End-to-End Delay Analysis in Wireless Sensor Networks with Service Vacation

Ibrahim Alabdulmohsin, Amal Hyadi, Laila Afify, and Basem Shihada
CEMSE Division, KAUST, Saudi Arabia
E-mail: {ibrahim.alabdulmohsin, amal.hyadi, laila.afify, basem.shihada}@kaust.edu.sa

***Abstract*—In this paper, a delay-sensitive multi-hop wireless sensor network is considered, employing an $M/G/1$ with vacations framework. Sensors transmit measurements to a predefined data sink subject to maximum end-to-end delay constraint. In order to prolong the battery lifetime, a sleeping scheme is adopted throughout the network nodes. The objective is to present an expression for maximum hop-count as well as an approximate expression of the probability of blocking at the sink node upon violating certain end-to-end delay threshold. Using numerical simulations, we validate the proposed model and demonstrate that the blocking probability of the system for various vacation time distributions matches the simulation results.**

***Index Terms*—Wireless sensor networks, queueing theory, end-to-end delay, vacations, QoS**

I. Introduction

Wireless Sensor Networks (WSNs) have been a revolutionary emerging technology for many fields in science and industry [1]. This is because of their ability to form a vast network of small sensing devices, called *motes*, distributed in a sparse area to observe the surrounding environment. In such networks, not only nodes are capable of sensing and forwarding data, but also are capable of carrying out simple computations and filtering out transmitted data. Such features ensure wide range of applications for wireless sensor networks in many fields such as habitat monitoring, health monitoring, industrial applications, and simple home applications.

Sensor nodes in WSN are battery-equipped, low-power, and low-cost devices with limited sensing, data processing, transmission range, memory, and communication capabilities. For instance, the MICAz mote from Crossbow Technology is based on the Atmel ATmega128L 8-bit microcontroller. It supports only 8 MHz clock frequency, 128-KB flash program memory and 4-KB EEPROM. Moreover, the transmit data rate is limited to only 250 Kbps. In most applications, sensors are stationary or with limited mobility, and communication is typically either *many-to-one* (from sensors to sink) or *one-to-one* (between nodes themselves).

WSNs operate in a complex real-time, real world noisy environment. Such environment raises several challenges for WSNs design due to the unreliability of wireless communication medium and the real-time requirements of control applications. That is, WSNs not only share wireless communication challenges with regard to sensor-to-sensor communication, but also introduce their own unique challenges. For example, energy consumption in WSNs plays, in general, a much more crucial role than energy consumption in other wireless networks. Moreover, sensors often sense and sample data in real-time subject to maximum end-to-end delay constraints.

There is a tremendous amount of research presented on WSNs applications and challenges. However, limited research went into investigating the tradeoff between sensor sleeping time and latency. Today, most standards and protocols for WSNs lack the support of real-time requirements and sensitivity to delays. This limits the usefulness and applicability of these protocols in WSNs and hence, large scale deployments are hard or inefficient.

Considering the wireless sensor network applications, it is critical to analyze and quantify the inherent tradeoffs between latency and power efficiency. On one hand, end-to-end delay is a key metric in analyzing the system performance. In fact, wireless sensor networks are often subject to maximum latency constraint in mission critical applications such as in combat zones, where sensors detect moving targets, or in process and control systems, where sensors detect chemical gases and operating conditions of plant facilities. On the other hand, wireless sensors are designed to operate on substantially long time durations on small inexpensive batteries with limited lifetimes [2]. Therefore, there have been extensive efforts to devise efficient schemes for conserving energy in the network, albeit at the expense of incurring higher latencies. Introducing the concept of sleep and wake-up modes for relay hops is one such scheme that conserves energy [3], [4]. This can be done by allowing nodes to start a random vacation period when their queues are empty.

