

Chapter 2

Stochastic Modeling of Delay, Energy Consumption, and Lifetime

Yunbo Wang, Mehmet C. Vuran and 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 state, probabilistic analysis of QoS is essential. For most WSNs applications, the end-to-end delay for packet delivery and the energy consumption are the most important QoS metrics. In this chapter, a comprehensive cross-layer probabilistic analysis framework is presented to investigate the probabilistic evaluation of QoS performance provided by WSNs. In particular, the QoS performance is evaluated in two levels. In the node level, using a Discrete-Time Markov queueing model, the distribution of single-hop delay and single-node energy consumption and lifetime are analyzed. In the network level, based on the node level analysis, the distributions of end-to-end delay, the network lifetime, and the event detection delay are then analyzed. Fluid models are utilized in the network level analysis. The framework also considers a realistic channel environments. Compared to the first-order QoS statistics, such as the mean and the variance, the distribution of QoS metrics reveals the relationship between the performance and reliability with QoS-based operations in WSNs. Using the framework, effective network development can be performed.

Y. Wang (✉) · M. C. Vuran · S. Goddard
University of Nebraska-Lincoln, Lincoln, NE, USA
e-mail: yunbow@cse.unl.edu

M. C. Vuran
e-mail: mcvuran@cse.unl.edu

S. Goddard
e-mail: goddard@cse.unl.edu

1 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 [4]. Increasingly, a large number of WSN applications are investigated with various quality requirements for different network services specific to low-cost hardware, and unpredictable environment conditions [3, 12]. These requirements necessitate a comprehensive analysis of the Quality of Service (QoS) provided by the network.

QoS issues and techniques have been intensively investigated for ATM networks [14, 45], IP networks [5, 44, 45], and traditional wireless networks [11, 47]. In these studies, the evaluation of QoS is mainly focused on the communication quality characterized by communication delay, jitter, bandwidth, and loss rate. Traditional metrics, however, cannot fully characterize the QoS in WSNs [12], because of the distinct characteristics of WSN applications.

WSNs are utilized for a different set of applications from those with traditional networks [4]. These applications emphasize different characteristics of the network and require different *services* to be provided by the network. Thus, the metrics to evaluate the *quality* of these services are also different from traditional QoS metrics. For example, for most WSN applications, sensor nodes are powered by batteries with limited capacity, and replacing the batteries is difficult. Thus, the network lifetime under battery constraints is a QoS measure that is more important than in traditional network analysis. Other examples of such QoS measures include the delay for event detection, and sensing rate of individual sensors.

Due to limited resource availability, QoS analysis must be performed in a cross-layer manner. In traditional network analysis, with adequate resources assumed, the maintainability and modularity are emphasized at the expense of additionally consumed resources such as storage, computing power, and energy supply. Hence, the QoS is separately provided by different network layers. In contrast, with limited resources, WSNs are usually designed to exploit cross-layer operations and meet QoS requirements more efficiently [46]. For example, cross-layer integration can lead to significant energy conservation [65, 66]. Moreover, requirements on different QoS metrics can contradict with each other, and a tradeoff must be made to provide optimal services. For example, lower delay and longer lifetime are usually contradicting design goals. Lower delay usually requires a high duty cycle, whereas longer lifetime usually favors low duty cycle. Therefore, a QoS analysis framework that captures the tradeoffs for the protocols and operations in the entire software stack is desirable.

1.1 Why Stochastic Models?

In this chapter, we discuss stochastic analysis of QoS provision in WSNs. Stochastic models are necessary for WSNs due to three main factors that result in random operation characteristics.

- **Environmental Conditions:** WSNs encounter environment conditions that are unreliable and random in nature. Many applications in harsh environments such as wild fields and battlegrounds further impose possible physical damage to the nodes [4]. Any degradation in the performance of one of the nodes would result in unpredicted fluctuations in network performance.
- **Low-cost Hardware:** Sensor nodes are usually manufactured *en masse* with low-cost hardware. Thus, it is expected that the nodes may randomly cease to work, resulting in a random network topology.
- **Wireless Channel:** The wireless communication among nodes are also prone to random noises due to low-profile radio transceivers and limited communication power as well as wireless channel impurities such as multi-path fading and shadowing.

All these random factors result in a large variance in QoS metrics, and cannot be thoroughly evaluated using traditional approaches, such as mean delay analysis [1, 7, 28], or worst-case analysis [9, 22].

As wireless sensor network (WSN) technology has matured, more demanding applications have emerged. These applications require quality of service (QoS) guarantees with respect to end-to-end delay, energy consumption, lifetime, and throughput, with high confidence. One example is a smart space application, where resource-constrained wireless sensors and actuators provide situational awareness and assistance to disabled or elderly people. Therein, a random and a rare event, such as a person falling, should be detected within a bounded delay, while still maintaining acceptable throughput and energy consumption levels.

Designing such systems, however, is challenging due to their stochastic characteristics. The common practice of optimizing QoS metrics based on first-order statistics (e.g., mean) is convenient but may be insufficient—especially when the cumulative distribution function (*cdf*) of the QoS metric is not based on a Gaussian distribution. In the case of packet delay, for example, this technique results in good average-case performance, but some packets may experience delays 20 times longer than the average [69, 71]. The problem becomes significantly harder when one considers multiple QoS metrics simultaneously.

We envision that the utilization of these *stochastic analysis techniques* will transform the network design approaches from utility-based solutions to stochastic design principles. In other words, the developed tools are expected to provide the necessary *knobs* that can be tuned to satisfy system requirements and the principles on *how* to tune them.

1.2 Chapter Overview

The remainder of the chapter is organized as follows: In Sect. 2, an overview of the anycast protocol used for illustration of the models is provided. Then, in Sect. 3, stochastic analysis of delay in WSNs is discussed. Following the model described in

this section, the event detection delay distribution is analyzed in Sect. 4. Finally, the stochastic distribution of energy consumption and lifetime in WSNs is presented in Sect. 5. As a whole, this chapter provides a comprehensive tool set for the stochastic analysis of QoS provision in WSNs and presents validations of these analysis tools through simulations and testbed experiments.

2 Anycast Protocol

To illustrate the practical applicability of the analysis tools, anycast protocol is used as a case study throughout the chapter. Other protocols can easily be used following the techniques illustrated for the anycast protocol. The derivations for other protocols (e.g., TDMA, geographical routing, etc.) are left to the reader.

Recent protocol developed for WSNs employ duty cycle operations to save communication energy [8, 57]. To combat the delay incurred due to this duty cycle operation, opportunistic routing techniques, particularly anycast protocols, are utilized along with a high node density to exploit node deployment redundancy [33, 35, 41, 55, 66]. The anycast technique is a cross-layer approach that exploits both temporal and spatial efficiency.

With the anycast technique, if a node has packets to send, it first broadcasts a series of beacon messages. Then, one of the responding neighbors is chosen as the next-hop node according to predefined rules (e.g., the first node that responds, or the closest node to the destination). Finally, the sender forwards the data packet to the chosen neighbor. There are several variations of this basic anycast technique in WSNs. In this chapter, we consider the following representative protocol as an example to illustrate the stochastic analysis techniques.

The anycast protocol operation is illustrated in Fig. 1. Sensor nodes report their readings to the sink through multi-hop routes in the network. The nodes (excluding the sink) turn off their radio periodically to save energy. When a node x has a packet to send it starts to repeatedly transmit Request-to-Send (RTS) beacon packets. These packets are sent using a channel sensing mechanism before the beacon transmission. As shown in Fig. 1, when any other node x_1 in the transmission range is awake and

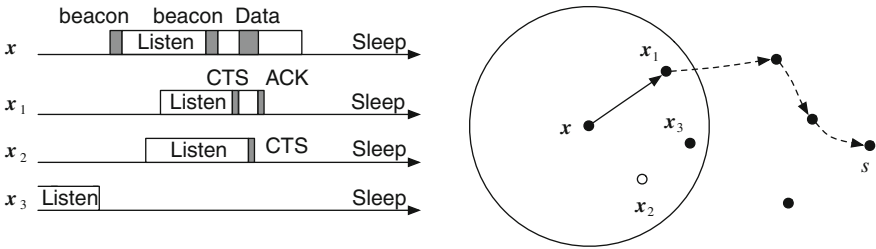


Fig. 1 The transmission process and routing path for a packet with the anycast protocol

hears the packet, it checks for the following criteria: (1) node x_1 is closer to the sink than x , (2) the signal-to-noise ratio (SNR) of the received RTS packet, ψ , is greater than some predefined threshold ψ_{th} , and (3) if the node does not have any packets to send. If all criteria are met, node x_1 sends a Clear-to-Send (CTS) packet. The same conditions may be satisfied at other nodes (e.g., node x_2), which send their CTS as well. Then, node x chooses the first node that sends 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.

3 Single-Hop and End-to-End Delay Distribution

As WSNs are increasingly used in critical applications, delay becomes one of the most important metrics in their design. Complex and cross-layer interactions in multi-hop WSNs require a complete stochastic characterization of the delay. Because of the randomness in wireless communication and the low power nature of the communication links in WSNs, average delay or worst-case delay provides limited insight into the operations of WSNs. Probabilistic analysis of delay has been performed for broadcast networks [6, 51, 58, 59, 61] considering several medium access control (MAC) protocols. Indeed, the cumulative distribution function (*cdf*) of the delay for a given deadline can be used as a probabilistic metric for reliability and timeliness. In addition to single-hop delay, additional delay due to multi-hop communication, queuing delay, and wireless channel errors have to be captured as they are imperative to completely characterize the delay distribution in WSNs.

In this section, we discuss the stochastic analysis of delay in WSNs. The resulting delay distribution is an important metric to evaluate the communication services provided by the network, since it measures the probability that the network meets a given deadline. The developed framework highlights the *relationship 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.

3.1 Background

Historically, the problem of probabilistic end-to-end delay has attracted a large interest. The concept of Network Calculus [15] can be extended to support probabilistic delay *bounds* in [9, 22, 34, 60]. 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. Accordingly, the *worst case* performance bounds can be analyzed. This approach has limited applications in WSNs due to the randomness in wireless communication,

large variance in the end-to-end delay, the fact that most applications tolerate packet loss for a lower delay of higher priority packets since the efficiency of the system is improved. Accordingly, *probabilistic delay characteristics* are of interest in addition to the worst case bounds.

Real-time theory and queueing theory are combined to provide stochastic models for unreliable networks [38, 75] for networks with heavy traffic rates. The approach in this section is similar to the real-time queueing theory [38] in that a stochastic queueing model for the analysis is used. However, we do not focus on the real-time scheduling problem [38, 40, 75] but instead characterize delay for the development of communication solutions.

Recently, the delay distribution of medium access control (MAC) protocols has been analyzed for IEEE 802.11b DCF protocol [6, 59, 61], IEEE 802.15.4 protocol [56, 58], and TDMA protocols [51], where a broadcast network is considered. Saturated traffic cases are also investigated extensively [6, 59, 61]. In addition, distribution of link layer retransmissions [31], end-to-end delay in a linear network with infinite queues [72], and delay in data aggregation networks [25] are recently modeled. These theoretical approaches are also complemented with empirical approaches that use on-the-fly measurements estimate probabilistic characteristics of end-to-end delay [21, 26, 54].

The remainder of this section is an extension to this state-of-the art to capture multi-hop communication effects, hidden node problems, and the low traffic rate of WSNs. In the following, we first provide a problem definition and then, discuss the probabilistic end-to-end delay analysis model.

3.2 Problem Definition and System Model

Let us assume a network deployment, where each node is indexed by its location \mathbf{x} . We are interested in the following two problems:

- (1) What is the probability distribution function (*pdf*) of single-hop delay, $f_{\text{sh}(\mathbf{x}, \mathbf{y})}(t)$, between two nodes \mathbf{x} and \mathbf{y} for a new arriving packet?
- (2) Given the single-hop delay distribution, what is the *pdf* of the end-to-end delay, $f_{\text{e2e}(\mathbf{x}, \mathbf{s})}(t)$, between a node \mathbf{x} and a sink located at \mathbf{s} ?

In the following, we provide an overview of solutions for the two problems above and the detailed descriptions are deferred to Sects. 3.3 and 3.4.

3.2.1 Single-Hop Delay Distribution

In this setting, we can model each node according to a queueing model, which is characterized by its inter-arrival distribution and service process. More specifically, the traffic inter-arrival can be modeled according to a Geometric distribution [68].

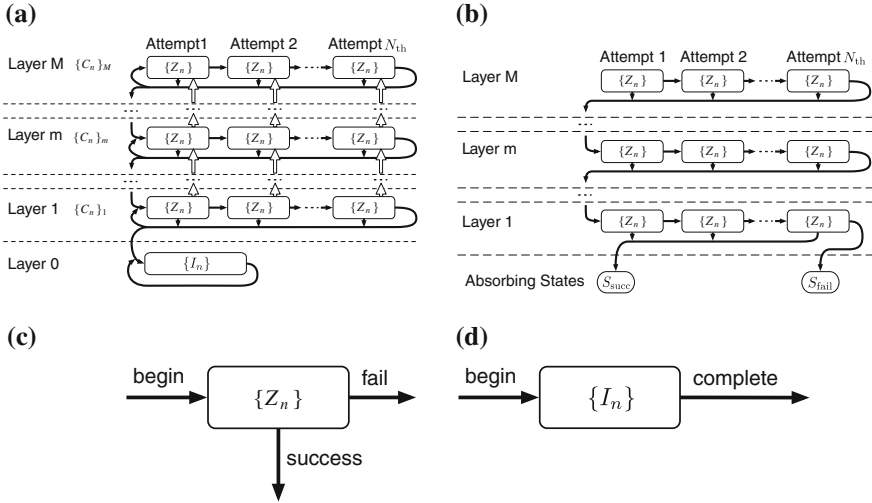


Fig. 2 The structures of Markov chains: **a** $\{X_n\}$ and **b** $\{Y_n\}$. Their building blocks are also shown: **c** $\{Z_n\}$ and **d** $\{I_n\}$

The Geometric distribution is selected as experiment results with different types of traffic (event-based and periodic) indicate that it is a valid model for WSN traffic [68]. Furthermore, a Discrete Time Markov Process (DTMP) is used to model the service behavior. In such a model, the service time is Phase-Type (PH) distributed by definition [52]. 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.

The communication system at each node is modeled as a discrete-time recurrent Markov chain, $\{X_n\}$. As shown in Fig. 2a, this discrete time Markov chain (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 steady-state parameters found in $\{X_n\}$ are used in $\{Y_n\}$ to derive the associated distributions. The detailed explanation of these DTMCs is provided in Sect. 3.3.

