\boldsymbol{A}_{u0}$. 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 state according to $\boldsymbol{A}_{d0}$. Finally, the transition probability with which the node stays in the idle state is given in A_{s0}.

Using (7)-(10), the transition probability matrix $\boldsymbol{Q}_X$ for the entire recurrent Markov chain $\{X_n\}$, can be constructed as follows:

$$\boldsymbol{Q}_X = \begin{bmatrix} A_{s0} & \boldsymbol{A}_{u0} & & & \boldsymbol{0} \\ \boldsymbol{A}_{d0} & \boldsymbol{A}_s & \boldsymbol{A}_u & & \\ & \boldsymbol{A}_d & \ddots & \ddots & \\ & & \ddots & \boldsymbol{A}_s & \boldsymbol{A}_u \\ \boldsymbol{0} & & & \boldsymbol{A}_d & \boldsymbol{A}_s + \boldsymbol{A}_u \end{bmatrix}, \tag{11}$$

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. Moreover, since packet arrivals are governed by a Bernoulli process, 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 in $\boldsymbol{Q}_X$ corresponds to the transition probabilities from and to the idle state, respectively.

B. Equilibrium State Probabilities for $\{X_n\}$

After $\boldsymbol{Q}_X$ is known, the equilibrium state probability vector, $\boldsymbol{\pi}$, for $\{X_n\}$ can be calculated. This vector is used to derive the initial probability vector for $\{Y_n\}$ in order to derive the single-hop delay distribution. Suppose $\boldsymbol{\pi} = (\pi_0, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_m, \ldots, \boldsymbol{\pi}_M)$, where $\boldsymbol{\pi}_m$ is the stationary probability vector for states in layer m, $0 \leq m \leq M$. According to [27]:

$$\boldsymbol{\pi}_m = \boldsymbol{\pi}_{m-1}\boldsymbol{R},\ 2 \leq m \leq M-1, \tag{12}$$

where $\boldsymbol{R}$ is a constant rate matrix. Since,

$$\boldsymbol{\pi}_{m-1}\boldsymbol{A}_u + \boldsymbol{\pi}_m \boldsymbol{A}_s + \boldsymbol{\pi}_{m+1}\boldsymbol{A}_d = \boldsymbol{\pi}_m,\ 2 \leq m \leq M-1,$$

the following iterative computation is conducted to solve for $\boldsymbol{R}$ [27]:

$$\boldsymbol{R}^{(n+1)} = -\boldsymbol{A}_u(\boldsymbol{A}_s - \boldsymbol{I})^{-1} - (\boldsymbol{R}^{(n)})^2 \boldsymbol{A}_d \boldsymbol{A}_s^{-1}, \tag{13}$$

where $\boldsymbol{R}^{(0)} = \boldsymbol{0}$. The iteration continues until there is a negligible difference between $\boldsymbol{R}^{(n+1)}$ and $\boldsymbol{R}^{(n)}$. Consequently, considering $\boldsymbol{\pi}_m = \boldsymbol{\pi}_1 \boldsymbol{R}^{m-1}, 2 \leq m \leq M-1$, $\boldsymbol{\pi}$ can be solved for $M \geq 2$, as follows:

$$\begin{cases} \pi_0 + \boldsymbol{\pi}_1(\boldsymbol{I} - \boldsymbol{R}^{M-1})(\boldsymbol{I} - \boldsymbol{R})^{-1}\boldsymbol{e} + \boldsymbol{\pi}_M \boldsymbol{e} = 1 \\ \pi_0(A_{s0} - 1) + \boldsymbol{\pi}_1 \boldsymbol{A}_{d0} = 0 \\ \pi_0 \boldsymbol{A}_{u0} + \boldsymbol{\pi}_1(\boldsymbol{A}_s - \boldsymbol{I} + \boldsymbol{R}\boldsymbol{A}_d) = \boldsymbol{0} \\ \boldsymbol{\pi}_{M-1}\boldsymbol{A}_u + \boldsymbol{\pi}_M(\boldsymbol{A}_s + \boldsymbol{A}_u - \boldsymbol{I}) = \boldsymbol{0} \end{cases} \tag{14}$$

and for $M = 1$, as follows:

$$\begin{cases} \pi_0 + \boldsymbol{\pi}_1 \boldsymbol{e} = 1 \\ \pi_0(A_{s0} - 1) + \boldsymbol{\pi}_1 \boldsymbol{A}_{d0} = 0 \\ \pi_0 \boldsymbol{A}_{u0} + \boldsymbol{\pi}_1(\boldsymbol{A}_s + \boldsymbol{A}_u - \boldsymbol{I}) = \boldsymbol{0} \end{cases} \tag{15}$$

where $\boldsymbol{e}$ is a properly dimensioned column vector of all 1's.