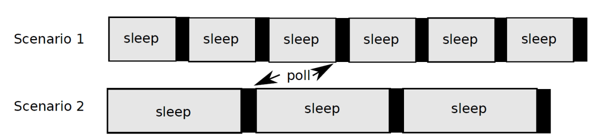

Fig. 1. A motivating scenario on the effect of sleep cycle time on energy consumption

The motivation behind introducing sleep cycles or vacations is depicted in Figure 1. In this figure, it is assumed that data traffic is limited, which is often the case in WSNs. Under such conditions, power consumption is primarily driven by polling. If the sleeping cycle is doubled, as is depicted in Figure 1 where Scenario 2 has a sleeping cycle that is twice as large as Scenario 1, then power consumption is nearly reduced by a factor of 200%. Pictorially, it is apparent that scenario one consumes more energy since it is switching to poll for received packets more often. The effect of sleep cycle duration extends to scenarios with active receptions and can be intuitively derived. Figure 2 shows the effect of sleep cycle times on the longevity of sensor motes. It is shown that a sleep time of 1.0 seconds increase longevity by a factor of 10000% compared to 0.01 seconds when the amount of traffic is negligible.

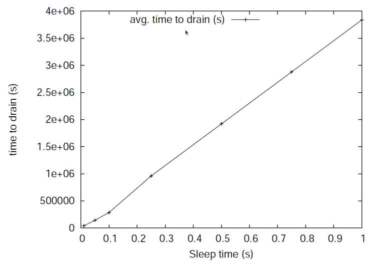

Fig. 2. Amount of time required to drain batteries for different sleep cycle times with no workload

Nevertheless, introducing sleep cycles or vacations increases latency due to residual sleeping time. To mitigate such impact, several approaches have been proposed to optimize the vacation scheduling algorithm such that the delay experienced in the network from source to sink does not harm the desired quality of service (QoS). For example, on-demand sleep-wake schemes have been suggested where a node wakes up upon receiving packets, thus triggering its working mode [5], [6]. Moreover, Kim, et al. proposes an anycast packet-forwarding scheme, where each node has a set of forwarding nodes and it chooses the node that wakes up the earliest to forward its packet to [7]. Using this approach, the delay introduced due to the residual vacation time is reduced.

Analyzing the performances of WSNs from a delay perspective has been the focus of a number of research works. For instance, authors in [8] investigated the system performance, in terms of energy consumption and data delivery delay, of a sensor network with sleeping mode presented as a Markov model. In [9], the delay performance of a wireless sensor network with date aggregation has been presented from a scheduling viewpoint. Authors in [10] presented event delay analysis of an event-driven WSN data transmission considering an M/G/1 vacation queueing model.

In this paper, we examine the performance of wireless sensor networks with an arbitrary vacation time distribution. Specifically, we look into a relay network that carries sensor measurements to a data sink node, and analyze how its maximum hop-count is determined by maximum end-to-end delay constraint. In addition, we look into probability of packet dropping, which arises if sensor packets do not adhere to the desired delay threshold constraint.

The rest of the paper is organized as follows. First, we introduce the system model in Section II. Then, we provide the mathematical analysis for the proposed model in Section III. Section IV demonstrates validation of the analytic model via simulation results. Finally, conclusions and future work are provided in Section V.

II. System Model

We consider a wireless sensor network, shown in Figure 3, in which M sensors collect observations from the surrounding environment. Then, transmit them to the adjacent relay nodes. Sensor events (i.e., data packet arrival) is modeled as an exponential inter-arrival times with rates $\gamma_m, m = 1, 2, ..., M$. In addition to data packets, control packets are also injected into the network from external sources. The relay nodes are modeled as single server facilities with two priority queues and are responsible for forwarding sensor measurements to the data sink node.

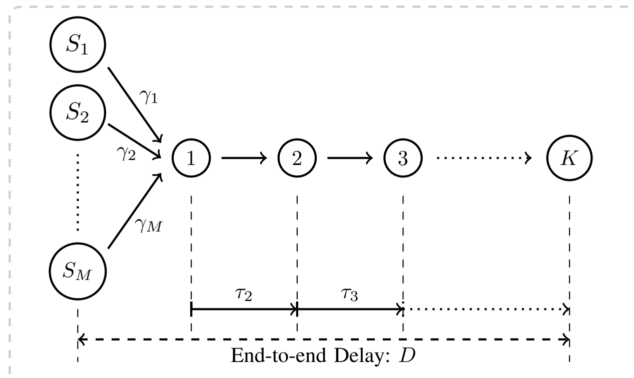

Fig. 3. An overview of the system: we have multiple sensors reporting measurements to relay node, which forwards packets across the network to a data sink node.