3.2.2 End-to-End Delay Distribution

With each hop modeled as a Geom/PH/1/ M queue, the entire network can be considered as a queueing network. Nodes are interrelated according to the traffic constraints. More specifically, the successfully transmitted traffic rate from one node should equal 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 discussion, we focus on the class of routing protocols wherein each node maintains

a probabilistic routing table for its neighbors, e.g., anycast routing protocols [2]. By first calculating the relay traffic and the single hop delay distribution for each pair of nodes, the end-to-end delay can be obtained using an iterative procedure as explained in Sect. 3.4.

3.3 Single-Hop Delay Distribution

For single-hop delay distribution, we are interested in capturing the delay in transmitting a packet from one node to another. This delay includes the queueing delay at the transmitting node and the communication delay due to transmission time, medium access, and wireless channel errors and retransmissions. The communication system at each node can be modeled by a DTMC $\{X_n\}$ and its absorbing variant $\{Y_n\}$. The first DTMC, $\{X_n\}$, captures the equilibrium behavior of the system. The second DTMC, $\{Y_n\}$, is then used to analyze the transient communication behavior after a specific packet arrives. The single-hop delay of the packet transmission is then represented as the absorption time of $\{Y_n\}$. In the following, we will discuss the construction of $\{X_n\}$ and $\{Y_n\}$ and show how the single-hop delay distribution is derived.

3.3.1 Steady-State Analysis

The DTMC, $\{X_n\}$, as shown in Fig. 2a, is two dimensional, where the horizontal dimension captures the transmission attempts and the vertical dimension captures the queue state. Accordingly, $\{X_n\}$ is composed of $M + 1$ layers, where each layer m ($0 \leq m \leq M$) represents the number of packets in the queue and M is the queue capacity. The first layer, the quiescent layer, $\{I_n\}$, ($m = 0$) represents the *quiescent 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 more transmission attempts are conducted, until either the packet is successfully transmitted, or the maximum number of transmission attempts, N_{tx} , is exceeded. In the latter case, the packet is dropped.

A layer m in $\{X_n\}$ is denoted by $\{C_n\}_m$, and is composed of N_{tx} blocks. The b th block in layer m is denoted by $\{Z_n\}_{m,b}$.¹ As shown in Fig. 2c, 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 N_{tx} transmission attempts fail. Consequently, the v th state in layer m and transmission attempt b is denoted by $S_{m,b,v}$.

In a typical network operation, each node is exposed to two types of incoming traffic: locally generated and relay traffic. While the locally generated traffic can arrive at any time and depends on the application, the relay traffic can only arrive

¹ In the following, we drop the indices m and b , where appropriate, to simplify the notation.

when the node is listening. Therefore, the total traffic rate depends on the state of the process. Let us denote the locally generated traffic rate and the relay traffic rate for a node by λ_{lc} and λ_{re} , respectively. Then, in the states where the node is listening, the total traffic rate is $\lambda_{lc} + \lambda_{re}$. When the radio is in sleep, the total traffic rate is only λ_{lc} .

The quiescent and the communication layers, $\{I_n\}$ and $\{C_n\}$, are parameterized by the following notations:

- P_I and P_C : the transition probability matrices among the states in $\{I_n\}$ and $\{C_n\}$, respectively.
- α_I and α_C : the initial probability vector for $\{I_n\}$ and $\{C_n\}$, respectively.
- t_I^s and t_C^s : the probability vector from each state in $\{I_n\}$ and $\{C_n\}$ to complete the quiescent process and the communication process successfully, respectively.
- t_C^f : the probability vector from each state in $\{C_n\}$ to complete the communication process unsuccessfully.
- λ_I and λ_C : the packet arrival probability vector for each state in $\{I_n\}$ and $\{C_n\}$, respectively. Each element in the vectors equals to the probability of a new packet arrival in a time unit when the process is in the corresponding state.

The Markov chain block for each transmission attempt, $\{Z_n\}$, is characterized by the following:

- P_Z , the transition probability matrix among the states in $\{Z_n\}$,
- α_Z , the initial probability vector for $\{Z_n\}$, and
- t_Z^s and 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, let us assume that these matrices are known and leave the discussion on how to obtain them to the case study in Sect. 3.5. Then, the transition probability matrix among the states in a single layer $\{C_n\}$ in $\{X_n\}$ is

$$P_C = \begin{bmatrix} P_Z t_Z^f \alpha_Z & & \mathbf{0} \\ & \ddots & \ddots \\ & & P_Z t_Z^f \alpha_Z \\ \mathbf{0} & & & P_Z \end{bmatrix}, \quad (1)$$

where the number of P_Z blocks in P_C is equal to N_{tx} , i.e., the maximum number of attempts for each packet transmission. Similarly, the initial probability vector, α_C , and the probability vectors, t_C^s and t_C^f to complete a layer in success and failure are

$$\alpha_C = [\alpha_Z \mathbf{0} \cdots \mathbf{0}] \quad (2)$$

$$t_C^s = [t_Z^s t_Z^s \cdots t_Z^s]^T \quad (3)$$

$$t_C^f = [\mathbf{0} \mathbf{0} \cdots t_Z^f]^T \quad (4)$$

respectively.

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 [68, 71] as follows:

$$\mathbf{Q}_X = \begin{matrix} \text{layer} & 0 & 1 & 2 & \cdots & M \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \cdots \\ M \end{matrix} & \begin{pmatrix} \mathbf{A}_{s0} & \mathbf{A}_{u0} & & & \mathbf{0} \\ \mathbf{A}_{d0} & \mathbf{A}_s & \mathbf{A}_u & & \\ & \mathbf{A}_d & \ddots & \ddots & \\ & & \ddots & \mathbf{A}_s & \mathbf{A}_u \\ \mathbf{0} & & & \mathbf{A}_d & \mathbf{A}_s + \mathbf{A}_u \end{pmatrix} \end{matrix}, \quad (5)$$

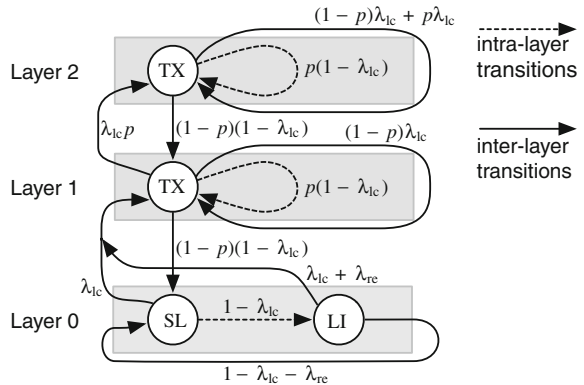
where each nonzero block corresponds to the transition probability among all layers. The duration of the time unit T_u is chosen to be small enough such that the probability of having two or more transitions in a single time unit is negligible. Therefore, it is only possible for $\{X_n\}$ to have intra-layer transitions and inter-layer transitions to adjacent layers.

The first row and column of blocks in \mathbf{Q}_X corresponds to the transition probabilities from and to the quiescent layer, respectively. Then, the equilibrium state probability vector, $\boldsymbol{\pi}$, for $\{X_n\}$ is calculated by solving $\boldsymbol{\pi} \mathbf{Q}_X = \boldsymbol{\pi}$ and $\sum_i \pi_i = 1$. We will illustrate this process with a basic example next.

Example 1. Let us now consider an example protocol, where a node conducts a duty cycle operation every 2 s. It first sleeps for 1 s and then listens to the channel for another 1 s. If a packet is received during the listening period with a probability λ_{lc} , or if a local packet is generated in any period with a probability of λ_{lc} , the node attempts to transmit the packet. The transmission attempt takes 1 s with a failure rate p , and the node persistently attempts to transmit the packet until successful. While transmitting, the node cannot receive any packets, but can still generate packets. The queue length is $M = 2$.

For this protocol, a time unit of 1 s can be chosen since all time periods are 1 s. Then, the quiescent process can be modeled by two states, and the communication process can be modeled by one state, as shown in Fig. 3. The quiescent process,

Fig. 3 The structure of $\{X_n\}$ for the example



$\{I_n\}$, contains a sleeping state (SL) and a listening state (LI), whereas the communication process, $\{C_n\}$, contains a single transmission state (TX). Accordingly, $P_I, P_C, \alpha_I, \alpha_C, t_I^s, t_C^s, t_C^f, \lambda_I$ and λ_C are found as:

$$\begin{aligned} P_I &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, & P_C &= p, \\ \alpha_I &= \begin{bmatrix} 1 & 0 \end{bmatrix}, & \alpha_C &= 1, \\ t_I^s &= \begin{bmatrix} 0 & 1 \end{bmatrix}^T, & t_C^s &= 1 - p, & t_C^f &= 0, \\ \lambda_I &= [\lambda_{lc} \ \lambda_{lc} + \lambda_{re}], & \lambda_C &= \lambda_{lc}, \end{aligned} \quad (6)$$

where $t_C^f = 0$ because the communication persistently attempts to transmit until successful, thus it can never fail. Therefore, the blocks in \mathbf{Q}_X (see (5)) are expressed as

$$\begin{aligned} A_u &= \lambda_{lc} p, & A_{u0} &= [\lambda_{lc} \ \lambda_{lc} + \lambda_{re}]^T \\ A_s &= \lambda_{lc}(1 - p) + (1 - \lambda_{lc})p, & A_{s0} &= \begin{bmatrix} 0 & 1 - \lambda_{lc} \\ 1 - \lambda_{lc} - \lambda_{re} & 0 \end{bmatrix} \\ A_d &= (1 - \lambda_{lc})(1 - p), & A_{d0} &= [(1 - \lambda_{lc})(1 - p) \ 0] \end{aligned} \quad (7)$$

3.3.2 Transient Analysis

Given the steady state characteristics of the system, we can now find the distribution of single-hop delay for a packet by an absorbing DTMC. 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 in the system. To derive the delay distribution, we use an absorbing variant of $\{X_n\}$ that is denoted as $\{Y_n\}$ as shown in Fig. 2b. In $\{Y_n\}$, the quiescent 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 in a first in first out (FIFO) queue. 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 as follows.

Before the packet arrives, the system is in one of the states according to the equilibrium state probability vector, π . After the new packet arrives, if the queue is full, the packet is immediately dropped. The probability of queue full is

$$p_{qf} = \pi_M A_u \mathbf{1}, \quad (8)$$

where π_M is the sub-vector in π 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 probabil-

ity matrix of $\{Y_n\}$ conditioned on the fact that the new packet arrives. \mathbf{Q}_Y^{up} is derived from \mathbf{Q}_X in (5) by replacing λ_I and λ_C with vectors of all 1's 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, $\boldsymbol{\pi}'$ is the initial probability vector for $\{Y_n\}$.

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

$$\mathbf{Q}_Y = \begin{bmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ \mathbf{t}_Y^s & \mathbf{t}_Y^f & \mathbf{P}_Y \end{bmatrix}, \quad (9)$$

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 transition probability matrix among the transient states, i.e., all states except S_{succ} and S_{fail} , is given by

$$\mathbf{P}_Y = \begin{bmatrix} \mathbf{P}_C & & \mathbf{0} \\ \mathbf{t}_C \boldsymbol{\alpha}_C & \mathbf{P}_C & \\ & \ddots & \ddots \\ \mathbf{0} & & \mathbf{t}_C \boldsymbol{\alpha}_C & \mathbf{P}_C \end{bmatrix}. \quad (10)$$

This is obtained from (5) by removing the first row and first column of blocks, and replacing λ_I and λ_C with vectors of all 0's for each remaining block. The transition probability vectors from each of the transient states to the absorbing states are

$$\mathbf{t}_Y^s = [\mathbf{t}_C^s \ \mathbf{0} \ \mathbf{0} \ \cdots]^T, \mathbf{t}_Y^f = [\mathbf{t}_C^f \ \mathbf{0} \ \mathbf{0} \ \cdots]^T, \quad (11)$$

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

Theorem 1. *The probability mass function (pmf) of the number of time units, K , a packet should wait before being transmitted and dropped are*

$$f_K^s(k) = \boldsymbol{\alpha}_Y \mathbf{P}_Y^{k-1} \mathbf{t}_Y^s, f_K^f(k) = \boldsymbol{\alpha}_Y \mathbf{P}_Y^{k-1} \mathbf{t}_Y^f, \quad (12)$$

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

Proof. The theorem follows from [50, 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*

$$f_K(k) = \alpha_Y \mathbf{P}_Y^{k-1} \mathbf{t}_Y. \quad (13)$$

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*

$$p_{\text{deli}} = \sum_{k=1}^{+\infty} f_K^s(k) = \alpha_Y (\mathbf{I} - \mathbf{P}_Y)^{-1} \mathbf{t}_Y^s. \quad (14)$$

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

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

$$\mu_K = \frac{\alpha_Y (\mathbf{I} - \mathbf{P}_Y)^{-2} \mathbf{t}_Y^s}{p_{\text{deli}}}, \quad (15)$$

$$\sigma_K^2 = \frac{\alpha_Y (2(\mathbf{I} - \mathbf{P}_Y)^{-3} - (\mathbf{I} - \mathbf{P}_Y)^{-2}) \mathbf{t}_Y^s}{p_{\text{deli}}} - \mu_K^2, \quad (16)$$

respectively.

The derivations are straightforward since

$$\mu_K = E[K] = \frac{\sum_{k=1}^{+\infty} k \cdot f_K^s(k)}{\sum_{k=1}^{+\infty} f_K^s(k)}, \quad (17)$$

$$\sigma_K^2 = E[K^2] - (E[K])^2 = \frac{\sum_{k=1}^{+\infty} k^2 \cdot f_K^s(k)}{\sum_{k=1}^{+\infty} f_K^s(k)} - \mu_K^2, \quad (18)$$

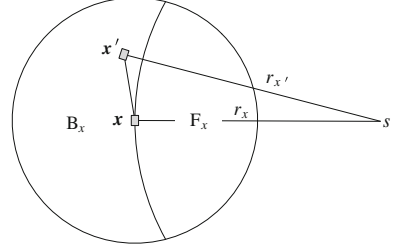
where $E[\cdot]$ represents expectation.

Based on the single-hop delay distribution, we will next derive the end-to-end delay distribution in WSNs.

3.4 End-to-End Delay Distribution

In a multi-hop WSN, the delay is generally calculated as the time it takes for a packet generated in the network to reach the sink. This delay, accordingly, depends on the route taken by the packet. Thus, the topology of the network and the routing protocol defines the end-to-end delay performance. Next, we discuss the case of random deployments.

Fig. 4 The feasible region, \mathbb{F}_x , and the infeasible region, \mathbb{B}_x , of node x



For the random deployment, the nodes are assumed to be located in the network according to a Poisson point process with density ρ . For these types of networks, geographic routing protocols [2] 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 $p_{i,j}^{\text{fw}}$ between any pair of nodes i and j , the routing probability between any pair of *locations* x and y , $p_{x,y}^{\text{fw}}$ is of interest.