C. Absorbing time for $\{Y_n\}$

To obtain the distribution of single-hop delay for a packet, $\{Y_n\}$ is constructed similar to $\{X_n\}$ as explained in Section III. 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 with the probability $p_{qf} = \boldsymbol{\pi}_M \boldsymbol{A}_u \boldsymbol{e}$. 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 $\boldsymbol{\pi}' = \boldsymbol{\pi}\boldsymbol{Q}_Y^{up}$, where $\boldsymbol{Q}_Y^{up}$ is the transition probability matrix of $\{Y_n\}$ conditioned on the fact that the new packet arrives. It is derived from $\boldsymbol{Q}_X$ in (11) by assigning $\lambda = 1$ (7), (8), and (9) and replacing $\boldsymbol{A}_s + \boldsymbol{A}_u$ with $\boldsymbol{A}_s$. Note that $\boldsymbol{A}_u$ in the bottom-right block accounts for the transition that will cause a packet to drop because of a full queue. Then, $\boldsymbol{\pi}'$ is the initial probability vector for $\{Y_n\}$.

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

$$\boldsymbol{Q}_Y = \begin{bmatrix} 1 & 0 & \boldsymbol{0} \\ 0 & 1 & \boldsymbol{0} \\ \boldsymbol{t}_Y^s & \boldsymbol{t}_Y^f & \boldsymbol{P}_Y \end{bmatrix}, \tag{16}$$

where the transition probabilities from and to the absorbing states S_{succ} and S_{fail} are listed in the first two rows and columns, respectively. The bottom-right block of $\boldsymbol{Q}_Y$ is the

transition probability matrix among the transient states, and is given by

$$\boldsymbol{P}_Y = \begin{bmatrix} \boldsymbol{P}_L & & & \boldsymbol{0} \\ \boldsymbol{t}_L\boldsymbol{\alpha}_L & \boldsymbol{P}_L & & \\ & \ddots & \ddots & \\ \boldsymbol{0} & & \boldsymbol{t}_L\boldsymbol{\alpha}_L & \boldsymbol{P}_L \end{bmatrix}. \tag{17}$$

This is obtained from (11) by removing the first row and first column, and letting $\lambda = 0$ in (7), (8), and (9) for each remaining block. The transition probability vectors from each of the transient states to the absorbing states are

$$\boldsymbol{t}_Y^s = \begin{bmatrix} \boldsymbol{t}_L^s & \boldsymbol{0} & \boldsymbol{0} & \cdots \end{bmatrix}^{\mathrm{T}}, \ \boldsymbol{t}_Y^f = \begin{bmatrix} \boldsymbol{t}_L^f & \boldsymbol{0} & \boldsymbol{0} & \cdots \end{bmatrix}^{\mathrm{T}}, \tag{18}$$

respectively, where $\boldsymbol{t}_L^s$ and $\boldsymbol{t}_L^f$ are given in (5) and (6), respectively. Finally, the *pmf* of the number of transitions, K, a packet should wait before being transmitted and dropped are

$$f_K^s(k) = \boldsymbol{\alpha}_Y \boldsymbol{P}_Y^{k-1}\boldsymbol{t}_Y^s, \ \ f_K^f(k) = \boldsymbol{\alpha}_Y \boldsymbol{P}_Y^{k-1}\boldsymbol{t}_Y^f, \tag{19}$$

respectively, where $\boldsymbol{\alpha}_Y = (\pi'_1, \pi'_2, \cdots, \pi'_M)$, i.e., $\boldsymbol{\pi}'$ without the first element π'_0. Adding $f_K^s(k)$ and $f_K^f(k)$ yields (1) and, $f_{sh}(t)$, the distribution of successful single-hop transmission delay measured in time units is obtained:

$$f_{sh}(t) = f_K^s\left(\lfloor \frac{t}{T_u} \rfloor\right). \tag{20}$$

Using this model, the probability of successful transmission can also be found as follows:

$$p_{deli} = \sum_{k=1}^{+\infty} f_K^s(k) = \boldsymbol{\alpha}_Y(\boldsymbol{I} - \boldsymbol{P}_Y)^{-1}\boldsymbol{t}_Y^s. \tag{21}$$