We assume that the control packets are of high priority, and thus will be scheduled to the higher priority queue denoted as Q_k^H, where $k = 1, 2, ..., K$ is the node index. The data packets are scheduled to the lower priority queue $Q_k^L, k = 1, 2, ..., K$. In some emergency situations, some data packets are scheduled in the high-priority queue with probability $1-p_t$. In general, both control and data packet arrivals are assumed to be Poisson distributed with rates λ_C and $\lambda_D = \sum_{m=1}^{M} \gamma_m$, respectively. Whereas data traffic is assumed to originate only from the sensors, control traffic is assumed to be injected into each relay hop from external sources as mentioned earlier. The model presented herein can be easily extended to the case where both data and control traffic might arrive from external sources, but this setup will be avoided in this paper to simplify discussion and notation.

Since there are two different types of traffic and two different priority queues, where data traffic can be scheduled in some emergency cases to the high-priority queues, the total rate of high-priority traffic λ_H and low-priority traffic λ_L is given by:

$$\lambda_H = \lambda_C + (1 - p_t) \cdot \lambda_D \quad (1)$$

$$\lambda_L = p_t \cdot \lambda_D \quad (2)$$

Generally, nodes are modeled as non-preemptive priority queues with exponentially distributed service time of mean

$\overline{X}_i$ for node $i \in \{1, 2, \ldots, K\}$. For the purpose of energy conservation at each node in the network, a sleeping policy is adopted such that a node goes immediately to sleep when it finds that both of its queues are empty. Such sleeping period or *vacation time* is assumed to have an arbitrary distribution with mean $\overline{V_i}$ and second moment $\overline{V_i^2}$ for node $i \in \{1, 2, \ldots, K\}$. To ensure that the Poisson property of traffic is not severely disturbed, we assume that $\overline{V} \ll \frac{1}{\lambda_H + \lambda_L}$. In other words, the mean inter-arrival time of traffic is much larger than mean vacation time. In practice, this is a reasonable assumption since the purpose of introducing sleeping times is to conserve energy without severely impacting end-to-end delay. In addition, we assume that a node continues to receive packets during its sleep periods, and starts to serve them upon waking-up. A node is assumed to wake-up when the random sleeping period ends, which depends on the vacation time and follows a certain distribution. This introduces extra delay for the jobs of both flows due to the residual vacation time they encounter.

In order to maintain a predefined QoS, the expected delay experienced by each data packet since its transmission till it reaches the data sink node must not exceed a predefined threshold T_{QoS}. Therefore, a data packet with a cumulative delay exceeding T_{QoS} upon arrival to the data sink node will be dropped. Thus, if we let D be a random variable that stands for end-to-end delay, then the probability of dropping a data packet at the sink node is given by:

$$P_{\text{blocking}} = \Pr\{\mathrm{D} > \mathrm{T_{QoS}}\} \tag{3}$$

The objective of the framework folds into:

1) Computing the maximum allowable hop-count K^* such that $\mathbb{E}[D] < T_{QoS}$.
2) Estimating the packet dropping probability for a given number of hops.

It is worth mentioning that the objective is not to minimize packet dropping probability since selecting maximum hop-count such that $\mathbb{E}[D] \approx T_{QoS}$ actually implies that almost half of the packets will be dropped. A more realistic approach would be to compute the maximum hop-count $K(\nu)$ such that $D < T_{QoS}$ with a probability that is at least given by $1 - \nu$. However, in this framework, we will not pursue the latter approach, and we will restrict our analysis to the two above objectives. Looking into how the above metrics vary with service, scheduling, and sleeping policy in the network presents a complex model that captures many of the inherent tradeoffs in wireless sensor network design.

III. Analysis

In this section, we present analytical expressions for the maximum hop-count K^* as well as the packet dropping probability. The approach used in this paper follows closely standard derivations for $M/G/1$ queues (see for instance [11], [12]). A summary of notations is given in Table I.