A common scenario is considered, where the nodes in the network generate the same amount of local traffic to a sink. Moreover, each node x forwards packets to the neighboring nodes within its *feasible region* \mathbb{F}_x , i.e., the region in which nodes are closer to the sink, but are still in the transmission range. 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 λ_{lc} . By symmetry, the relay traffic $\lambda_{\text{re},r}$ is the same for all nodes in the same ring r . We assume a polar coordinates system with the sink located at the origin.

As shown in Fig. 4, for a node x located at $x = (r_x, \theta_x)$, the relay traffic arrives from any node y in the *infeasible region* $\mathbb{B}_x = \mathbb{C}_x \setminus \mathbb{F}_x$, i.e., the region in which nodes are farther to the sink but are still in the transmission range. To derive the relay traffic rate for x and other nodes in ring r_x , consider the small area $(r_x : r_x + \Delta r, \theta : \theta + \Delta \theta)$ around node x located at (r_x, θ) . Similar to the deterministic deployment, the relay traffic rate λ_{re,r_x} is given by

$$\begin{aligned} \lambda_{\text{re},r_x} &= \bar{\lambda}_{\text{re},r_x} / \pi r_x^{\text{listen}}, \\ \bar{\lambda}_{\text{re},r_x} &= \frac{\int_{\mathbb{B}_x} \rho (\bar{\lambda}_y^{\text{f}} + \lambda_{\text{lc}}) p_{y,x}^{\text{fw}} p_{\text{deli},y,x} d\mathbf{y}}{\rho \Delta r \Delta \theta r_x}, \end{aligned} \quad (19)$$

where ρ is the network density of the Poisson node distribution, $p_{y,x}^{\text{fw}}$ and $p_{\text{deli},y,x}$ are similarly defined as $p_{m,i}^{\text{fw}}$ and $p_{\text{deli},m,i}$, except that the nodes are indexed by their locations.

Finally, $p_{y,x}^{\text{fw}}$ in (19) is the routing protocol-specific probability that the node at y transmit packets to a node at x . A case study for the anycast protocol will be provided in Sect. 3.5 to show how this probability is obtained.

Thus, according to (19), the traffic rate of node x at each state can be determined. Accordingly, the input traffic rate vectors λ_I and λ_C of node x can be found according to Sect. 3.3. Then, the equilibrium state probability for the DTMC $\{X_n\}$, π_{r_x} is obtained. Note that in (19), the traffic rate for nodes in ring r_x 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 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_x to the sink, and is obtained by

$$f_{e2e(r_x)}(t) = \int_{\mathbb{F}_x} p_{x,y}^{\text{fw}} f_{\text{sh}(r_x)} * f_{e2e(r_y)}(t) dy. \quad (20)$$

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

Next, the anycast protocol is used as a case study to illustrate the end-to-end delay analysis in a randomly deployed network.

3.5 Case Study: Anycast Protocol

Most WSN communication protocols employ a duty cycle mechanism to save energy. Here, we consider an anycast protocol, which aims to find next hops on the fly to forward packets to the sink. Accordingly, we show how the single-hop and the end-to-end delay analysis in Sects. 3.3 and 3.4 can be applied to protocols with *duty cycle* operations for a randomly deployed network.

We first show the DTMC $\{X_n\}$ for the protocol. Then, the protocol-specific parameters for the generic analysis in Sect. 3.3, 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.

3.5.1 Markov Process Overview

The structures of $\{I_n\}$ and $\{C_n\}$ in DTMC $\{X_n\}$ for this protocol are shown in Fig. 5. The quiescent layer $\{I_n\}$ consists of a chain of sleeping states and a chain of listening states of duration T_u . One may think that a single sleeping state and a single listening state are enough to model the duty cycle operation, similar to the example. However, because of the memoryless nature of Markov process, arbitrary values of duty cycle must be captured with a specific number of states representing the active period and sleeping period.

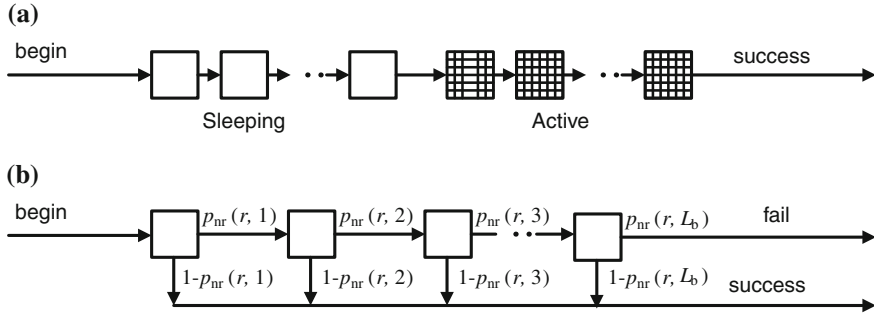


Fig. 5 The Markov chain structure of (a) the quiescent process, $\{I_n\}$, and (b) the communication process, $\{C_n\}$, for the anycast protocol

When there is no communication, the Markov process transitions through sleeping states and listening states periodically, representing the duty cycle operation. In the listening states, the node listens to the channel. Thus, both locally generated packets and relay packets can arrive. In the sleeping states, the node turns off its transceiver and only local packets can arrive. The number of states in $\{I_n\}$ is $L_c = T_{sl}/T_u + T_a/T_u = T_p/T_u$, where T_u is the time unit, and T_p is the duration of a duty-cycle period. The values of T_{sl} and T_a depend on the protocol parameters. A large T_u can reduce the number of states in the DTMC, thus, reducing complexity of the model, but at the cost of reducing the granularity and accuracy of the result.

When a packet arrives, the node terminates the quiescent process and begins the first layer of communication process $\{C_n\}$. In each $\{C_n\}$ layer, 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 time-out.

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 response, 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 transmitted successfully in limited number of (re)transmission attempts, which takes negligible time compared to the sleeping cycle T_p (usually longer than 10 s). Thus, $\{C_n\}$ only contains transmission states. In cases where retransmissions take longer durations, $\{C_n\}$ can be extended with additional 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 the cycle T_p . This is to ensure that each neighbor node can receive the beacon messages within their duty cycle period. The entire beacon communication process before packet delivery or timeout is regarded as a single transmission attempt. Thus, each communication layer $\{C_n\}$ contains only one block of $\{Z_n\}$.

3.5.2 Steady State and Transient Analyses

The transition probability matrices in $\{I_n\}$ and $\{C_n\}$, are obtained according to the Markov structure in Fig. 5. In either $\{I_n\}$ or $\{C_n\}$, there is only one initial state (denoted by “begin”) with probability of 1. States with outgoing transitions denoted by “success” or “fail” have a probability to complete the current process in a success or failure, respectively. The transition probabilities among states are shown in Fig. 5. Note that transitions with a probability of 1 are not labeled. The transition probabilities $p_{nr}(r, v)$, ($1 \leq v \leq L_b$), and the traffic rate λ_I , λ_C are explained in the following.

In the j th time unit in $\{C_n\}$, a node located at \mathbf{x} in ring r has a probability of $p_{nr}(r, v)$ of not receiving any CTS response, and enters the next state. If in all L_b states, the node 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 $p_{nr}(r, v)$ is the conditional probability that given the transmissions in the previous $v - 1$ states fails, the transmissions in the v -th state still fails. For simplicity, the hidden terminals are ignored. Therefore,

$$\begin{aligned} p_{nr}(r, 1) &= p_{nr}(r, 1 \sim 1) \\ p_{nr}(r, v) &= p_{nr}(r, 1 \sim v) / p_{nr}(r, 1 \sim v - 1), \quad 2 \leq v \leq L_b \end{aligned} \quad (21)$$

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

$$p_{nr}(r, 1 \sim v) = \prod_{\mathbf{y}=(r_y, \theta_y) \in \mathbb{F}(\mathbf{x})} (1 - p_{ex}(r_y) p_{ol}(r_y, v) p_{SNR}(\mathbf{x}, \mathbf{y})), \quad (22)$$

where each of the small areas at \mathbf{y} is located within the transmission range of \mathbf{x} , $\mathbb{C}(\mathbf{x})$, and is closer to the sink than \mathbf{x} (this range is called the *feasible region of \mathbf{x}* , $\mathbb{F}(\mathbf{x})$, as shown in Fig. 6a); r_y is the distance from the small area to the sink; $p_{ex}(r_y)$ is the probability that there exists a node in each area, and is given by

$$p_{ex}(r) = \rho \Delta r \Delta \theta r, \quad (23)$$

where ρ is the node density. Moreover, $p_{ol}(r_y, v)$ in (22) is the probability that the active period of a node located r_y away from the sink overlaps with the first j beacon transmission time units of the node at \mathbf{x} ; and $p_{SNR}(\mathbf{x}, \mathbf{y})$ is the probability that a packet, transmitted from a node at \mathbf{x} to a node at \mathbf{y} , has an SNR higher than some predefined threshold ψ_{th} . It is obtained by (10) in [76].

Next, let us derive the probability that the active period of a node at \mathbf{y} overlaps with the first v beacon transmission time units of a node at \mathbf{x} , $p_{ol}(r_y, v)$. If node \mathbf{x} receives no response in each of the small areas, at least one of the following statements is true: (1) a node does not exist in the area, (2) at least one node exists but they are sleeping

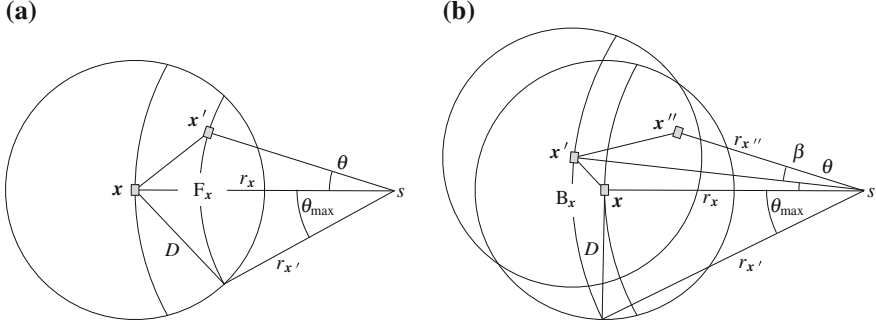


Fig. 6 The feasible region and infeasible region around node x , divided into small areas (a) Node y is in the feasible region of x (b) Node y is in the infeasible region of x , and node z is in the feasible region of y

during any of the first v slots, and (3) at least one node exists and is awake, but the SNR of the beacon packet they receive is lower than the predefined threshold ψ_{th} . Node y is awake during any of the first v slots means that the first beacon transmission time unit of node x either coincides with any of the awake time units of node y or coincides with the last $v - 1$ sleeping units of node y . Thus, $p_{ol}(r_y, v)$ is given by

$$p_{ol}(r_y, v) = \sum_{k=1}^{L_w} \pi_{W_k}(r_y) + \begin{cases} \sum_{k=L_{sl}-v+1}^{L_{sl}} \pi_s(r_y), & 1 \leq v < L_{sl} \\ \sum_{k=1}^{L_{sl}} \pi_s(r_y), & v \geq L_{sl} \end{cases} \quad (24)$$

where L_{sl} is the number of sleeping time units in $\{I_n\}$, $\pi_{W_k}(r_y)$ and $\pi_s(r_y)$ are the equilibrium probability that node y is in the k th awake state or sleeping state in $\{X_n\}$, respectively. L_c and L_w are the number of total and awake states in $\{I_n\}$, respectively.

Therefore, $p_{nr}(r, 1 \sim v)$ in (22) is determined using (23) and (24), and $p_{nr}(r, v)$ in (21) is obtained using (22).

Now, we focus on the traffic rates at each state, λ_I and λ_C . In sleeping states and listening states, the traffic arrival rate is λ_{lc} and $\lambda_{re}(r) + \lambda_{lc}$, respectively. In beacon transmission states, since nodes are assumed not to respond to any relay packets, the traffic rate is λ_{lc} .

If we consider the small area $(r : r + \Delta r, \theta : \theta + \Delta \theta)$, the forwarded traffic arrives from any node $y = (r_y, \theta_y)$ in the *infeasible region* $\mathbb{B}(x) = \mathbb{C}(x) \setminus \mathbb{F}(x)$, as shown in Fig. 6b. Therefore $\lambda_{re}(r)$ is given by

$$\lambda_{re}(r) = \frac{\sum_{y \in \mathbb{B}(x)} p_{ex}(r_y) \lambda_o(r_y) p_{fw}(y, x)}{p_{ex}(r) \pi_{li}(r)}, \quad (25)$$

where $\lambda_o(r_y)$ is the output traffic transmitted from y . $\pi_{li}(r)$ is the probability that node x is in any listening state, and is the sum of the probabilities corresponding to all listening states in $\pi(r)$. Moreover, $\lambda_o(r_y)$ is calculated by

$$\lambda_o(r_y) = \lambda(r_y)(\pi(r_y))^T(1 - p_{qfull}(r_y) - p_{drop}(r_y)), \quad (26)$$

where $p_{fw}(y, x)$ is the probability that a node y forwards a packet to node x , among all possible forward targets, and $\lambda(r_y)$ is the traffic rate vector for all states in $\{X_n\}$ for y . The probability that the packet is dropped due to beacon transmission timeout, $p_{drop}(r_y)$, is easily obtained as $p_{drop}(r_y) = p_{nr}(r, 1 \sim L_b)$ (see (22)). The probability that the queue is full when the packet arrives, $p_{qfull}(r_y)$, is obtained by $p_{qfull}(r_y) = \pi_M(r_y)A_u\mathbf{1}$, where $\pi_M(r_y)$ is the probability vector corresponding to the layer M in $\pi(r_y)$. In (25), $p_{fw}(y, x)$ is proportional to the probability that node x is *available* when y transmits a beacon, and is normalized on the total probability of availability for all possible nodes. The probability of availability is given by

$$p_{avail}(y, x) = p_{ex}(r)p_{wake}(r)p_{SNR}(y, x), \quad (27)$$

where $p_{wake}(r) = \sum_{j=1}^{L_w} \pi_{w_j}(r)$ is the probability that node x is awake, and $\pi_{w_j}(r)$ is the equilibrium probability that node x is in the j th active state in $\{X_n\}$. Then, $p_{fw}(y, x)$ in (25) is calculated as

$$p_{fw}(y, x) = \frac{p_{avail}(y, x)}{\sum_{z \in \mathbb{F}(y)} p_{avail}(y, z)}, \quad (28)$$

where node z , with the polar coordinates (r_z, β) , can be in any small area in $\mathbb{F}(y)$.

Thus, according to (25), the traffic rate of node x at each state is determined. Accordingly, $\{I_n\}$ and $\{C_n\}$ are characterized by:

- The (v, v') th element in P_I and P_C is the transition probability from state v to v' shown in Fig. 5.
- The element in α_I and α_C is 1 for states denoted 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 denoted by “success” and “fail”, respectively.
- The elements in λ_I that correspond to the sleeping states, and the elements in λ_C are set to λ_{lc} . Other elements in λ_I are set to $\lambda_{lc} + \lambda_{re}(r)$.