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

$$\begin{aligned} \mu_K &= \frac{\sum_{k=1}^{+\infty} k f_K^s(k)}{\sum_{k=1}^{+\infty} f_K^s(k)} = \frac{\sum_{k=1}^{+\infty} k\boldsymbol{\alpha}_Y \boldsymbol{P}_Y^{k-1}\boldsymbol{t}_Y^s}{p_{deli}}, \\ &= \boldsymbol{\alpha}_Y(\boldsymbol{I} - \boldsymbol{P}_Y)^{-2}\boldsymbol{t}_Y^s / p_{deli}, \end{aligned} \tag{22}$$

$$\begin{aligned} \sigma_K^2 &= \frac{\sum_{k=1}^{+\infty} k^2 f_K^s(k)}{\sum_{k=1}^{+\infty} f_K^s(k)} - \mu_K^2 \\ &= \frac{\boldsymbol{\alpha}_Y(2(\boldsymbol{I} - \boldsymbol{P}_Y)^{-3} - (\boldsymbol{I} - \boldsymbol{P}_Y)^{-2})\boldsymbol{t}_Y^s}{p_{deli}} - \mu_K^2, \end{aligned} \tag{23}$$

respectively.

Next, we describe a case study for the construction of the Markov chain block, $\{Z_n\}$, for a single transmission attempt using TinyOS CSMA protocol.

V. Case Study: TinyOS CSMA/CA protocol

In this section, we illustrate how single-hop delay distribution can be obtained for a particular MAC protocol. We use the TinyOS default CSMA/CA protocol [29], which is widely adopted by applications due to the popularity of TinyOS. Similar to the IEEE 802.15.4 protocol [30], 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 [15] for our analysis.

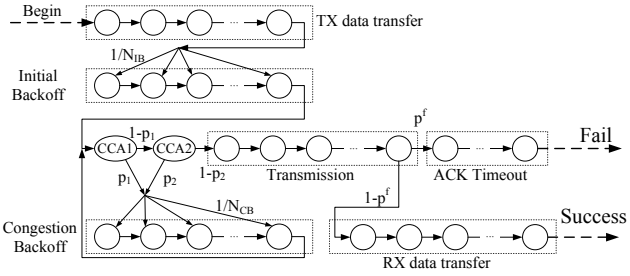


Fig. 3. Markov chain structure for each attempt for TinyOS CSMA protocol. N_{IB} and N_{CB} are the number of states representing the initial backoff and congestion backoff, respectively.

Since multi-hop traffic and the hidden node problem are not considered in [15], 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 TinyOS CSMA/CA protocol. Instead, we illustrate how the existing models for delay analysis of MAC protocols can be extended for our framework to determine the end-to-end delay distribution.

The Markov chain, $\{Z_n\}$, that models each transmission attempt is depicted in Fig. 3. 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 data transfer* and *RX data transfer* in Figure 3, respectively.

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.

For each transmission attempt, the corresponding block of the Markov chain is depicted in Figure 3, which is characterized by three variables in the chain: p_1 and p_2 are the probability that the node senses the channel busy in the first and second CCA, respectively and p_i^f is the probability that a transmission attempt fails due to either channel noise or collisions. The derivation of each variable is explained next.

Suppose a packet is being transmitted from node i to node j. We start by defining the probability of successful transmission, $p_{i,j}^s$, between two nodes i and j as follows:

$$p_{i,j}^s = p_i^w(1 - p_{i,j}^{coll})(1 - PER_{i,j}), \tag{24}$$

where p_i^w is the probability that only node i starts to transmit a packet, $p_{coll,i,j}$ is the probabilities of collision due to hidden terminal transmissions, and $PER_{i,j}$ is the packet error rate due to channel noise. In the remaining of this section, we describe the derivation of each term in (24), which will be used to find p_1, p_2, and p_i^f in $\{Z_n\}$.

For the derivations, 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., $\{i,j\} \in \mathbb{C}_{i,j}$, 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 $\mathbb{H}_{j,i}$.[2] 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. Furthermore, the number of nodes in these areas depends on the network topology and density. We consider nodes are distributed according to a Poisson distribution with density ρ. Accordingly, the number of nodes in an area of size $|A|$ is $\rho|A|$.