As stated in the model description, we have K non-preemptive relay nodes in the network with two priority queues and, possibly, different service rate or vacation time distributions. Despite the fact that both packet arrival rate as well as service times are assumed to be exponentially distributed, the overall system is not $M/M/1$ because vacation time distribution is arbitrary. Hence, the overall system is model as $M/G/1$. Initially, we are interested in obtaining the mean waiting time $\overline{W_D}$ and second moment $\overline{W_D^2}$ that will be experienced by data packets at each relay hop. Without loss of generality, we can look into each relay hop in isolation and temporarily drop the node index i. This is because, under stated assumptions of exponential service times and small average vacation times, the output of each queue is approximately Poisson distributed. To achieve our goal, we first note that the waiting time of a high priority packet, denoted W_H, is given by:

TABLE I
NOTATIONS

Symbol	Definition
M	Number of sensors taking measurements
K	Total number of relay nodes in the network up to the data sink node
$\lambda_{C,i}$	Poisson arrival rate of control packets at hop i
$\lambda_{D,m}$	Poisson arrival rate of data packets from sensor m
λ_p	Aggregate Poisson arrival rate of priority p, $p \in \{H, L\}$
X_j	Service time random variable for packet j
χ_i	Service time random variable at node i
p_t	probability of scheduling a data packet to Q_L in an emergency situation
$\rho_{p,i}$	Utilization of priority p flow, $p \in \{H, L\}$ at node i
$\overline{V_i}$	Mean vacation time duration at node i
$\overline{V_i^2}$	Second moment of vacation time duration at node i
R_v	Residual vacation time
R_s	Residual service time
$W_{p,i}$	Waiting time in the queue for packets of priority p, $p \in \{H, L\}$ at node i
$\omega_{p,i}$	Waiting time in node i for packet of priority p, $p \in \{H, L\}$
T_i	Average response time at node i
τ_i	Propagation delay between node $i-1$ and node i
N_p	Number of packets scheduled in the queue with priority p, $p \in \{H, L\}$

$$W_H = \sum_{j=1}^{N_H} X_j + R. \tag{4}$$

where, X_j is the service time of packet j and R is the residual time. Here, R can be either residual service time R_s or residual vacation time R_v depending on utilization. The first moment of this residual time can then be written as

$$\overline{R} = \frac{1}{2}\left[(\rho_H + \rho_L) \cdot \frac{\overline{X^2}}{\overline{X}} + (1 - \rho_L - \rho_H) \cdot \frac{\overline{V^2}}{\overline{V}}\right]. \tag{5}$$

From (4), and upon applying Little's formula to the average size of high priority queue N_H, we obtain:

$$\overline{W}_H = \frac{\overline{R}}{(1-\rho_H)}. \quad (6)$$

To reiterate, $\rho_H = \lambda_H \cdot \bar{X}$ is the fraction of time a relay node is serving high priority traffic. Proceeding with additional algebraic manipulations yield the desired second moment of waiting time for high-priority traffic:

$$\overline{W^2}_H = \overline{N}_H \cdot \mathrm{Var(X)} + \left[\left(1 + \frac{\rho_{\mathrm{H}}}{1-\rho_{\mathrm{H}}}\right)\overline{\mathrm{R}}\right]^2 + \mathrm{Var(R)}, \quad (7)$$

where

$$\begin{aligned} \overline{N}_H &= \lambda_H \, \overline{W}_H, \\ \rho_H &= \lambda_H \cdot \overline{X}, \\ \mathrm{Var}(R) &= \overline{R^2} - \overline{R}^2. \end{aligned}$$

Note that equation (7) was obtained by raising both sides of (4) to the 2[nd] power and taking expectation.