Then, the equilibrium state probability vector, $\pi(r)$, for the DTMC $\{X_n\}$ is obtained for each node x . Consequently, the single-hop delay distribution and end-to-end delay distribution for each ring are obtained according to (13) and (20), respectively.

Next, we focus on empirical experiments to evaluate the analysis model for anycast protocols.

3.6 Experiments

The discussed analytical model can be used to effectively study the distribution of end-to-end delay distribution. It is essential to evaluate the analysis model using empirical studies and define the limitations of the model. In the following, we provide some insight into the model through experiments. The end-to-end delay distribution model is evaluated using MATLAB to determine the single-hop and multi-hop delay distributions [68, 71]. Moreover, empirical experiments and TOSSIM-based simulations are discussed. The default radio and timing parameters of the experiments are listed in Table 1, and the parameters for the channel model are listed in Table 2.

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 Sect. 3.5 is implemented in TinyOS 2.0. The evaluation testbed consists of 25 Crossbow TelosB motes. The nodes are randomly placed in a circular area of radius $R = 4.5$ m, where the density is roughly $\rho = 0.39$. Each node generates the same amount of local traffic to be sent to the sink according to a Bernoulli process with average rate $\lambda_{lc} = 0.001$ in each time unit $T_u = 0.01$ s, which equals to 0.1 packet per second. The default duty cycle is $x = 0.5$. The simulation is performed on the same topology. Both the simulations and the testbed experiments have been run for 2.5 h and the end-to-end delay distribution for a node at distance $r = 4.3$ m is recorded, respectively. Other parameters are shown in Table 3.

When compared with the simulations and experiments, as shown in Fig. 7a, it is observed that the analytical results agree well with both results with an error of less than 10 %. With this accuracy, we can focus on simulations to analyze networks larger than 25 nodes.

Table 1 List of radio and timing parameters for TinyOS CSMA/CA protocol

Group	Notation	Description	Default value
Radio	l_p	Data packet size	40 bytes
	R_b	Channel bit rate	250 kbps
	P_t	Transmit power	−15 dBm
Timing	T_u	Time unit	320 μ s
	T_{ibo}^{max}	Maximum initial backoff	9.77 ms
	T_{cbo}^{max}	Maximum congestion backoff	2.44 ms

Table 2 List of channel-related constants and parameters

Group	Notation	Description	Default value
Channel	P_n	Noise floor	−105 dBm
	$PL(D_0)$	Path loss at reference distance	52.1 dB
	D_0	Reference distance	1 m
	η	Path loss exponent	3.3
	σ^s	Standard deviation of log-normal fading/shadowing	5.5

Table 3 List of parameters for the anycast protocol

Group	Notation	Description	Default Value
Radio	l_p	Data packet size	40 bytes
	R_b	Channel bit rate	250 kbps
	P_t	Transmit power	−15 dBm
	l_m	Beacon and CTS message size	22 bytes
Timing	T_p	Duty cycle period	1 s
	T_a	Active period	0.5 s
	T_b	Beacon transmission timeout	1 s
	T_{to}	Beacon transmission interval	12 ms
	T_u	Time unit	0.01 s
	T_{ibo}^{\max}	Maximum initial backoff	9.77 ms
	T_{cbo}^{\max}	Maximum congestion backoff	2.44 ms
Protocol	r_{th}	Threshold radius	2.7 m
	ψ_{th}	Threshold SNR	10 dBm

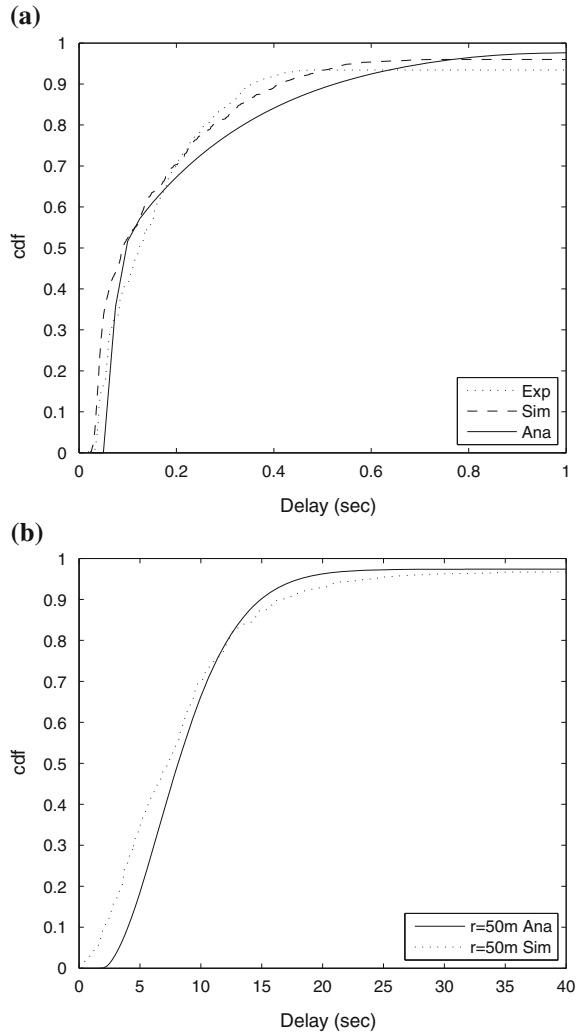
Now, we look at a larger network setting, where the network radius is set to 50m and the transmission power is increased to −10 dBm. Accordingly, the threshold distance is changed to $r_{th} = 10$ m. Moreover, the network density is $\rho = 0.1$. Durations T_p , T_a , and T_b are 10, 5 and 10 s, respectively, and the traffic rate is 0.01 pkt/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 an hour. The end-to-end delay distribution from all nodes with a distance of 50m to the sink is shown in Fig. 7b, along with the analytical results. It can be observed that the analytical result is also within an error of 10% of the simulation result.

For any network setup in the experiments, the calculation for the end-to-end delay distribution during any given duration takes less than 2 min. 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 h, the simulation takes roughly 30 min. Thus, the analytical approach provides insights significantly faster.

4 Event Detection Delay Distribution

So far, we have focused on the distribution of delay of *one packet* as it traverses in a multi-hop network. In typical event monitoring applications, however, numerous sensor nodes are deployed in the space, and operate collaboratively to monitor, report, and react to various physical events. In most WSN data monitoring applications, *events* of interest are detected by sensor nodes, and the user is interested in these events that can be detected by *multiple packets* being reported to a sink via multi-hop communication. The event detection delay consists of *discovery delay* for individual nodes to sense and detect the event, and the *delivery delay* for the network to relay

Fig. 7 **a** The analysis, simulation, and experiment results of end-to-end delay distribution with the Anycast protocol for a node with distance $r = 4.3$ m to the sink. **b** The analysis and simulation results for a distance $r = 50$ m to the sink



reports to the sink. When a given number, n , of packets are received by the sink, the event is considered to be detected. Analyzing the event detection delay is a crucial task for real-time WSN applications, which require predictable event detection delay guarantees to be provided by the network.

In this section, we discuss the distribution of event detection delay in WSNs [70]. A spatio-temporal fluid model is presented to derive the distribution of event detection delay. Accordingly, the mean event detection delay and soft-delay bounds for event detection can be modeled. We also show that motivated by the fact that queue build up in low-rate traffic is negligible, a lower-complexity model can be developed. This model extends the delay analysis for single packets, and derives the event detection delay by first obtaining the end-to-end delay for each packet.

4.1 Background

Event detection delay is associated with *flows* of packets in the network. Historically, characterizing timing performance for traffic flows has been investigated in different contexts. Several models have been developed to analyze probabilistic *bounds* on the delay of traffic flows. As an example, the concept of Network Calculus [15] is extended to derive probabilistic bounds for delay through worst case analysis [9, 22]. However, due to the randomness in and the low power nature of the communication links in WSNs, these worst case bounds cannot capture the stochastic characteristics of end-to-end delay. The communication capacity bounds for wireless networks are investigated [20, 23, 29, 43, 73] with saturated traffic flows. In WSNs, the wireless channel utilization is often well below the transmission capacity as nodes are constantly forced into a sleeping state to preserve energy.

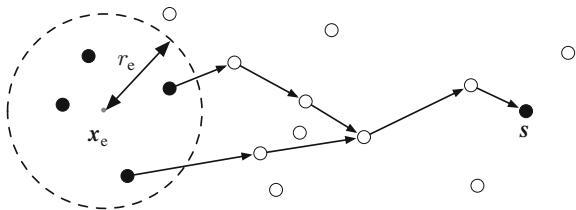
In IP network analysis, fluid-based models have been widely exploited [36, 42], and these models have recently been utilized in the analysis of WSNs [19]. Motivated by the fact that the individual packet behavior is less significant when a flow is concerned, the traffic is considered as a continuous flow instead of individual packets in these models. Accordingly, the complexity of the model can be greatly reduced. Furthermore, spatial fluid-based models have also been utilized recently to model stationary properties [13, 64], such as traffic rate and energy consumption for large-scale WSNs. These models greatly reduce the complexity of the (otherwise intractable) problem in either temporal or spatial domains. The discussion that follows builds on top these building blocks for the analysis of event detection delay.

So far, event detection delay in WSNs is analyzed in two main contexts: (1) the event discovery delay, i.e., the delay until the event is detected by an individual node [10, 17], or (2) the delivery delay in a broadcast network [27]. These models, unfortunately, cannot be easily employed for large-scale and multi-hop WSNs, where the models are intractable.

4.2 Problem Definition and System Model

Let us again assume a network deployment, where nodes are randomly located according to a Poisson point process and the node density is ρ . A sink node is deployed at location $s = (x_s, y_s)$, as shown in Fig. 8. At time $t = t_0$, a physical

Fig. 8 The network including the sink and the event generation area



event occurs at location $\mathbf{x}_e = (x_e, y_e)$, which is called the event center, and lasts for duration T_e . As shown in Fig. 8, all sensor nodes within the detection range, r_e , can discover the event. Each sensor node periodically measures the physical world every t_e seconds using its attached sensors. During the event duration $[t_0, t_0 + T_e)$, whenever the value of the measurement satisfies a predefined rule, e.g., temperature higher than a given threshold, a report packet of size L is generated and is forwarded to the sink using an anycast protocol. Due to inherent noise in the sensor readings, n ($n \geq 1$) readings from multiple sensor nodes are required at the sink to successfully detect the event occurrence. Accordingly, we define the following:

Definition 1 *An event is **n-detected** if n report packets for that event are received by the sink.*

The delay characteristics of event detection can then be modeled based on the following definitions:

Definition 2 *The **n-delay** of an event is the delay between when the physical event occurs and when the event is n -detected.*

Definition 3 *The **(p, n)-delay bound** of an event is delay within which the event is n -detected with probability p .*

To evaluate the delay characteristics of event detection in WSNs, given network and protocol parameters, n and p , we are interested in the following problems:

- What is the n -delay distribution of an event?
- What is the average n -delay of an event?
- What is the (p, n) -delay bound of an event?

In Sects. 4.3 and 4.4, we discuss spatio-temporal fluid models to address these questions.

4.3 Transient Analysis

Instead of capturing each individual node, the network can be represented by a continuous fluid entity distributed in the entire network area. Similarly, instead of individual packets, the traffic can be considered as a continuous packet fluid. By utilizing a spatio-temporal fluid model, the complexity of the problem in both spatial and temporal domains can then be reduced, and the problem becomes tractable. Testbed and simulation evaluations (Sect. 4.5) reveal that this approximation still preserves the accurately of the model.

Consider a location in the network area denoted by $\mathbf{x} = (x, y)$. We consider the nodes as a fluid entity over the entire space. Then, in an infinitesimal area around location \mathbf{x} with size $d\mathbf{x}$,² the amount of nodes is $\rho d\mathbf{x}$, where ρ is the node density.

² With a slight abuse of denotation, this infinitesimal area is henceforth denoted by $d\mathbf{x}$.

We also denote the *feasible region* of \mathbf{x} as $\mathbb{F}_{\mathbf{x}}$, and the *infeasible region* of \mathbf{x} as $\mathbb{B}_{\mathbf{x}}$. To describe the fluid traffic in the spatial fluid network, we begin by introducing the following traffic concepts:

Definition 4 *The generated, incoming, and outgoing traffic rate density for an infinitesimal area $d\mathbf{x}$ is respectively defined as the average number of packets generated, received, and transmitted by the nodes within the area, if any, in an infinitesimal duration dt , divided by the duration dt , and the size of the area $d\mathbf{x}$.*

In other words, the traffic rate densities define the speed at which packets are generated, received, and transmitted in unit space, respectively. In the transient analysis, their values change over time, and thus, are functions of t . The generated, incoming, and outgoing traffic rate density are denoted by $g_{\mathbf{x}}(t)$, $\lambda_{\mathbf{x}}(t)$, and $\omega_{\mathbf{x}}(t)$, respectively. Note that by assuming a fluid model, the *amount of nodes* in an infinitesimal area $d\mathbf{x}$, and the *amount of packets* sent in an infinitesimal duration dt , are not necessarily an integer number.

Definition 5 *The buffered traffic density for an infinitesimal area $d\mathbf{x}$ is defined as the average number of packets buffered in the queue by the nodes within the area divided by the size of the area $d\mathbf{x}$.*

The buffered traffic density is also a function of t , and is denoted by $q_{\mathbf{x}}(t)$.

In the following, we derive the set of equations that describe the fluid traffic characteristics of the network after $t = t_0$. Without loss of generality, let $t_0 = 0$. For each node, the generated traffic rate density is given by

$$g_{\mathbf{x}}(t) = \begin{cases} \frac{\rho}{t_e}, & |\mathbf{x} - \mathbf{x}_e| < r_e, \text{ and } 0 \leq t < T_e, \\ 0, & \text{otherwise,} \end{cases} \quad (29)$$

where ρ is the density, t_e is the reporting interval, and $|\mathbf{x} - \mathbf{x}_e|$ denotes the Euclidean distance between \mathbf{x} and \mathbf{x}_e . During an infinitesimal duration dt , the amount of arriving traffic, along with the traffic already stored in the queue is

$$a_{\mathbf{x}}(t) = q_{\mathbf{x}}(t) + (\lambda_{\mathbf{x}}(t) + g_{\mathbf{x}}(t)) \cdot dt \quad (30)$$

which is the available traffic that needs to be transmitted.