The first term p_i^w in (24) is found by considering that no node $k \in \mathbb{C}_j$ other than node i starts to transmit as follows:

$$p_i^w = \phi_i(1-p_1)(1-p_2)\prod_{k\in\mathbb{C}_j}(1-\phi_k), \tag{25}$$

where ϕ_i is the probability that node i is in the first CCA state and is given in $\boldsymbol{\pi}$, p_1 and p_2 are the probability that the node senses the first and the second CCA busy, respectively. Note that since heterogeneous network traffic is considered, ϕ_i may be different for different nodes.

The second parameter, p_1, in (25) is the probability that a node senses channel busy in the first CCA. It is equal to the probability that at least one other neighbor of i starts a transmission in the past L_{TX} or $L_{TX}+L_{ACK}$ slots, depending on whether the transmission is successful or not. L_{TX} and L_{ACK} are the number of slots required for a packet and an acknowledgment transmission, respectively. Thus,

$$p_1 = p_{send,\mathbb{C}_i}\, L_{TX} + p_{ack}\, L_{ACK}, \tag{26}$$

where $p_{send,\mathbb{C}_i}$ is the probability with which *at least one* node $k \in \mathbb{C}_i$ begins a transmission, and is given by

$$p_{send,\mathbb{C}_i} = (1-p_1)(1-p_2)\left(1-\prod_{k\in\mathbb{C}_i}(1-\phi_k)\right), \tag{27}$$

and p_{ack} in (26) is the probability that an ACK packet is transmitted by at least one node in $\mathbb{C}_i$ during a time unit. Its value depends on the number of successful transmissions targeted into $\mathbb{C}_i$ and is not trivial to determine. Motivated by the fact that the traffic rate and channel conditions do not change dramatically within a small area in most WSN applications, p_{ack} is approximated by the average probability of successful transmissions *from* inside $\mathbb{C}_i$. Thus,

$$p_{ack} = \sum_{k\in\mathbb{C}_i} p_k^s, \tag{28}$$

and p_k^s is the average value of $p_{k,r}^s$ defined in (24), with $r \in \mathbb{C}_i$, weighted on the traffic load node k sends to each of node r in its proximity.

[2]With a slight abuse of notation, in the following, we exclude the nodes i and j from the nodes in these areas.

Next, we consider p_2 in (25), which is the probability that a node senses the channel busy in the second CCA given that the first CCA is successful. According to [15]:

$$p_2 = \left[1-\frac{2-p_{\mathbb{C}_i}^{nc}}{2-p_{\mathbb{C}_i}^{nc}+\frac{1}{1-\prod_{k\in\mathbb{C}_i}(1-\phi_k)}}\right](1-\prod_{k\in\mathbb{C}_i}(1-\phi_k)) + \frac{1-p_{\mathbb{C}_i}^{nc}}{2-p_{\mathbb{C}_i}^{nc}+\frac{1}{\prod_{k\in\mathbb{C}_i}(1-\phi_k)}}, \tag{29}$$

where $p_{\mathbb{C}_i}^{nc}$ is the probability that a collision is observed on the channel on the condition that a transmission was going on. It is given by

$$p_{\mathbb{C}_i}^{nc} = 1 - \frac{\sum_{k\in\mathbb{C}_i} p_k^w}{p_{send,\mathbb{C}_i}}. \tag{30}$$

Using (26) and (29) in (25), the probability, p_i^w, that only node i starts to transmit a packet can be found.

The second term in (24) is the collision rate, $p_{i,j}^{coll}$, and models the hidden node collisions. A collision of a packet from i to j with another packet occurs: (1) when a node in $\mathbb{H}_{j,i}$ transmits a packet in the $(-L_{TX}, +L_{TX})$ window around the time instance when i starts transmission, or (2) when a node in $\mathbb{H}_{j,i}$ transmits an ACK in the $(-L_{ACK}, +L_{TX})$ window around the time instance when i starts transmission. Therefore $p_{i,j}^{coll}$ is

$$p_{i,j}^{coll} = 2L_{TX}\, p_{send,\mathbb{H}_{j,i}} + (L_{TX}+L_{ACK})p_{ack,\mathbb{H}_{j,i}}, \tag{31}$$

where $p_{ack,\mathbb{H}_{j,i}}$ can be defined the similar way as in (28), and can be obtained by solving (24)(28)(31) for every node in the network. Considering in most WSN applications the collision probability is rather small, we use $p_{send,\mathbb{H}_{j,i}}$ to approximate $p_{ack,\mathbb{H}_{j,i}}$.