The above expression for $\overline{W_H^2}$ is nearly complete except that we need to evaluate $\overline{R^2}$. To do this, we use the law of total expectation, which states that $\mathbb{E}[Y] = \mathbb{E}[\mathbb{E}[Y|X]]$. Applying this, we obtain:

$$\begin{aligned} \overline{R^2} &= \rho_H \overline{R_H^2} + \rho_L \overline{R_L^2} + (1-\rho_H-\rho_L)\overline{R_v^2} \\ &= \frac{1}{3}\left((\lambda_H+\lambda_L)\overline{X^3} + (1-\rho_H-\rho_L)\frac{\overline{V^3}}{\overline{V}}\right). \end{aligned} \quad (8)$$

The above equations were derived for the high-priority traffic. As a sanity check, suppose that we have very limited high priority traffic, i.e. $\rho_H \to 0$. Then, from (7) we obtain $\overline{W_H^2} \approx \overline{R^2}$, which agrees with expectation because in the latter case $W_H \to R$. In other words, if high priority traffic is limited, then high priority queues contain at most one packet almost all the time, and waiting time for high priority packets is chiefly due to residual time only. Similarly, as $\rho_H \to 1$, we know that $\rho_L \to 0$ by assumption of stability. Consequently, the same equation implies that $\mathrm{Var}(W_c) \approx \overline{N}_c \cdot \mathrm{Var}(X_c)$, under specified limit, which agrees with well-known *sum of variance law* that holds in this particular scenario.

For lower-priority traffic, the waiting time of a low priority packet can be expressed as

$$W_L = \sum_{j=1}^{N_H} X_j + \sum_{j=1}^{N_L} X_j + \sum_{j=1}^{W_L \cdot \lambda_H} X_j + R. \quad (9)$$

Equation (9) essentially states that the waiting time of a low-priority packet can be decomposed into four components: 1. residual time due to service or vacation, 2. the time to serve existing high priority packets, 3. the time to serve existing low-priority packets that are ahead in the queue, and 4. the time to serve new high priority packets that arrive while the low-priority packet is waiting. Algebraic manipulations yield the following expressions for the first and second moments of the waiting time

$$\overline{W}_L = \frac{\overline{R}}{(1-\rho_H)(1-\rho_H-\rho_L)}, \quad (10)$$

and

$$\begin{aligned} \overline{W_L^2} &= q_L \cdot \mathrm{Var}(X) + q_H \cdot \mathrm{Var}(X) \\ &+ (q_H+q_L)^2 \cdot \overline{X}^2 + \overline{R^2} + q_Z \cdot \overline{R} \end{aligned} \quad (11)$$

where the coefficients q_L, q_H and q_Z are defined as

$$\begin{aligned} q_L &= \overline{N}_L, \\ q_H &= \overline{N}_H + \lambda_H \cdot \overline{W}_L, \\ q_Z &= 2\,(q_H+q_L) \cdot \overline{X}. \end{aligned}$$

The desired equations for the first two moments of the waiting time are complete. One immediate sanity check is to note that 11 reduces to 7 whenever $\lambda_H \to 0$, which agrees with expectation because in the latter case lower-priority traffic essentially becomes higher-priority traffic.

A. Maximum Hop-count

The maximum allowable hop-count that respects the QOS delay constraint is given by

$$K^* = \arg\max_K \left\{ \sum_{i=1}^{K} \overline{T}_i \le T_{QoS} \right\}, \quad (12)$$

where

$$\overline{T}_i = (1-p_t) \cdot \overline{W}_{H,i} + p_t \cdot \overline{W}_{L,i} + \sum_{i=1}^{K} \tau_i, \quad (13)$$

with $W_{H,i}$ and $W_{L,i}$ are respectively given by (10) and (11) for each $i = 1, 2, \cdots K$.

In a particular case, if all relay nodes were identical, then we obtain a simpler expression as,

$$K^* \approx \frac{T_{QoS}}{\overline{T}}, \quad (14)$$

with $\overline{T} = \overline{T}_i,\ i = 1, 2, \cdots K$.

B. Blocking Probability

Knowing the first and second moments of the random variables, we assume that the overall sum of all service times and waiting times experienced by a packet from source to sink can be approximated by a normal distribution. Such assumption is quite reasonable even if distributions of the random variables are not necessarily identical. In fact, many variants of the central limit theorem such as Lyapunov's show that the Gaussian property of the sum of random variables hold under more general conditions, including the sum of independent but not necessarily identical random variables [13]. In our case, we are interested in end-to-end delay $D = \sum_{i=1}^{K} W_i + \sum_{i=1}^{K} X_i$ for both high-priority traffic D_H and low-priority traffic D_L. Thus, we have