For each infinitesimal area $d\mathbf{y}$ in the feasible forwarding region $\mathbb{F}_{\mathbf{x}}$ of \mathbf{x} , the amount of nodes with good channel quality is

$$c_{\mathbf{x},\mathbf{y}} = \rho \cdot p_{\mathbf{x},\mathbf{y}}(\psi_{\text{th}}), \quad (31)$$

where $p_{\mathbf{x},\mathbf{y}}(\psi_{\text{th}})$ is the probability that the CTS message sent from a node at \mathbf{y} has a higher SNR than a given threshold ψ_{th} when received by the node at \mathbf{x} ((10) in [76]). Thus, the total amount of nodes in $\mathbb{F}_{\mathbf{x}}$ with good channel quality is

$$c_{\mathbb{F}_x} = \int_{\mathbb{F}_x} c_{x,y} dy. \quad (32)$$

Note that between any pair of nodes, at most one packet can be transmitted in a cycle T_p . Thus the maximum amount of traffic transmitted during a cycle T_p from $d\mathbf{x}$ to anywhere in \mathbb{F}_x is

$$\Omega_x^{\max} = \rho d\mathbf{x} \cdot c_{\mathbb{F}_x}. \quad (33)$$

Since the traffic is considered as a fluid and a packet takes one cycle to be transmitted between a pair of nodes, in dt , the maximum amount of traffic sent from $d\mathbf{x}$ is $\Omega_x^{\max} \cdot dt / T_p$. In the case where each node in $d\mathbf{x}$ has less than 1 available packet in its queue, i.e., $a_x(t) < 1 \cdot \rho$, it still takes an entire cycle to transmit them. In this case the actual transmitted traffic during dt is $\frac{a_x(t)}{1 \cdot \rho} \cdot \Omega_x^{\max} \cdot \frac{dt}{T_p}$. Accordingly, the transmitted traffic rate density at \mathbf{x} is

$$\omega_x(t) = \min \left[1, \frac{a_x(t)}{1 \cdot \rho} \right] \Omega_x^{\max} \frac{dt}{T_p} \cdot \frac{1}{d\mathbf{x}dt} = \min [a_x(t), \rho] \frac{c_{\mathbb{F}_x}}{T_p}, \quad (34)$$

where $a_x(t)$ is the available traffic density given by (30).

The outgoing traffic in each infinitesimal area is equally distributed to every node with good channel quality in its feasible region. Thus, the incoming traffic rate density, $\lambda_x(t)$, that is received from each infinitesimal area in the backward region, \mathbb{B}_x , is given by

$$\lambda_x(t) = \int_{\mathbb{B}_x} \omega_y(t) \cdot \frac{c_{y,x}}{c_{\mathbb{F}_y}} dy. \quad (35)$$

Within duration dt , the change in buffered traffic density is

$$dq_x(t) = (g_x(t) + \lambda_x(t) - \omega_x(t))dt, \quad (36)$$

and the buffered traffic density at time $t + dt$ changes to

$$q_x(t + dt) = q_x(t) + dq_x(t) \quad (37)$$

Thus, (34), (35), (36) and (37) describe the traffic dynamics of the network after $t = t_0$. Given the initial value of $q_x(t_0)$, the traffic rates in the network can be evaluated for any time instance $t > t_0$. Accordingly, the total incoming traffic rate at the sink, which models the total number of packets received by the sink, can be obtained.

Within the transmission range of the sink, the outgoing traffic rate density in (34) becomes

$$\omega_x(t) = a_x(t), \quad (38)$$

since the sink is always awake and the traffic can all be transmitted to the sink directly. Moreover, for these nodes, in (35), the backward region \mathbb{B}_x excludes the areas within the transmission range of the sink. Then, at the sink, the incoming traffic rate is calculated as

$$\Lambda(t) = \int_{\mathbf{x}: |\mathbf{x}-s| \leq r_{\text{th}}} \omega_x(t) d\mathbf{x}, \quad (39)$$

where r_{th} is the distance threshold around the sink within which all nodes directly send packets to the sink.

To calculate the incoming traffic rate at the sink, the entire network area can be discretized into small areas, where the time is also divided into small time steps. Initially, the buffered traffic density for every infinitesimal area in the network at time $t = 0$ is q_x^0 . $\lambda_x(t)$ and $\omega_x(t)$ are set as 0. Then, $\omega_x(t)$ and $\lambda_x(t)$ are calculated using (34) and (35), respectively. Then, $q_x(t)$ is updated for the next time step according to (36). This process is repeated for each time step, and $\Lambda(t)$ as a function of t is obtained.

To obtain the n -delay distribution from $\Lambda(t)$, we consider the traffic arrival process to be a Poisson process with variable rate according to $\Lambda(t)$. This model is also observed in empirical evaluations. Consequently, the n -delay distribution, $f_n(t)$, of the non-homogeneous Poisson process is given by [24, Ch. 2.4]:

$$f_n(t) = \frac{[\hat{\Lambda}(t)]^{(n-1)} \Lambda(t) e^{-\hat{\Lambda}(t)}}{(n-1)!}, \quad (40)$$

where $\hat{\Lambda}(t)$ is the integral of $\Lambda(t)$ over duration $(0, t]$.

Accordingly, we have the following theorem:

Theorem 2. *For a WSN system described in Sect. 4.2, the average n -delay and the (p, n) -delay bound of an event are*

$$\mu(n) = \int_0^\infty t f_n(t) dt, \quad (41)$$

$$j(p, n) = f_n^{-1}(p), \quad (42)$$

respectively, where $f_n(t)$ is given by (40).

Proof. Since $f_n(t)$ in (40) is the pdf of the n -delay, (41) and (42) are directly obtained according to the definition of the pdf.

4.4 Simplified Delay Model

The spatio-temporal fluid model decreases the complexity of the solution significantly. To further reduce the complexity, the low traffic in WSNs can be put to use.

In this simplified model, the network area is divided into small rings. Thus, the spatial calculation complexity is also reduced from 2D to 1D.

For a low traffic rate WSN, the queueing effect can be neglected. Moreover, the channel aware anycast operation allows one to assume that the channel error is negligible within a transmission range of R . For a node located at \mathbf{x} , after it receives a packet (locally generated or forwarded), in the duration t , the probability that there is no node in its feasible forwarding region $\mathbb{F}_{\mathbf{x}}$ waking up is

$$\begin{aligned} p_{\mathbf{x}}^{\text{nf}}(t) &\approx \prod_{\mathbf{y}=(l,\theta) \in \mathbb{F}_{\mathbf{x}}} \left[1 - \rho d\mathbf{y} p^{\text{wake}}(t) \right] = \left[1 - \rho d\mathbf{y} p^{\text{wake}}(t) \right]^{\frac{A_{\mathbb{F}_{\mathbf{x}}}}{d\mathbf{y}}} \\ &= \exp \left(-A_{\mathbb{F}_{\mathbf{x}}} \rho p^{\text{wake}}(t) \right), \end{aligned} \quad (43)$$

where the product is over $\mathbb{F}_{\mathbf{x}}$, divided according to the polar coordinates originated at \mathbf{x} , ρ is the network density, $A_{\mathbb{F}_{\mathbf{x}}}$ is the size of $\mathbb{F}_{\mathbf{x}}$, and $p^{\text{wake}}(t)$ is the probability that a node in each region wakes up during the period t . Since the wake period of each node is unsynchronized with each other, $p^{\text{wake}}(t)$ is irrelevant to the location. Moreover, since each node wakes up at uniformly distributed times, we have

$$p^{\text{wake}}(t) = \begin{cases} \frac{t}{T_p}, & 0 \leq t \leq T_p \\ 1, & t > T_p \end{cases}. \quad (44)$$

Therefore, the probability that at least one node in $\mathbb{F}_{\mathbf{x}}$ wakes up during t is

$$p_{\mathbf{x}}^{\text{fwake}}(t) = 1 - p_{\mathbf{x}}^{\text{nf}}(t) = \begin{cases} 1 - e^{-A_{\mathbb{F}_{\mathbf{x}}} \rho t / T_p}, & 0 \leq t \leq T_p \\ 1 - e^{-A_{\mathbb{F}_{\mathbf{x}}} \rho}, & t > T_p \end{cases}. \quad (45)$$

This is exactly the *cdf* of the single hop delay. Therefore, the *pdf* of single hop delay for a node at \mathbf{x} is

$$f_{\text{sh}(\mathbf{x})}(t) = d p_{\mathbf{x}}^{\text{fwake}}(t) / dt = \begin{cases} \frac{A_{\mathbb{F}_{\mathbf{x}}} \rho}{T_p} e^{-A_{\mathbb{F}_{\mathbf{x}}} \rho t / T_p}, & 0 \leq t \leq T_p \\ 0, & t > T_p \end{cases}. \quad (46)$$

The end-to-end delay distribution from location \mathbf{x} to the sink can be found as the convolution of single-hop delay distributions in the path as explained in Sect. 3. Thus, the *pdf* of end-to-end delay from \mathbf{x} to the sink is

$$f_{\text{e2e}(\mathbf{x})}(t) = \int_{\mathbf{y} \in \mathbb{F}_{\mathbf{x}}} f_{\text{e2e}(\mathbf{x}')} * f_{\text{sh}(\mathbf{x})}(t) \rho d\mathbf{y}, \quad (47)$$

where \mathbf{y} is the location (l, θ) . Since the queueing effect is neglected, the nodes with the same distance to the sink have the same end-to-end delay to the sink. Therefore,

the end-to-end delay distribution is calculated only once for all nodes with the same distance to the sink. This fact results in a significant reduction on the calculation time.

Suppose the packet generation function for a node at \mathbf{x} is $g_{\mathbf{x}}(t)$, then the packet reception rate from \mathbf{x} by the sink is

$$\lambda_{\mathbf{x}}(t) = g_{\mathbf{x}} * f_{e2e(\mathbf{x})}(t). \quad (48)$$

Then, the packet reception rate at the sink is the sum of traffic generated from each location in the event detection region. Thus,

$$\Lambda(t) = \int_{\mathbf{x} \in \mathbb{E}} g_{\mathbf{x}} * f_{e2e(\mathbf{x})}(t) d\mathbf{x}, \quad (49)$$

where \mathbb{E} is the region within the detection range, r_e , of the event location, \mathbf{x}_e , i.e., $\mathbb{E} = \{\mathbf{x} : |\mathbf{x} - \mathbf{x}_e| \leq r_e\}$.

Finally, the distribution of event detection delay is obtained by using (49) in (40), and the average n -delay and the (p, n) -delay bound of an event are obtained by Theorem 2.

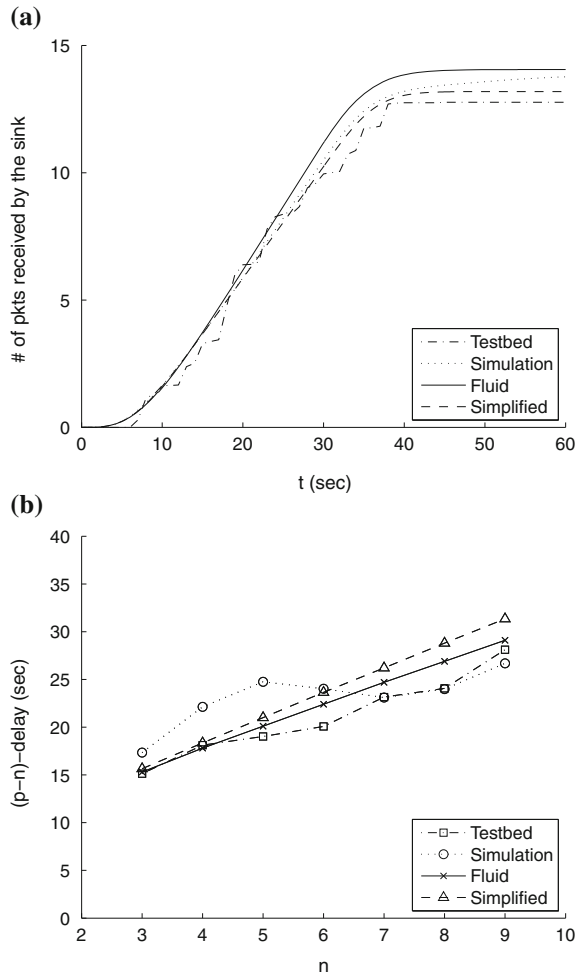
4.5 Experiments

Now, we evaluate the accuracy of the both models using testbed experiments and simulations. The average n -delay and the (p, n) -delay bound of an event in the experiments and simulations are used to compare against the framework. The spatio-temporal fluid model in the framework is implemented using C++ and the simplified model is implemented using MATLAB.

Again, using testbed experiments, we can evaluate the effectiveness of the models in small-scale deployments. Using a similar setup as in Sect. 3, experiments are performed. In addition, we also discuss simulation results using TOSSIM [39] with the same parameters.

The results are shown in Fig. 9a along with the results given by the two analytical models in (39) and (49). As can be seen in Fig. 9a, both testbed experiments and simulations validate the models. The (p, n) -delay bound for $p = 0.75$ is also calculated for different n 's. The results are shown in Fig. 9b. For the majority of the cases, testbed and simulation results are within 5 % of the model. Moreover, the results also confirm the accuracy of TOSSIM simulations, which are used in further evaluations of the two models in larger-scale networks. The results show that the model is accurate for low traffic high density WSN cases.

Fig. 9 The event delay results of testbed experiment, simulation, and models (a) The reception rate at the sink (b) The $(p;n)$ -delay bound of the event delay for $p = 0.75$



4.5.1 Comparison Between the Models

The spatio-temporal model and the simplified fluid model provides different advantages and disadvantages for network modeling. Both models yield the results in a significantly less time than simulations. For a typical network of 400 nodes, the simulation takes more than one day to complete, while the complete fluid model takes around 10 min to calculate, and the simplified only takes less than 1 min. It is shown in Fig. 9 that both models yield accurate results. The only cases where the result is less accurate is when the density is low, or when the traffic rate is high, which are not common configurations for WSNs.

Compared to the spatio-temporal model, the simplified model decreases the complexity at the cost of accuracy. The simplified model requires $O(\sqrt{A})$ time, where A

is the area of the network. On the other hand, since the spatio-temporal fluid model calculates the traffic rates for each location in the entire 2D network, it requires $O(A)$ time. On the other hand, the simplified model is less accurate when the nodes with the same distance to the sink do not have the same end-to-end delay. An example is a non-regular network where nodes density varies over the space. The complete fluid model, however, can be extended to provide accurate result in such networks when the density ρ in (29)–(34) by $\rho(\mathbf{x})$, a density function of corresponding location \mathbf{x} .

5 Network Lifetime Distribution

Now we turn our attention to energy consumption and lifetime in WSNs. Due to similar reasons, energy consumption also has a stochastic characteristic and its distribution characteristics should be analyzed. It is important to predict and guarantee the lifetime of a network before it is deployed for practical reasons.