Finally, the third component in (24) is the packet error rate, $PER_{i,j}$ between i and j, which depends on the transmission distance, transmission power, random multi-path and shadowing effects. In our model, we define the expected packet error rate for a pair of nodes according to the log-normal fading model in [31] as follows:

$$\mathbb{E}[PER_{i,j}] = \int_{-\infty}^{+\infty} PER(\psi)Pr(\Psi_{dB}(d)=\psi)\mathrm{d}\psi, \tag{32}$$

where the integration is taken w.r.t. the signal to noise ratio (SNR), ψ, $PER(\psi)$ is the PER that corresponds to a particular SNR, which can be found for the MicaZ or TelosB motes according to [32], d is the distance between i and j, and $\Psi(d)$ is given by [31]:

$$\Psi_{dB}(d) = P_t - P_n - PL(d_0) - 10\eta\log_{10}(\frac{d}{d_0}) + X_\sigma, \tag{33}$$

where P_t is the transmit power, P_n is the noise power in dBm, and $P_r(d)$ is the received signal power. $PL(d_0)$ is the path loss at a reference distance d_0, η is the path loss exponent, and X_σ is the log-normal random variable.

As a result, using (25), (31), and (32) in (24), the probability of successful transmission can be found. Consequently, the probability that a transmission from i to j fails due to either channel noise or collision, given that the channel access was successful is given by

$$p_{i,j}^f = 1 - \frac{p_{i,j}^s}{\phi_i(1-p_1)(1-p_2)}, \tag{34}$$

where $p^s_{i,j}$, p_1, and p_2 are given in (24), (26), and (29), respectively. Then, $p^f_{i,j}$ is averaged among all destinations, j, as the approximation of p^f_i for each node i. As suggested in (24), the value of $p^f_{i,j}$ depends on the channel conditions and the collision probability. Considering a channel-aware routing protocol is employed, $p^f_{i,j}$ 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^f_i, is the same for all packets in the queue.

As discussed in the beginning of this section, the Markov chain, $\{Z_n\}$, is characterized by three probability values: p_1, p_2, and p^f_i. Each of these values depends on each other as well as ϕ_i, which is the probability that the node i is in the first CCA state. Note that ϕ_i, p_1 and p_2 cannot be determined without the knowledge of $\boldsymbol{\pi}$, which can only be obtained after the construction the Markov chain as explained in Section IV. Consequently, an iterative procedure is used to find these parameters. First, an initial guess of ϕ_j, p_1 and p_2 is used to construct the Markov chains for each node, based on which $\boldsymbol{\pi}$ is calculated. Then, values for ϕ_i, p_1 and p_2 are updated accordingly to the knowledge of $\boldsymbol{\pi}$. The calculation of ϕ_i, p_1, p_2, and $\boldsymbol{\pi}$ is conducted iteratively, until the difference of the value for any variable between two iterations is negligible.

Of interest, the calculation of the iterative procedure is quite intensive but still affordable. In our computing environment with a Xeon 5150 CPU working at 2.66GHz and 2G RAM, the Matlab program runs for less than 10 seconds for a typical hop with 6 neighbors, with $N_{tx} = 3$ and $M = 5$.

VI. End-to-end Delay Distribution

To model the end-to-end delay distribution in a heterogeneous network, we consider the steady-state period of dynamic routing policies. The calculation of the end-to-end delay distribution depends on the topology and the routing algorithm used. Suppose in a generic network, each node i generates a local traffic of $\lambda_{l(i,j)}$ to each destination j. Each packet from i is routed to the destination j using a relay $k \in \mathcal{N}_{i,j}$ with probability $p_{i,k,j}$, where $\mathcal{N}_{i,j}$ is the set of potential relays from i to j. Thus, $\sum_{k\in\mathcal{N}_{i,j}} p_{i,k,j} = 1$, $\forall i, j$. We first calculate the relay traffic $\lambda_{r(i,j)}$ from node i to destination j by solving the following equation system for every pair of nodes:

$$\lambda_{r(i,j)} = \sum_{m\in\mathcal{M}_{i,j}} (\lambda_{r(m,j)}+\lambda_{l(m,j)})p_{m,i,j}\ p_{deli,m,i},\ \forall i, j \quad (35)$$

where $\mathcal{M}_{i,j}$ is the set of nodes that use i as the next hop to reach j, $p_{deli,m,i}$ is the probability that a packet is successfully delivered from node m to i, as defined in (21). Accordingly, the overall input traffic rate of a node i can be found as $\lambda_i = \sum_j \lambda_{r(i,j)} + \lambda_{l(i,j)}$, which is then used in (7)-(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.