$$D_p \sim \mathcal{N}(\mu_p, \sigma_p) \quad (15)$$

where

$$\mu_p = \sum_{i=1}^{K} \overline{\chi}_i + \sum_{i=1}^{K} \overline{\omega}_{p,i}, \quad (16)$$

$$\sigma_p^2 = \sum_{i=1}^{K} \mathrm{Var}(\chi_i) + \sum_{i=1}^{K} \mathrm{Var}(\omega_{p,i}), \quad (17)$$

with χ_i denotes service time at node i and $\omega_{p,i}$ denotes waiting time at node i for traffic with priority $p \in \{L, H\}$.

Therefore, the final desired blocking probability is a weighted sum according to whether a data packet is scheduled in the high-priority queue or the low-priority queue:

$$\begin{aligned} \mathrm{P}[\mathrm{D} > \mathrm{T_{QoS}}] \approx\; & \frac{1}{2}(1-p_t)\left[1-\mathrm{erf}\left(\frac{1}{\sqrt{2}}\cdot\frac{\mathrm{T_{QoS}}-\mu_\mathrm{H}}{\sigma_\mathrm{H}}\right)\right] \\ +\; & \frac{1}{2}p_t\left[1-\mathrm{erf}\left(\frac{1}{\sqrt{2}}\cdot\frac{\mathrm{T_{QoS}}-\mu_\mathrm{L}}{\sigma_\mathrm{L}}\right)\right]. \quad (18) \end{aligned}$$

where $\mathrm{erf}(\cdot)$ is the error function [14, Eq.(8.250.1)].

IV. Numerical Results

To validate the above framework, different scenarios have been simulated. First, to verify correctness of the analytic expressions for $\overline{W_H}$ and $\overline{W_H^2}$ (i.e., moments of waiting time in single priority queues), we simulated a single $M/G/1$ queue with generally distributed vacation and service times. Here, we opt to test the analytic expression for arbitrary service time distributions even though service time was originally assumed to be exponentially distributed because the formulas hold, in general, for single hops since data traffic is always Poisson by assumption. A summary of test scenarios is shown in Table II. Here, the number of packets is set to $N = 10^5$ and $\lambda = 1$. As shown in Table III, simulation results match analytic expressions for the different distributions used.

TABLE II
Simulation Scenarios

Service/Vacation Time Distribution	$\overline{X}$	$\overline{X^2}$	$\overline{X^3}$	$\overline{V}$	$\overline{V^2}$	$\overline{V^3}$
Exp / Exp	$\frac{1}{3}$	$\frac{2}{9}$	$\frac{2}{9}$	1	2	6
Exp / Deterministic	$\frac{1}{3}$	$\frac{2}{9}$	$\frac{2}{9}$	1	1	1
Gaussian / Exp	$\frac{1}{3}$	$\frac{109}{900}$	$\frac{127}{2700}$	1	2	6
Deterministic / Deterministic	$\frac{1}{3}$	$\frac{1}{9}$	$\frac{1}{27}$	1	1	1

With regard to end-to-end delay, a summary of test results is shown in Table IV. As expected, the analytical model becomes more accurate as $\overline{V} \cdot (\lambda_H + \lambda_L) \to 0$, and $\lambda_C/\lambda_D \to 0$, and $K \to \infty$. In the first two cases, the Poisson property of data traffic is preserved at the specified limits, and hence queues remain $M/G/1$ so that traffic sees time averages (a.k.a PASTA). In the last case, our normal approximation to the variance of end-to-end delay becomes more accurate as K increases.