We start by modeling the distribution of energy consumption at the node level and extend this analysis to the network level. Accordingly, lifetime of a node and the whole network is modeled using stochastic measures. The analysis in this section is based on the models presented in Sect. 3, since the analysis of both delay and energy consumption utilize the Discrete-Time Markov model at the node level. Moreover, we show that for large enough time durations, energy consumption *converges to a Normal distribution*. This result greatly reduces the computation cost for the analysis. We again use the anycast protocol as a case study to illustrate the use of the model and discuss validations with experiments and simulations.

5.1 Background

Historically, energy consumption and lifetime has mainly been analyzed in the average sense. Average energy efficiency is evaluated for specific protocols [8, 57, 74], and average energy consumption models are developed [16, 48]. The effects of routing strategies on energy consumption have also been investigated recently using a realistic radio model is adopted to analyze the tradeoff between single-hop long-distance transmissions and multi-hop short-distance transmissions [30]. The same problem is also investigated using an energy model focused on circuit level hardware [67]. Stochastic characteristics of energy consumption, however, cannot be captured by these models.

Lifetime of WSNs has also been investigated predominantly using average measures [18, 32, 37]. The probabilistic lifetime analysis of a cluster-based network for a TDMA MAC protocol [53] is a first step towards higher order statistics and here we provide a stochastic energy consumption and network lifetime model for multi-hop WSNs.

5.2 Problem Definition and System Model

In WSNs, energy is consumed by each node for various activities including sensing, data processing, and communication. We assume that each node is equipped with K sensors, and each sensor $k \in \{1, \dots, K\}$ is used to sense the physical environment every $T_{s,k}$ seconds (subscript “s” refers to “sensing”) with an energy consumption of $\varepsilon_{s,k}$. Based on the application requirements, a packet is generated locally if the sensed information satisfies event definitions. For each received and locally generated packet, the node processes the data with an energy consumption of ε_p . Moreover, the energy consumption for the communication, ε_c , is a variable dependent on the network parameters and the protocols running on each node.

For a given random network topology, we are interested in the following problems:

1. Given a period of time T , what is the energy consumption distribution, $F_{E(x,T)}(e)$, of a node at \mathbf{x} ?
2. Given the energy consumption distribution, what is the lifetime distribution, $F_{L(x,C(x))}(t)$, of a node at \mathbf{x} ?
3. Given the energy consumption distribution for each node \mathbf{x} in the network, what is the distribution of the network lifetime, $F_{NL}(t)$?

Next let us provide an overview of the solutions for the above problems. The details of the models are then elaborated in Sects. 5.3 and 5.4.

5.2.1 Single Node Energy Consumption Distribution

The randomness in energy consumption and the associated lifetime is due to the randomness because of the wireless channel errors and queueing operation. Accordingly, for a node at \mathbf{x} , the r.v. for energy consumption during a given time period, T , is expressed as:

$$E(\mathbf{x}, T) = E_s(\mathbf{x}, T) + E_{cp}(\mathbf{x}, T), \quad (50)$$

where $E_s(\mathbf{x}, T)$ is the r.v. of energy consumption for sensing, and $E_{cp}(\mathbf{x}, T)$ is the r.v. of energy consumption for communication and processing. These two terms capture the randomness due to protocol operation. Accordingly, the *pdf* of the total energy consumption of a node at \mathbf{x} is

$$f_{E(\mathbf{x},T)}(e) = f_{E_s(\mathbf{x},T)} * f_{E_{cp}(\mathbf{x},T)}, \quad (51)$$

where the *pdf* of the corresponding r.v.s. in (50) are convolved.

(1) Energy Consumption for Sensing: During any given time duration T starting at t_0 , i.e., the period $[t_0, t_0 + T)$, for some sensor k with periodic sensing interval $T_{s,k}$ and energy consumption per sensing $\varepsilon_{s,k}$, denote the first sensing activity for sensor k after t_0 occurs at t_{k1} . The number of sensing activities during T is then $n_k(T) = \lceil (t_0 + T - t_{k1}) / T_{s,k} \rceil$. Since t_0 is independent of sensing activities, $t_{k1} - t_0$

is a r.v. uniformly distributed in range $[0, T_{s,k})$. Therefore, the *pmf* of $n_k(T)$ is given by

$$f_{n_k(T)}(n) = \begin{cases} N_{s,k} - n + 1, & n = \lfloor N_{s,k} \rfloor + 1 \\ n + 1 - N_{s,k}, & n = \lfloor N_{s,k} \rfloor \\ 0, & \text{otherwise} \end{cases}, \quad (52)$$

where $N_{s,k} = \frac{T}{T_{s,k}}$. The *pdf* of energy consumption for all K sensors during T is obtained as

$$\begin{aligned} f_{E_s(T)}(e) &= \sum_{k=1}^K \sum_n n \cdot \delta(e - f_{n_k(T)}(n)) \\ &= \sum_{k=1}^K \left[(N_{s,k} - \lfloor N_{s,k} \rfloor) \cdot \delta(e - (\lfloor N_{s,k} \rfloor + 1) \varepsilon_{s,k}) \right. \\ &\quad \left. + (\lfloor N_{s,k} \rfloor + 1 - N_{s,k}) \cdot \delta(e - \lfloor N_{s,k} \rfloor \varepsilon_{s,k}) \right]. \end{aligned} \quad (53)$$

(2) Energy Consumption for Communication and Processing: Next, we briefly introduce the model for the analysis of the energy consumption for communication and processing, $E_{cp}(T)$, and leave the details of the model to Sect. 5.3.

The energy consumed by communication and data processing at each node in the network is modeled by a discrete-time queueing system with time unit T_u , which is characterized by its traffic inter-arrival distribution and service process. More specifically, in each time unit, the traffic inter-arrival is modeled according to a Bernoulli process, and a variant of the Discrete Time Markov Process (DTMP) discussed in Sect. 3.3 is used to model the service behavior.

Similar to the DTMP model in Sect. 3.3, the DTMP for the energy consumption analysis is represented by a Discrete-Time Markov Chain (DTMC) $\{X_n\}$. Contrary to the delay analysis, each state v in $\{X_n\}$ is also associated with an amount of energy, ε_v , consumed for the corresponding activity during T_u . The communication and data processing behaviors of each node are represented by transitions among states in $\{X_n\}$. We defer the detailed explanation of this DTMC to Sect. 5.3. Based on this DTMC, the *pdf* of the single-node energy consumption for communication and data processing, $E_{cp}(T)$, is found for any given duration T .

5.2.2 Node Lifetime and Network Lifetime Distributions

For a battery-powered sensor node, its lifetime distribution depends on the energy consumption distribution during any given period T , and the total capacity of its battery C . The network lifetime distribution depends on the lifetime distribution for each node, and how the network lifetime is defined. In the following, we focus on the lifetime defined as follows.

Definition 6 *The network lifetime is defined as the duration before the battery depletion of the first node.*

5.3 Single Node Energy Consumption Distribution

The energy consumed by communication and data processing for a node is represented by the energy costs associated with transitions among states in Markov chain $\{X_n\}$. In the following, based on the discussion in Sect. 3.3, the construction of states and transitions in $\{X_n\}$ is discussed.

5.3.1 Structure of Markov Chain $\{X_n\}$

According to the MAC protocol employed, the structures of $\{C_n\}_m$ and $\{I_n\}$ are parameterized by the following variables: \mathbf{P}_I , \mathbf{P}_C , α_I , α_C , \mathbf{t}_I^s , \mathbf{t}_C^s , \mathbf{t}_C^f , λ_I , and λ_C . The definitions of these variables are given in Sect. 3. Accordingly, the transition probability matrix, \mathbf{Q}_X , of the entire Markov chain $\{X_n\}$ can be found based on (5). Then, the equilibrium state probability vector, π , for $\{X_n\}$ is calculated by solving $\pi \mathbf{Q}_X = \pi$.

5.3.2 Energy Consumption for Communication and Processing

Now, based on the DTMC, we derive the distribution of energy consumption for communication and processing. Suppose at the beginning of a time unit T_u , the node is in state v of $\{X_n\}$, and during the time unit, the energy consumption of the node for communication and data processing is ε_v . The value of ε_v is obtained from measurement, or is calculated according to the specifications of the hardware platform. The *cdf* and the *pdf* of $E_{cp}(T_u)$, the energy consumption during the time unit, are $G_v(e) = u(e - \varepsilon_v)$ and $g_v(e) = \delta(e - \varepsilon_v)$, respectively, where $u(\cdot)$ is the unit step function and $\delta(\cdot)$ is the delta function.³ We denote

$$H_{v,v'}^{(1)}(e) = G_v(e)q_{v,v'} = \Pr\{E_{cp}(T_u) \leq e \cap v \xrightarrow{1} v'\}, \quad (54)$$

$$h_{v,v'}^{(1)}(e) = g_v(e)q_{v,v'} = dH_{v,v'}^{(1)}(e)/de, \quad (55)$$

where $v \xrightarrow{1} v'$ represents the event that $\{X_n\}$ transitions from state v to v' in one time unit, and $q_{v,v'}$ is the (v, v') th element of the transition probability matrix, \mathbf{Q}_X , in (5).

³ Although a discrete time Markov process is used for the model, the energy consumption is continuous. Thus the *pdf*, as opposed to the *pmf*, is used to characterize the distribution.

For a given period T , the number of time units of T_u is $\hat{T} \sim T/T_u$ (since T_u is usually chosen to be very small, T is approximated as an integer multiple of T_u). After \hat{T} time units ($\hat{T} > 1$), the *cdf* of energy consumption is found to be

$$\begin{aligned} H_{v,v'}^{(\hat{T})}(e) &= \Pr\{E_{cp}(T) \leq e \cap v \xrightarrow{\hat{T}} v'\} = \int_0^e h_{v,v'}^{(\hat{T})}(\epsilon) d\epsilon \\ &= \int_0^e \sum_{v'' \in \mathcal{S}} (h_{v,v''}^{(1)} * h_{v'',v'}^{(\hat{T}-1)})(\epsilon) d\epsilon \end{aligned} \quad (56)$$

where \mathcal{S} is the set of all states in $\{X_n\}$. Therefore, if the matrix of $h_{v,v'}^{(\hat{T})}(e)$ is denoted by $\mathbf{h}^{(\hat{T})}(e)$, then $\mathbf{h}^{(\hat{T})}(e)$ is the \hat{T} -fold convolution of $\mathbf{h}^{(1)}(e)$.

The energy consumption distribution during T depends on the initial state of the system at the beginning of this period, which is usually randomly chosen. Thus, the initial state probability vector is represented by the equilibrium state probability vector $\boldsymbol{\pi}$. After \hat{T} time units, the *pdf* and the *cdf* of the energy consumption are

$$\begin{aligned} f_{E_{cp}(T)}(e) &= \boldsymbol{\pi} \mathbf{h}^{(\hat{T})}(e) \mathbf{1}, \\ F_{E_{cp}(T)}(e) &= \int_0^e f_{E_{cp}(T)}(\epsilon) d\epsilon, \end{aligned} \quad (57)$$

respectively, where $\mathbf{1}$ is the appropriately dimensioned column vector containing all 1's.

5.3.3 Asymptotic Energy Consumption Distribution

When the given duration T is large, the calculation of energy consumption distribution may require extensive computing power, especially for (57). On the other hand, if a QBD process is modeled by a DTMC, and each state in the DTMC is associated with a cost, then the sum of the total cost during a given period T asymptotically approaches the Normal distribution as $T \rightarrow \infty$ [49]. This is the same case for the energy consumption distribution model, which can be approximated by a Normal distribution, whose mean and the variance are given by

$$\lim_{T \rightarrow \infty} \mu_{cp}(T) = \hat{T} \mu_{cp,u} = \hat{T} \boldsymbol{\pi} \boldsymbol{\epsilon}, \quad (58)$$

$$\lim_{T \rightarrow \infty} \sigma_{cp}^2(T) = \hat{T} \sigma_{cp,u}^2 = \hat{T} \left(\sum_{v \in \mathcal{S}} (\epsilon_v - \boldsymbol{\pi} \boldsymbol{\epsilon})^2 \pi_v + 2\boldsymbol{\beta} \boldsymbol{\epsilon} \right), \quad (59)$$

respectively, where $\hat{T} = T/T_u$ is the number of time units in T , $\mu_{cp,u}$, and $\sigma_{cp,u}^2$ are the mean and variance of E_{cp} during each time unit T_u . Moreover, $\boldsymbol{\pi}$ is the equilibrium state probability vector of $\{X_n\}$, \mathcal{S} is the set of states in $\{X_n\}$, and π_v

is the equilibrium state probability for state v . Finally, $\boldsymbol{\varepsilon}$ is the vector of ε_v for each state v in $\{X_n\}$, and $\boldsymbol{\beta}$ is an intermediate vector variable which is obtained by solving the following set of equations [49]:

$$\boldsymbol{\beta}(\boldsymbol{Q}_X - \boldsymbol{I}) = -\boldsymbol{\gamma} \boldsymbol{Q}_X, \quad (60)$$

$$\boldsymbol{\beta} \mathbf{1} = 0, \quad (61)$$

where $\boldsymbol{\gamma}$ is a row vector whose v -th element is $(\varepsilon_v - \boldsymbol{\pi} \boldsymbol{\varepsilon}) \pi_v$.

Then, the asymptotic distribution for $E_s(T)$, the energy consumption by sensing, is also derived. For each sensor k , when $T \rightarrow \infty$, $\left\lfloor \frac{T}{T_{s,k}} \right\rfloor \approx \frac{T}{T_{s,k}} \approx \left\lfloor \frac{T}{T_{s,k}} \right\rfloor + 1$. Thus, (53) becomes

$$f_{E_s(T)}(e) \approx \delta \left(e - \sum_{k=1}^K \frac{\varepsilon_k T}{T_{s,k}} \right). \quad (62)$$

In other words, the energy consumption is approximately linear to T with a constant coefficient equal to $\sum_{k=1}^K \varepsilon_k / T_{s,k}$.

Therefore, the following results are obtained:

Theorem 3. *When $T \rightarrow \infty$, the energy consumption of a node during T asymptotically approaches the Normal distribution, with the mean and variance linear to T and given by*

$$\mu(T) = \hat{T} \left(\mu_{cp,u} + \sum_{k=1}^K \frac{\varepsilon_k T_u}{T_{s,k}} \right), \quad (63)$$

$$\sigma^2(T) = \hat{T} \sigma_{cp,u}^2 + cT^2, \quad (64)$$

where $\hat{T} = T / T_u$ is the number of time units in T .

Proof. The proof is trivial by combining (58), (62).

5.4 Lifetime Distribution Analysis

Using the *pdf* of energy consumption $f_{E(T)}(e)$ in (57) for any given period T , the lifetime distribution of a node, and further, the entire network, can be found as follows.