In a WSN with multiple sinks, the end-to-end delay distribution, $f_{e(i,j)}(t)$, between node i and sink j can be solved in an iterative way as follows:

$$\begin{aligned} f^{(0)}_{e(i,j)}(t) &= f_{sh(i,j)}(t), \\ f^{(n+1)}_{e(i,j)}(t) &= \sum_{k\in\mathcal{N}_{i,j}} f_{sh(i,k)}(t) * f^{(n)}_{e(k,j)}(t)p_{i,k,j} + f_{sh(i,j)}(t)p_{i,j,j}, \end{aligned} \quad (36)$$

where $(*)$ is the convolution operator and $p_{i,k,j}$ is the probability that the routing policy chooses k as the next hop for a packet from i to j. The iteration process terminates when the difference between two consequent iterations is small enough. Note that the traffic load, channel conditions, and transmission power, which affects packet error rate and collision range, varies in a heterogeneous network. Thus, in each iteration, the $p_{deli,i,j}$ and $f_{sh(i,k)}$ are different for each pair of nodes.

The above iteration process can be further simplified for a WSN with a single sink, which is the case for most practical applications. Considering a routing protocol with no routing cycles, the network can be 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 sink node is denoted by index N and (35) becomes

$$\lambda_{r(i,N)} = \sum_{m=1}^{i-1} (\lambda_{r(m,N)} + \lambda_{l(m,N)})p_{m,i,N}\ p_{deli,m,i},\ \forall i, \quad (37)$$

and $\lambda_{r(1,N)} = 0$, where $\lambda_i = \lambda_{r(i,N)} + \lambda_{l(i,N)}$. Note that the traffic load increases for a node closer to the sink. Then, the single-hop delay distribution, $f_{sh(i,j)}$, between each pair of nodes i and j can be obtained and finally, the end-to-end delay distribution is given as

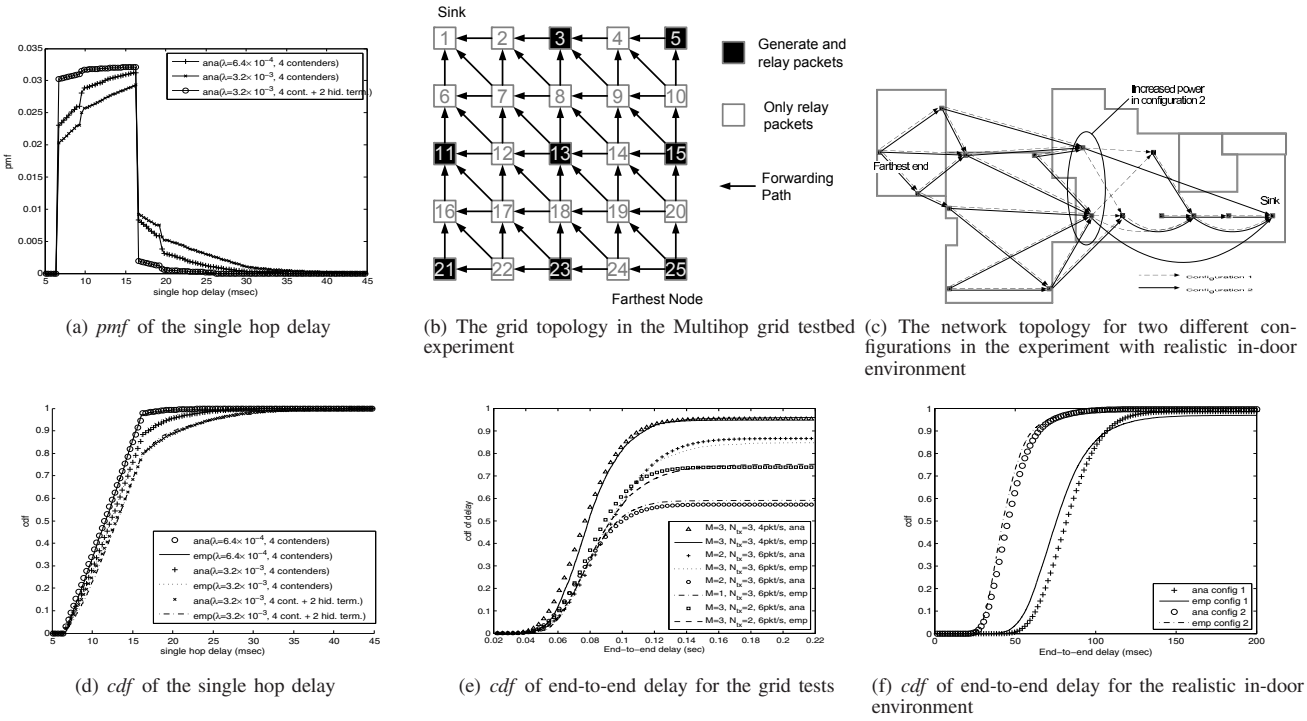
$$\begin{aligned} f_{e(i,N)}(t) \quad = \quad & \sum_{k=i+1}^{N-1} f_{sh(i,k)}(t) * f_{e(k,N)}(t)p_{i,k,N} \\ & + f_{sh(i,N)}(t)p_{i,N,N} \end{aligned} \quad (38)$$