TABLE III
Basic Simulation Results

Service/Vacation Time Distribution	Analytic $\overline{W_H}$	Sim $\overline{W_H}$	Analytic $\overline{W_H^2}$	Sim $\overline{W_H^2}$
Exp / Exp	1.1667	1.171	2.2932	2.485
Exp / Det	0.6667	0.6785	0.6173	0.6858
Gauss / Exp	1.091	1.102	2.021	2.250
Det / Det	0.5833	0.5924	0.4236	0.4599

TABLE IV
End-to-End Delay Results

Test Parameters μ, $\overline{V}$, λ_D, λ_C, K	Analytic μ_T	Sim μ_T	Analytic σ_T	Sim σ_T
1, 0.1, 0.5, 0.1, 35	92.5	93.7	13.8	10.1
5, 0, 1, 0.2, 50	13.7	13.8	1.89	1.91
10, 0.2, 3, 0.1, 60	23.9	21.4	2.35	1.74
3, 0, 0.7, 0.1, 100	46.8	46.8	4.63	4.20

In order to verify that the normal approximation for large $K \gg 0$ is valid, we plotted histograms for various settings and the resulting distribution is indeed approximately normal. Two samples are depicted in Figures 4 and 5 for $K = 30$ and $K = 60$ respectively.

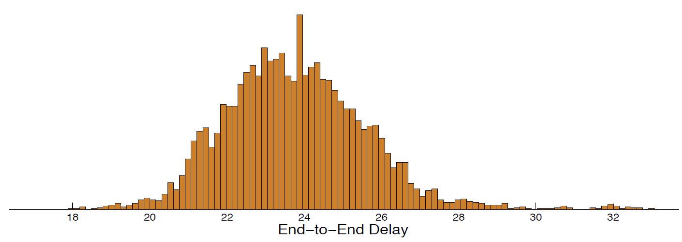

Fig. 4. Histogram of end-to-end delay for $K = 60$. End-to-end delay distribution is approximately Gaussian.

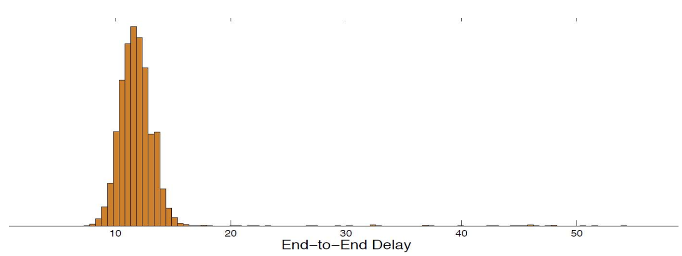

Fig. 5. Histogram of end-to-end delay for $K = 30$. End-to-end delay distribution is also approximately Gaussian.

With regard to maximum hop-count, Table V shows maximum hop-count for various settings of vacation time distribution. As shown in the table, simulation results match almost perfectly with analytical results. As stated earlier, the mean vacation time $\overline{V}$ is assumed to be small compared to mean inter-arrival time $\lambda_D + \lambda_C$. So, Table V only shows the results for $\overline{V} \cdot \lambda_D \leq 0.2$.

TABLE V
MAXIMUM HOP-COUNT VS. VACATION TIME DISTRIBUTION (SIMULATION / ANALYTICAL)

$(\lambda_D = 1, \lambda_C = 0.1, \mu = 5)$

Vacation Time Distribution	$\overline{V}$ 0	0.05	0.1	0.15	0.2
Exponential	18/18	14/15	13/13	11/11	10/11
Gaussian	18/18	17/17	15/15	14/14	13/13
Deterministic	18/18	16/17	15/15	14/14	13/13

Finally, the packet dropping probability as a function of K is plotted in Figure 6. Both simulation and analytic results are displayed for the three vacation time distributions: Exponential, Normal, and Deterministic. Here, the setting of parameters is similar to that of Table V, with a fixed mean vacation time at $\overline{V} = 0.1$. As shown in the figure, simulation results closely match analytical results. Similar results were obtained when relay hops had different settings of control traffic rate and sleeping policy.

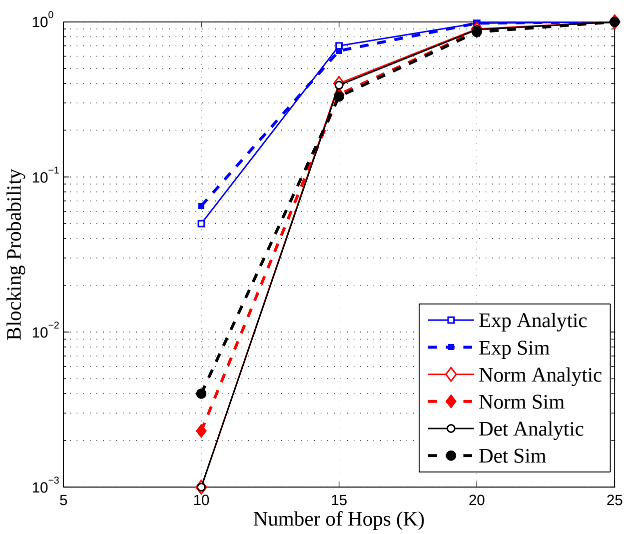

Fig. 6. Packet dropping probability vs. number of hops for three different distributions of vacation time: Exponential, Normal, and Deterministic.