5.4.1 Single-Node Lifetime Distribution

The distribution of the lifetime for a given node, $L(C)$, is a function of its total battery capacity C . Initially, the node has a battery residual of C . After duration T , the *pdf*

of remaining energy in the battery is $f_{C-E(T)}(e)$. The probability that the node has a shorter lifetime than duration T is the probability that the remaining energy after T is lower than 0. Thus, the *cdf* of the node lifetime is

$$F_{L(C)}(T) = \Pr\{L(C) \leq T\} = \Pr\{C - E(T) \leq 0\}. \quad (65)$$

As explained in Sect. 5.3.3, when T is large, $E(T) \sim \mathcal{N}(\mu(T), \sigma^2(T))$, where $\mu(T)$ and $\sigma^2(T)$ are given by Theorem 3. Thus, the *cdf* of single-node lifetime is approximated as

$$F_{L(C)}(t) \approx \mathcal{Q}\left(\frac{\mu(t) - C}{\sqrt{\sigma^2(t)}}\right). \quad (66)$$

5.4.2 Network Lifetime Distribution

Since every node needs to be alive during the network lifetime, the network lifetime (NL) distribution is obtained for a WSN with random deployment as:

$$F_{NL}(t) \approx 1 - \prod_{\mathbf{x} \in \mathcal{A}} (1 - p_{\text{ex}}(\mathbf{x}) \Pr\{L(\mathbf{x}, C(\mathbf{x})) \leq t\}), \quad (67)$$

where $L(\mathbf{x}, C(\mathbf{x}))$ is the lifetime for a node located at \mathbf{x} , if any, with battery capacity $C(\mathbf{x})$. Using the approximation in (66) for the single-node lifetime distribution, the network lifetime distribution is approximated by

$$F_{NL}(t) \approx 1 - \prod_{\mathbf{x} \in \mathcal{A}} \left(1 - p_{\text{ex}}(\mathbf{x}) \mathcal{Q}\left(\frac{C(\mathbf{x}) - \mu(\mathbf{x}, t)}{\sqrt{\sigma^2(\mathbf{x}, t)}}\right)\right), \quad (68)$$

where $\mu(\mathbf{x}, t)$, $\sigma^2(\mathbf{x}, t)$ are given by Theorem 3 for the node located at \mathbf{x} . Moreover, \mathcal{A} is the network area. To calculate the product, area \mathcal{A} is discretized into small areas of size $\Delta\mathbf{x}$, and $p_{\text{ex}}(\mathbf{x})$ is the probability that there exist a node in the small area around \mathbf{x} , and is a function of the network density ρ . It is obtained by $p_{\text{ex}}(\mathbf{x}) = \rho\Delta\mathbf{x}$.

5.5 Case Study: Anycast Protocol

Using the anycast protocol, now we illustrate how the analysis model can be used for communication protocols. For the energy and lifetime analysis with anycast protocol, we assume that nodes are deployed in a circular plane of radius R , have a homogeneous battery capacity C , and generate a homogeneous amount of local traffic to a sink located at the center of the plane. Because of the symmetry, node-specific variables are the same for each narrow ring with radius r , and are indexed by r . In

the following analysis, when there is no ambiguity, the subscript r in ring-specific variables is omitted.

5.5.1 Energy Consumption in Each State

During the protocol operation,, a node conducts one of the following communication tasks: transmission, listening, receiving, and sleeping. Listening and receiving are considered the same since most popular architectures, such as Mica2 [62] and TelosB [63], consume similar power for these tasks. We also ignore the energy consumed for the data packet transmission. This is a valid simplification because majority of the energy is consumed for idle listening and beacon transmissions. As a result, there are three types of states in $\{X_n\}$: Beacon transmission, Sleeping, and Listening. Nodes consume a specific amount of energy ε_v in each state v .

In practice, since battery voltage drops over time, battery capacity is often measured with normalized voltage. Therefore, energy is represented in the units of A·s. In sleeping and listening states, the energy consumed during a time unit, T_u , are $\varepsilon_{sl} = I_{sl}T_u$, and $\varepsilon_{li} = I_{li}T_u$, respectively, where I_{sl} and I_{li} are the measured current drawn from the battery in the sleep and listening modes, respectively.

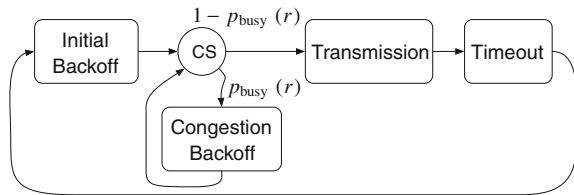
The power consumption when the node is transmitting beacon packets, ε_b , depends on the beacon transmission process shown in Fig. 10. The node transmits beacons only in a portion of time, and the portion, ω_b , should be obtained first to determine ε_b . For a node within ring r , ω_b is expressed as

$$\omega_b(r) = \frac{T_{tx}}{\left(\frac{T_{ibo}^{\max}}{2} + \frac{T_{cbo}^{\max} p_{\text{busy}}(r)}{2(1-p_{\text{busy}}(r))^2} + T_{tx} + T_{to}\right)}, \quad (69)$$

where $p_{\text{busy}}(r)$ is the probability of sensing the channel busy, and is derived as follows.

First, the region within the transmission range of location \mathbf{x} , $\mathbb{C}(\mathbf{x})$, is divided into small areas according to the polar coordinates centered at the sink. Thus, in the small area ($r : r + \Delta r, \theta : \theta + \Delta \theta$), denote $p_{\text{ex}}(r)$ as the probability that there exists a node in this area, and $\phi_b(r)$ as the probability that at any time a node in this area, if it exists, is transmitting a beacon packet. Then $p_{\text{busy}}(r)$ is given by

Fig. 10 The process of transmitting beacon packets



$$p_{\text{busy}}(r) = 1 - \prod_{y=(r', \theta') \in \mathbb{C}(\mathbf{x})} \left(1 - p_{\text{ex}}(r') \phi_b(r')\right), \quad (70)$$

where $p_{\text{ex}}(r)$ is given by

$$p_{\text{ex}}(r) = \rho \Delta r \Delta \theta r, \quad (71)$$

where ρ is the node density. The probability that a node in this area is transmitting a beacon packet, $\phi_b(r)$, is given by $\phi_b(r) = \pi_b(r) \omega_b(r)$, where $\pi_b(r)$ is the total probability that the node is in one of the beacon transmission states in the DTMC $\{X_n\}$, and is given by adding the probabilities in the equilibrium state probability vector, $\boldsymbol{\pi}(r)$, corresponding to the beacon transmission states. Therefore, according to (70), for nodes located at \mathbf{x} in ring r , the portion of time in which they transmit beacon messages, $\omega_b(r)$, depends on its values for other nodes in its neighborhood, $\mathbb{C}(\mathbf{x})$. An iterative procedure is used for all r 's to calculate $\omega_b(r)$ at the end of Sect. 5.5.

Then, $\varepsilon_b(r)$ is obtained by

$$\varepsilon_b(r) = \left(I_{\text{li}}(1 - \omega_b(r)) + I_{\text{tx}} \omega_b(r) \right) T_u, \quad (72)$$

where I_{li} and I_{tx} is the measured current when the node is listening and transmitting, respectively.

Finally, for this case study, we assume that the data processing time is far shorter than a time unit T_u .

Since data processing is conducted when packets are generated or received, a fixed amount of energy, ε_p , is added to the energy consumption in the first state of each $\{C_n\}$.

5.5.2 Communication and Data Processing Energy Consumption

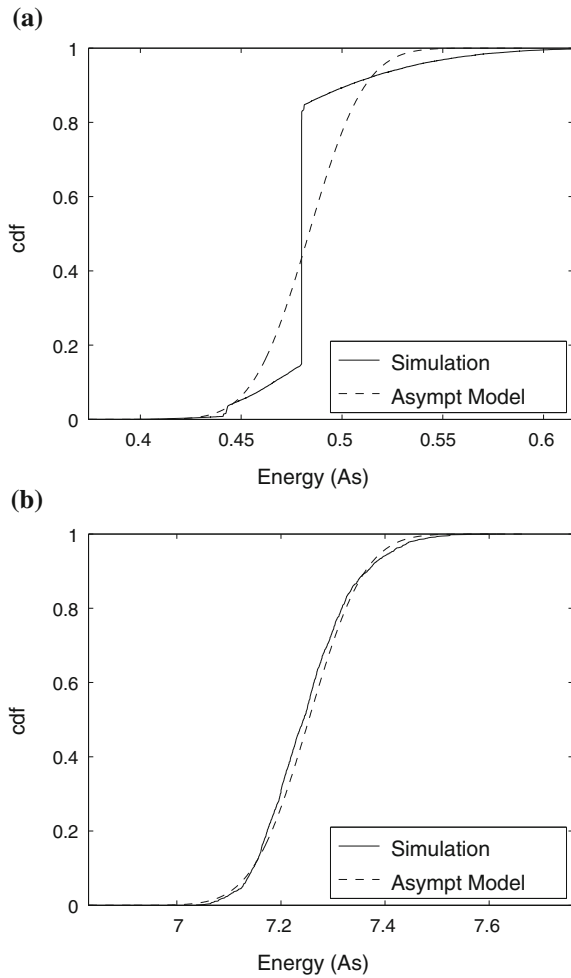
The other parameters in $\{X_n\}$, i.e., transition probability matrices and traffic rates in $\{I_n\}$ and $\{C_n\}$, are obtained according to the discussion in Sect 3.5. Then, the equilibrium state probability vector, $\boldsymbol{\pi}(r)$, for the DTMC $\{X_n\}$ is obtained for each node \mathbf{x} . It should be noted that while we solve $\omega_b(r)$, it is assumed that $\omega_b(r')$ for all nodes \mathbf{y} in range are known. This dependency is solved in an iterative manner. First, initial guesses of $\omega_b(r)$ for all rings are set to all 0's in our evaluation. Then, updated values of $\omega_b(r)$ are calculated. The iteration terminates when the difference between two consecutive iterations is negligible for each ring. Then, the energy consumption during a beacon time unit, $\varepsilon_b(r)$, is obtained according to (72). Finally, the communication and data processing energy consumption distribution for any single node is calculated according to (57).

5.6.2 Validation of the Normal Distribution Approximation

The asymptotic Normal distribution approximation of energy consumption for large T is validated using simulations [70]. Each topology is simulated for 10 days, and 100 different topologies are generated. In addition, additional energy consumptions for sensing and data processing are added to simulate a fully operational WSN application. The *cdf* of energy consumption for node at $r = 27$ m for $T = 2$ and 30 min are shown in Fig. 12a and b. The *cdf* of the asymptotic Normal distributions in Theorem 3 are also shown.

It can be observed in Fig. 12a and b that as the duration increases, the energy consumption distribution converges to the asymptotic Normal distribution.

Fig. 12 *cdf* of the energy consumption during longer periods. As the duration increases, the energy consumption approaches the asymptotic normal distribution (a) $T = 2$ min (b) $T = 30$ min



6 Conclusion

In this chapter, the stochastic analysis of end-to-end communication delay, event detection delay, energy consumption, and lifetime in WSNs is discussed. A Markov process based on the birth-death problem is used to model the transmission process in a multi-hop network. Accordingly, the effects of wireless channel errors and queuing delays on communication delay, energy consumption, and lifetime can be captured. The model is validated by extensive testbed experiments through several network configurations and parameters. The results show that the MARKov-based approach accurately models the distribution of the end-to-end delay and captures the heterogeneous effects of multi-hop WSNs.

An analytical model is also described to model the event detection in WSNs. In the framework, a spatio-temporal fluid model is utilized to obtain the distribution of the event detection delay. The average delay and soft delay bounds are then obtained. To reduce the calculation complexity, a simplified model is also derived, motivated by the fact that the queue build up in WSNs is negligible. Testbed experiments and simulations are used to validate the accuracy of both approaches.

Finally, the probabilistic analysis of the energy consumption is provided. Energy consumption for communication, data processing, and sensing are all captured by the analytical framework. The energy consumption distribution for each node is derived. It is shown that, when the time duration is long, the energy consumption converges to a Normal distribution, and the mean and variance of such distribution are also provided. With the help of energy consumption distribution, the lifetime distributions for each node and the entire network are derived. The described model is validated by both testbed experiments and simulations. The results show that the distribution of the energy consumption can be accurately modeled.

References

1. T. Abdelzaher, S. Prabh, R. Kiran, On real-time capacity limits of multihop wireless sensor networks, in *Proceedings of IEEE RTSS 2004*, pp. 359–370 (2004). doi:[10.1109/REAL.2004.37](https://doi.org/10.1109/REAL.2004.37)
2. K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks. *Ad Hoc Netw.* **3**(3), 325–349 (2005)
3. I.F. Akyildiz, T. Melodia, K.R. Chowdhury, A survey on wireless multimedia sensor networks. *Comput. Netw.* **51**(4), 921–960 (2007). doi:<http://dx.doi.org/10.1016/j.comnet.2006.10.002>
4. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey. *Comput. Netw. J. (Elsevier)* **38**(4), 393–422 (2002). doi:[http://dx.doi.org/10.1016/S1389-1286\(01\)00302-4](http://dx.doi.org/10.1016/S1389-1286(01)00302-4)
5. G. Armitage, *Quality of Service in IP Networks: Foundations for a Multi-Service Internet* (Macmillan Publishing Co., Inc. Indianapolis, 2000)
6. G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J. Sel. Areas Commun.* **18**(3), 535–547 (2000). doi:[10.1109/49.840210](https://doi.org/10.1109/49.840210)