without the need for an iteration. Our testbed experiments show that it takes less than 5 minutes to obtain the end-to-end delay distribution between two nodes in a network consisting of 16 nodes. The calculation is quite intensive, but still affordable for protocol analyses.

In the following section, we use empirical evaluations to validate the analytical model for both single-hop and end-to-end delay distributions.

VII. 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. Moreover, empirical experiments have been conducted on our WSN testbed to validate the results. For the empirical validations, Crossbow TelosB motes with a data rate of 250 kbps are used. The packet size is $l_p = 39$ bytes, consisting of a header of 14 bytes, 23 bytes of payload, and 2 bytes of CRC field. Each node i generates local traffic to be sent to sink s according to a Poisson distribution with rate $\lambda_{l(i,s)}$. 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 a electric pulse to the destination node through a pair of wires. The destination node starts a timer when it receives a pulse, and then 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. Next, we present the evaluation results.



(a) *pmf* of the single hop delay

(b) The grid topology in the Multihop grid testbed experiment

(c) The network topology for two different configurations in the experiment with realistic in-door environment

(d) *cdf* of the single hop delay

(e) *cdf* of end-to-end delay for the grid tests

(f) *cdf* of end-to-end delay for the realistic in-door environment

Fig. 4. Single-hop delay distributions and end-to-end delay distributions in the multi-hop experiments. Both measured empirical experiment results and analytical prediction are given. *cdf*s are compared between analytical (ana) and empirical (emp) results.

A. Single-hop Delay Distribution

We first evaluate the single-hop delay distribution of the TinyOS CSMA/CA protocol according to the derivations in Section IV and Section V. 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 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. 4(a) and Fig. 4(d) for pmf and cdf of delay, respectively. For all cases, the analytical model agrees well with the empirical evaluations. 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.

B. Multi-hop Delay Distribution

To validate the model for multi-hop networks and illustrate the effects of network parameters and heterogeneity in WSNs, two sets of experiment have been performed. First, a network consisting of 25 TelosB nodes are used. The nodes are placed in a 5×5 grid, as illustrated in Figure 4(b). Nodes shown as light-colored boxes only relay packets while the 8 dark-colored 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, λ_l, the queue length, M, and the maximum number of transmission attempts, N_{tx} are varied to reveal the relationships between each of the parameters and the end-to-end delay distribution. Moreover, the packet error rate as well as the contention relationship between each pair of nodes are measured and hidden terminals are then identified. End-to-end delay is measured for approximately $3,000$ packets for each configuration.

The results are shown in Figure 4(e). As can be observed, the *cdf* of the analytical model match well with the empirical results. The slight difference in these results is due to the inaccurate collision models. The collision range in practice is not an arbitrary area for each node and a transitional area exists around the boundary [31]. 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, N_{tx}, the reliability decreases. However, when a low delivery rate (e.g., less than 50%) is sufficient, a lower M or N_{tx} 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 located in three rooms is used as shown in Figure 4(c). Two different network configurations are used to illustrated the effects of heterogeneity. 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. 4(c). 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 Figure 4(c) as solid lines. The *cdf*s of the results are shown in Fig. 4(f). Accordingly, increasing transmit power in only two nodes significantly impacts the end-to-end delay as also captured by the analytical evaluations, which is important for the design of heterogeneous WSNs.

VIII. Conclusion and Future Work

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 transmission 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.