V. CONCLUSION

End-to-end delay and energy consumption are the two key issues in wireless sensor networks. Since wireless sensor networks are deployed to operate for very long periods of time, it is essential to reduce energy consumption in the network nodes, giving rise to an inherent tradeoff between QoS and energy conservation in the network. In this paper, we modeled a delay-sensitive network that adopts a vacation scheme. We provided an analytical derivation for the proposed model and verified its results with numerical simulations.

There are several potential extensions to this framework. For instance, we may define the QoS by a tuple that is composed of the maximum end-to-end delay and the probability of packet blocking, and then calculate the corresponding maximum allowable hop-count beyond which the imposed QoS would be violated. In other words, for efficient relay nodes deployment, we impose a constraint on the blocking probability in the system as opposed to the mean end-to-end delay. Accordingly, we compute the maximum number of relay nodes that would deliver the sensing measurements without violating the QoS requirements with a high probability.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sunknrusubranwniam, and E. Cayirci, "A survey on sensor networks," in *IEEE Communication Magazine*, Aug. 2002, pp. 102–114.

[2] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay efficient sleep scheduling in wireless sensor networks," in *Proc. IEEE* 24^{th} *Annual Joint Conference of the IEEE Computer and Communications Societies (NFOCOM'2005)*, vol. 4, Miami, USA, Mar. 2005, pp. 2470–2481.

[3] S. K. Kanhere, M. Goudar, and V. M. Wadhai, "End-To-End delay optimization in wireless sensor network," *International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN)*, vol. 1, no. 3, pp. 2248–9738, 2012.

[4] G. Anastasi, M. Conti, M. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, May 2009.

[5] M. Nosovic and T. Todd, "Scheduled rendezvous and RFID wakeup in embedded wireless networks," in *IEEE International Conference on Communications (ICC'2002)*, vol. 5, New York, USA, May 2002, pp. 3325–3329.

[6] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proc.* 7^{th} *annual international conference on Mobile computing and networking (Mobi-Com'2001)*, Rome, Italy, Jul. 2001, pp. 272–287.

[7] J. Kim, X. Lin, N. B. Shroff, and P. Sinha, "Minimizing delay and maximizing lifetime for wireless sensor networks with anycast," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 515 –528, Apr. 2010.

[8] C.-F. Chiasserini and M. Garetto, "Modeling the performance of wireless sensor networks," in *Proceedings IEEE INFOCOM*, Hong Kong, China, Mar. 2004, pp. 220–231.

[9] C. Joo, J.-G. Choi, and N.B. Shroff, "Delay performance of scheduling with data aggregation in wireless sensor networks," in *Proceedings IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[10] R. Zhou, L. Li, and J. Zhang, "Analysis on event delay in event-driven wireless sensor networks," in *National Conference on Information Technology and Computer Science (CITCS'2012)*, Lanzhou, China, Nov. 2012, pp. 656–660.

[11] I. J. B. F. Adan and J. Resing., *Queueing theory*, eindhoven university of technology ed., 2002.

[12] B. Sinclair, "Priority M/G/1 queue with and without preemption," pp. http://www.shihada.com/F10–337/papers/mg1p.pdf, Nov. 2002.

[13] S. Zahl, "Bounds for the central limit theorem error," *SIAM Journal on Applied Mathematics*, vol. 14, no. 6, pp. 1225–1245, 1966.

[14] I. S. Gradshteyn and I. Ryzhik, *Table of Integrals, Series, and Products*, 7th ed. Amsterdam: Elsevier/Academic Press, 2007.