7. N. Bisnik, A. Abouzeid, Queuing network models for delay analysis of multihop wireless ad hoc networks, in *IWCMC 2006: Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*, pp. 773–778. Vancouver, British Columbia, Canada (2006). doi:[10.1145/1143549.1143704](https://doi.org/10.1145/1143549.1143704)
8. M. Buettner, G.V. Yee, E. Anderson, R. Han, X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks, in *Proceedings of ACM SenSys 2006*. Boulder, CO (2006)
9. A. Burchard, J. Liebeherr, S. Patek, A min-plus calculus for end-to-end statistical service guarantees. *IEEE Trans. Inf. Theory* **52**(9), 4105–4114 (2006). doi:[10.1109/TIT.2006.880019](https://doi.org/10.1109/TIT.2006.880019)
10. Q. Cao, T. Yan, J. Stankovic, T. Abdelzaher, Analysis of target detection performance for wireless sensor networks, in *Proceedings of DCOSS 2005*, pp. 276–292. Marina del Rey, CA (2005)
11. S. Chakrabarti, A. Mishra, QoS issues in ad hoc wireless networks. *IEEE Commun. Mag.* **39**(2), 142–148 (2002)
12. D. Chen, P. Varshney, QoS support in wireless sensor networks: a survey, in *International Conference on Wireless Networks*, pp. 227–233. Citeseer (2004)
13. C. Chiasserini, R. Gaeta, M. Garetto, M. Griboudo, D. Manini, M. Sereno, Fluid models for large-scale wireless sensor networks. *Perform. Eval.* **64**(7–8), 715–736 (2007)
14. E. Crawley, R. Nair, B. Rajagopalan, H. Sandick, A Framework for QoS-based Routing in the Internet. RFC2386, August (1998)
15. R. Cruz, A calculus for network delay. I. Network elements in isolation. *IEEE Trans. Inf. Theory* **37**(1), 114–131 (1991). doi:[10.1109/18.61109](https://doi.org/10.1109/18.61109)
16. W. Dargie, X. Chao, M. Denko, Modelling the energy cost of a fully operational wireless sensor network. *Telecommun. Syst.* **44**(1–2), 3–15 (2010)
17. O. Dousse, C. Tavouraris, P. Thiran, Delay of intrusion detection in wireless sensor networks, in *Proceedings of ACM MobiHoc 2006*, p. 165. Florence, Italy (2006)
18. E. Duarte-Melo, M. Liu, Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks, in *Proceedings of IEEE GLOBECOM 2002*. Taipei, Taiwan (2002)
19. E. Duarte-Melo, M. Liu, A. Misra, A modeling framework for computing lifetime and information capacity in wireless sensor networks, in *WiOpt 2004*. Cambridge, UK (2004)
20. E.J. Duarte-melo, M. Liu, Data-gathering wireless sensor networks: Organization and capacity. *Comput. Netw.* **43**, 519–537 (2003)
21. E. Felemban, C.G. Lee, E. Ekici, R. Boder, S. Vural, Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks, in *Proceedings of IEEE INFOCOM 2005*, vol. 4, pp. 2646–2657. Miami, FL (2005). doi:[10.1109/INFCOM.2005.1498548](https://doi.org/10.1109/INFCOM.2005.1498548)
22. M. Fidler, An end-to-end probabilistic network calculus with moment generating functions, in *Proceedings of IEEE IWQoS*, pp. 261–270. New Haven, CT (2006). doi:[10.1109/IWQOS.2006.250477](https://doi.org/10.1109/IWQOS.2006.250477)
23. M. Franceschetti, O. Dousse, D. Tse, P. Thiran, Closing the gap in the capacity of wireless networks via percolation theory. *IEEE Trans. Inf. Theory* **53**(3), 1009–1018 (2007)
24. R.G. Gallager, *Discrete Stochastic Processes* (Kluwer Academic Publishers, New York, 1995)
25. L. Galluccio, S. Palazzo, End-to-End delay and network lifetime analysis in a wireless sensor network performing data aggregation, in *Proceedings of IEEE GLOBECOM 2009*, pp. 146–151. Honolulu, HI (2009)
26. K. Gopalan, T.C. Chiueh, Y.J. Lin, Probabilistic delay guarantees using delay distribution measurement, in *Proceedings of ACM MULTIMEDIA 2004*, pp. 900–907. New York, NY (2004). doi: <http://doi.acm.org/10.1145/1027527.1027734>
27. M. Griboudo, D. Manini, A. Nordio, C. Nordio, Analysis of IEEE 802.15. 4 sensor networks for event detection, in *Proceedings of IEEE Globecom 2009*, pp. 152–157. Honolulu, Hawaii (2009)
28. G.R. Gupta, N.B. Shroff, Delay analysis for multi-hop wireless networks, in *Proceedings of IEEE INFOCOM 2009*, pp. 412–421. Rio de Janeiro, Brazil (2009)
29. P. Gupta, P.R. Kumar, The capacity of wireless networks. *IEEE Trans. Inf. Theory* **IT-46**(2), 388–404 (2000)

30. J. Haapola, Z. Shelby, C. Pomalaza-Raez, P. Mahonen, Cross-layer energy analysis of multihop wireless sensor networks, in *Proceedings of IEEE EWSN 2005* (2005)
31. T. Issariyakul, E. Hossain, Analysis of end-to-end performance in a multi-hop wireless network for different hop-level ARQ policies, in *Proceedings of IEEE GLOBECOM 2004*, vol. 5, pp. 3022–3026. Dallas, TX (2004). doi:[10.1109/GLOCOM.2004.1378907](https://doi.org/10.1109/GLOCOM.2004.1378907)
32. D. Jung, T. Teixeira, A. Savvides, Sensor node lifetime analysis: models and tools. *ACM Trans. Sens. Netw.* **5**(1), 1–33 (2009). doi:[10.1145/1464420.1464423](https://doi.org/10.1145/1464420.1464423)
33. J. Kim, X. Lin, N. Shroff, Optimal anycast technique for delay-sensitive energy-constrained asynchronous wireless sensor networks, in *Proceedings of IEEE INFOCOM 2009*, pp. 412–421. Rio de Janeiro, Brazil (2009)
34. A. Koubaa, M. Alves, E. Tovar, Modeling and worst-case dimensioning of cluster-tree wireless sensor networks, in *Proceedings of IEEE RTSS 2006*, pp. 412–421. Rio de Janeiro, Brazil (2006). doi:[10.1109/RTSS.2006.29](https://doi.org/10.1109/RTSS.2006.29)
35. S. Kulkarni, A. Iyer, C. Rosenberg, An address-light, integrated mac and routing protocol for wireless sensor networks. *IEEE/ACM Trans. Netw.* **14**(4), 793–806 (2006). doi:[10.1109/TNET.2006.880163](https://doi.org/10.1109/TNET.2006.880163)
36. V.G. Kulkarni, Fluid models for single buffer systems. *Front. Queue. Models Appl. Sci. Eng.*, pp. 321–338 (1997)
37. S. Kumar, A. Arora, T. Lai, On the lifetime analysis of always-on wireless sensor network applications, in *Proceedings of IEEE MASS 2005*. Washington, DC (2005)
38. J. Lehoczy, Real-time queueing network theory, in *Proceedings of IEEE RTSS 1997*, pp. 58–67. San Francisco, CA (1997). doi:[10.1109/REAL.1997.641269](https://doi.org/10.1109/REAL.1997.641269)
39. P. Levis, N. Lee, M. Welsh, D. Culler, TOSSIM: accurate and scalable simulation of entire tinyos applications, in *Proceedings of ACM SenSys 2003*. Los Angeles, CA (2003)
40. H. Li, P. Shenoy, K. Ramamritham, Scheduling messages with deadlines in multi-hop real-time sensor networks, in *Proceedings of IEEE RTAS 2005*, pp. 415–425. San Francisco, CA (2005). doi:[10.1109/RTAS.2005.48](https://doi.org/10.1109/RTAS.2005.48)
41. S. Liu, K.W. Fan, P. Sinha, CMAC: An energy efficient mac layer protocol using convergent packet forwarding for wireless sensor networks, in *Proceedings of SECON 2007*, pp. 11–20. San Diego, CA (2007). doi:[10.1109/SAHCN.2007.4292813](https://doi.org/10.1109/SAHCN.2007.4292813)
42. Y. Liu, F. Lo Presti, V. Misra, D. Towsley, Y. Gum Fluid models and solutions for large-scale IP networks, in *Proceedings of ACM SIGMETRICS 2003*, pp. 91–101. San Diego, CA (2003)
43. D. Marco, E.J. Duarte-Melo, M. Liu, D.L. Neuhoff, On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data, in *Proceedings of IPSN'03*. Palo Alto, CA (2003)
44. W. Marcus, An architecture for QoS analysis and experimentation. *IEEE/ACM Trans. Netw. (TON)* **4**(4), 603 (1996)
45. D. McDysan, *QoS and Traffic Management in IP and ATM Networks* (McGraw-Hill, New York, 1999)
46. T. Melodia, M.C. Vuran, D. Pompili, The state of the art in cross-layer design for wireless sensor networks, in *Wireless Systems and Network Architectures in Next Generation Internet*, pp. 78–92 (2006)
47. P. Mohapatra, J. Li, C. Gui, QoS in mobile ad hoc networks. *IEEE Wireless Commun.* **10**(3), 44–53 (2003)
48. A. Muhammad Mahtab, B. Olivier, M. Daniel, A. Thomas, S. Olivier, A hybrid model for accurate energy analysis of WSN nodes. *EURASIP J. Embed. Syst.* **2011** (2011)
49. Y. Nazarathy, G. Weiss, The asymptotic variance rate of the output process of finite capacity birth-death queues. *Queue. Syst.* **59**(2), 135–156 (2008). doi:[10.1007/s11134-008-9079-4](https://doi.org/10.1007/s11134-008-9079-4)
50. R. Nelson, *Probability, Stochastic Processes, and Queueing Theory: The Mathematics of Computer Performance Modeling* (Springer, New York, 1995)
51. M. Neuts, J. Guo, M. Zukerman, H.L. Vu, The waiting time distribution for a TDMA model with a finite buffer and state-dependent service. *IEEE Trans. Commun.* **53**(9), 1522–1533 (2005). doi:[10.1109/TCOMM.2005.855014](https://doi.org/10.1109/TCOMM.2005.855014)

52. M.F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: an Algorithmic Approach* (Dover, New York, 1981)
53. M. Noori, M. Ardakani, Lifetime analysis of random event-driven clustered wireless sensor networks. *IEEE Trans. Mob. Comput.* **10**(10), 1448–1458 (2010)
54. R. Oliver, G. Fohler, Probabilistic estimation of end-to-end path latency in wireless sensor networks, in *Proceedings of IEEE MASS 2009*, pp. 423–431. Macau, China (2009). doi:[10.1109/MOBHOC.2009.5336970](https://doi.org/10.1109/MOBHOC.2009.5336970)
55. W. Pak, J.G. Choi, S. Bahk, Tier based anycast to achieve maximum lifetime by duty cycle control in wireless sensor networks, in *Wireless Communications and Mobile Computing Conference, 2008. IWCNC 2008. International*, pp. 123–128 (2008). doi:[10.1109/IWCNC.2008.22](https://doi.org/10.1109/IWCNC.2008.22)
56. P. Park, P. Di Marco, P. Soldati, C. Fischione, K. Johansson, A generalized markov chain model for effective analysis of slotted IEEE 802.15.4, in *Proceedings of IEEE MASS 2009*, pp. 130–139. Macau, China (2009). doi:[10.1109/MOBHOC.2009.5337007](https://doi.org/10.1109/MOBHOC.2009.5337007)
57. J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in *Proceedings of ACM SenSys 2004*. Baltimore, MA (2004). doi: <http://doi.acm.org/10.1145/1031495.1031508>
58. S. Pollin, M. Ergen, S. Ergen, B. Bougard, F. Catthoor, A. Bahai, P. Varaiya, Performance analysis of slotted carrier sense IEEE 802.15.4 acknowledged uplink transmissions, in *Proceedings of IEEE WCNC 2008*, pp. 1559–1564. Las Vegas, NV (2008). doi:[10.1109/WCNC.2008.279](https://doi.org/10.1109/WCNC.2008.279)
59. T. Sakurai, H. Vu, MAC access delay of IEEE 802.11 DCF. *IEEE Trans. Wireless Commun.* **6**(5), 1702–1710 (2007). doi:[10.1109/TWC.2007.360372](https://doi.org/10.1109/TWC.2007.360372)
60. J. Schmitt, F. Zdarsky, L. Thiele, A comprehensive worst-case calculus for wireless sensor networks with in-network processing, in *RTSS 2007*, pp. 193–202 (2007). doi:[10.1109/RTSS.2007.17](https://doi.org/10.1109/RTSS.2007.17)
61. O. Tickoo, B. Sikdar, Modeling queueing and channel access delay in unsaturated IEEE 802.11 random access MAC based wireless networks. *IEEE/ACM Trans. Netw.* **16**(4), 878–891 (2008). doi:<http://dx.doi.org/10.1109/TNET.2007.904010>
62. MICA2 sensor node. <http://www.xbow.com>
63. TelosB sensor node. <http://www.xbow.com>
64. S. Toupmpis, L. Tassiulas, Packetostatics: deployment of massively dense sensor networks as an electrostatics problem, in *Proceedings of IEEE INFOCOM 2005*. Miami, FL (2005)
65. L. Van Hoesel, T. Nieberg, J. Wu, P. Havinga, Prolonging the lifetime of wireless sensor networks by cross-layer interaction, in *IEEE Wireless Communications*, p. 79 (2004)
66. M.C. Vuran, I.F. Akyildiz, XLP: a cross layer protocol for efficient communication in wireless sensor networks. *IEEE Trans. Mob. Comput.* **9**(11), 1578–1591 (2010)
67. Q. Wang, M. Hempstead, W. Yang, A realistic power consumption model for wireless sensor network devices, in *Proceedings of IEEE SECON 2006*, pp. 286–295. Reston, VA (2006)
68. Y. Wang, M.C. Vuran, S. Goddard, Cross-layer analysis of the end-to-end delay distribution in wireless sensor networks, in *Proceedings of IEEE RTSS 2009*. Washington, DC (2009)
69. Y. Wang, M.C. Vuran, S. Goddard, Stochastic analysis of energy consumption in wireless sensor networks, in *Proceedings of IEEE SECON 2010*. Boston, MA (2010)
70. Y. Wang, M.C. Vuran, S. Goddard: Analysis of event detection delay in wireless sensor networks, in *Proceedings of IEEE INFOCOM 2011*. Shanghai, China (2011)
71. Y. Wang, M.C. Vuran, S. Goddard, Cross-layer analysis of the end-to-end delay distribution in wireless sensor networks. *IEEE Trans. Netw.* **20**(1), 305–318 (2012)
72. M. Xie, M. Haenggi, Towards an end-to-end delay analysis of wireless multihop networks. *Ad Hoc Netw.* **7**(5), 849–861 (2009). doi:[10.1016/j.adhoc.2008.04.010](https://doi.org/10.1016/j.adhoc.2008.04.010)
73. Y. Xue, M.C. Vuran, B. Ramamurthy, Cost-efficiency of anycast-based forwarding in duty-cycled wsns with lossy channel, in *Proceedings of IEEE SECON 2010*. Boston, MA (2010)
74. W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.* **12**(3), 493–506 (2004). doi: <http://dx.doi.org/10.1109/TNET.2004.828953>

75. S.N. Yeung, J. Lehoczky, End-to-end delay analysis for real-time networks, in *Proceedings of IEEE RTSS 2001*, pp. 299–309. London, UK (2001)
76. M. Zúniga, B. Krishnamachari, An analysis of unreliability and asymmetry in low-power wireless links. *ACM Trans. Sens. Netw.* **3**(2) (2007)

The Art of Wireless Sensor Networks

Volume 2: Advanced Topics and Applications

Ammari, H.M. (Ed.)

2014, XVI, 696 p. 250 illus., 151 illus. in color.,

Hardcover

ISBN: 978-3-642-40065-0