As future work, we plan to extend the model to capture other aspects of heterogeneity in WSNs, such as networks consisting of subnetworks using different protocols and heterogeneous routing approaches. Moreover, models for other MAC protocols, for example, TDMA protocols and MAC protocol with Low Power Listening, such as BMAC [33], will be derived through our generic framework. More traffic patterns including CBR traffic and bursty will also be considered. Our ongoing work also suggests that the developed cross-layer framework can be extended to capture the energy consumption distribution.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar 2002.

[2] 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.

[3] N. Bisnik and A. Abouzeid, "Queuing network models for delay analysis of multihop wireless ad hoc networks," in *IWCMC '06*. New York, NY, USA: ACM, 2006, pp. 773–778.

[4] G. R. Gupta and N. B. Shroff, "Delay analysis for multi-hop wireless networks," in *Proc. IEEE INFOCOM 2009*, Rio de Janeiro, Brazil, Apr. 2009, pp. 412–421.

[5] T. Abdelzaher, S. Prabh, and R. Kiran, "On real-time capacity limits of multihop wireless sensor networks," in *Proc. IEEE RTSS '04*, Dec. 2004, pp. 359–370.

[6] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A stateless protocol for real-time communication in sensor networks," in *Proc. IEEE ICDCS '03*, Los Alamitos, CA, May 2003.

[7] C. Lu, B. Blum, T. Abdelzaher, J. Stankovic, and T. He, "RAP: a real-time communication architecture for large-scale wireless sensor networks," in *Proc. IEEE RTAS '02*, Sept 2002, pp. 55–66.

[8] 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.

[9] M. Fidler, "An end-to-end probabilistic network calculus with moment generating functions," in *Proc. IEEE IWQoS '06*, June 2006, pp. 261–270.

[10] 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, Sept. 2006.

[11] J. Lehoczky, "Real-time queueing network theory," in *Proc. IEEE RTSS '97*, Dec 1997, pp. 58–67.

[12] S.-N. Yeung and J. Lehoczky, "End-to-end delay analysis for real-time networks," in *Proc. IEEE RTSS '01*, Dec. 2001, pp. 299–309.

[13] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE JSAC*, vol. 18, no. 3, pp. 535–547, Mar 2000.

[14] 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, Sept. 2005.

[15] 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. IEEE WCNC '08*, Mar 2008, pp. 1559–1564.

[16] 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.

[17] 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. on Networking*, vol. 16, no. 4, pp. 878–891, Aug 2008.

[18] A. Koubaa, M. Alves, and E. Tovar, "Modeling and worst-case dimensioning of cluster-tree wireless sensor networks," in *Proc. IEEE RTSS '06*, Dec. 2006, pp. 412–421.

[19] J. Schmitt, F. Zdarsky, and L. Thiele, "A comprehensive worst-case calculus for wireless sensor networks with in-network processing," in *Proc. IEEE RTSS '07*, Dec. 2007, pp. 193–202.

[20] H. Li, P. Shenoy, and K. Ramamritham, "Scheduling messages with deadlines in multi-hop real-time sensor networks," in *Proc. IEEE RTAS '05*, March 2005, pp. 415–425.

[21] "IEEE Std 802.11: IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications," Jun. 2007.

[22] 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. IEEE GLOBECOM '04*, vol. 5, Dec. 2004, pp. 3022–3026.

[23] 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, 2009.

[24] 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. IEEE INFOCOM '05*, vol. 4, Mar 2005, pp. 2646–2657.

[25] K. Gopalan, T. cker Chiueh, and Y.-J. Lin, "Probabilistic delay guarantees using delay distribution measurement," in *Proc. ACM MULTIMEDIA '04*, New York, NY, Oct 2004, pp. 900–907.

[26] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: an Algorithmic Approach*. Dover Publications Inc., 1981.

[27] R. Nelson, *Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling*. New York, NY: Springer-Verlag, 1995.

[28] M. Zorzi and R. Rao, "Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance," *IEEE Trans. on Mobile Computing*, vol. 2, no. 4, pp. 337–348, Oct.-Dec. 2003.

[29] "Tinyos," web, http://webs.cs.berkeley.edu/tos/.

[30] "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.

[31] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *Proc. IEEE SECON '04*, Oct. 2004, pp. 517–526.

[32] M. Vuran and I. Akyildiz, "Cross-layer analysis of error control in wireless sensor networks," in *Proc. IEEE SECON '06*, vol. 2, Sept. 2006, pp. 585–594.

[33] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM SenSys '04*. New York, NY, USA: ACM, Nov 2004, pp. 95–107.