Error Analysis and Kernel Density Approach of Scheduling Sleeping Nodes in Cluster-Based Wireless Sensor Networks

Miao Peng, Yang Xiao, *Senior Member, IEEE*, and Pu Patrick Wang

***Abstract*—Energy consumption is an important research topic in wireless sensor networks. Putting sensor nodes to sleep is one of the most popular ways to save energy in battery-powered sensor nodes. Many existing research studies on sleeping techniques are based on preknowledge of deployment of sensor nodes, e.g., a known probability distribution of sensor nodes in a target-sensing field. Thus, whether a scheduling-sleeping scheme has good performance mostly depends on preknowledge of the deployment of sensor nodes. In this paper, we first show the discrepancy of system performance metrics, including energy consumption and network lifetime, based on inaccurate preknowledge of the deployment of sensor nodes in a cluster-based sensor network. Through analytical studies, we conclude that the discrepancy is very large and cannot be neglected. We hence propose a distribution-free approach to study energy consumption. In our approach, no assumption of the probability distribution of deployment of sensor nodes is needed. The proposed approach has yielded a good estimation of network energy consumption. Furthermore, previous studies normally assume that battery energy levels of sensor nodes are the same. However, in a real network, battery quality is different, and the energy in each sensor node is a random variable. We provide a mathematical approximation and a standard deviation study for energy consumption, as well as a more in-depth study for network lifetime under random batter energy.**

***Index Terms*—Analytical modeling, kernel density, network lifetime, performance evaluation, preknowledge, wireless sensor networks.**

I. Introduction

Wireless sensor networks have many applications, such as pollutant detection, military sensing and tracking, medical emergency response, etc. In most cases, a tiny battery-powered sensor node usually has three operations, i.e., sensing, communication, and computation, which easily deplete the battery. In addition, sensor nodes are sometime deployed in hostile environments, and hence, recharging batteries of sensor nodes is by no means a trivial task and is often even unfeasible. A great deal of research has recently been done to promote the efficient use of energy in sensor nodes. Some of these studies develop new uses of energy in terms of routing algorithms [1]–[5]. Others provide sleeping techniques [9]–[11], [16] to save energy and extend network lifetime. In these techniques, some sensor nodes are put in sleep mode, whereas other sensor nodes are in active mode for sensing and communication tasks. When a sensor node is in sleep mode, it shuts down all functions, except that a low-power timer is on to wake itself up at a later time [10], and therefore, it consumes only a tiny fraction of the energy consumed in the active mode [6]. Based on preknowledge of deployment of sensor nodes, many sleeping-scheduling schemes are proposed. These schemes perform well when saving energy [7].

In previous works, system performance based on an assumption that the sensor deployment distribution is known has been evaluated. The major disadvantages of this traditional analysis are listed as follows: First, it is very difficult to choose an accurate deployment distribution, and if the assumed distribution is wrong, inaccurate analysis and protocols/algorithms may be produced. Second, if the deployment distribution of sensor nodes changes, system performance will also change, and even the entire analysis may no longer be valid.

In this paper, we show the discrepancy of system performance metrics, including energy consumption and network lifetime, based on inaccurate preknowledge of the deployment of sensor nodes in a cluster-based sensor network. We propose a distribution-free approach to study energy consumption. In our approach, no assumption of the probability distribution of deployment of sensor nodes is needed. The proposed approach has yielded a good estimation of network energy consumption. Furthermore, previous studies normally assume that battery energy levels of sensor nodes are the same. However, in a real network, battery quality is different, and the battery energy in each sensor node is a random variable. We provide a mathematical approximation and a standard deviation study for energy consumption, as well as a more in-depth study for network lifetime under random battery energy. We adopt network energy consumption in [7] as an example to verify our ideas. However, our approach can easily be extended to solve other problems.

The rest of this paper is organized as follows: In Section II, we review some background. In Section III, we provide an error analysis with different distributions via a mathematical analysis. A standard deviation study and a network lifetime

Manuscript received September 24, 2008; revised March 12, 2009. First published July 21, 2009; current version published November 11, 2009. This work was supported in part by the U.S. National Science Foundation under Grant CCF-0829827, Grant CNS-0716211, and Grant CNS-0737325. The review of this paper was coordinated by Dr. J. Misic.

M. Peng and Y. Xiao are with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290 USA (e-mail: mpeng@ua.edu; yangxiao@ieee.org).

P. P. Wang is with the Department of Mathematics, The University of Alabama, Tuscaloosa, AL 35487-0290 USA (e-mail: pwang@bama.ua.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2009.2027908

0018-9545/$26.00 © 2009 IEEE

Authorized licensed use limited to: UNIV OF ALABAMA-TUSCALOOSA. Downloaded on November 17, 2009 at 14:06 from IEEE Xplore. Restrictions apply.

study are presented in Section IV. In Section V, we propose the distribution-free method and present some numerical results. Finally, we conclude this paper in Section VI.

II. Background

This section summarizes a randomized scheduling (RS) scheme [7], in which nodes are randomly selected to sleep in high-density cluster-based sensor networks. In other words, each sensor node is selected by the cluster head with the probability β under following assumptions: 1) Each sensor node belongs to the same cluster throughout its lifetime. 2) Nodes are randomly distributed as a 2-D Poisson point process with density ρ. In other words, the probability of finding n nodes in a region of area A is equal to $(\rho A)^n \exp(-\rho A)/n!$. Furthermore, these n nodes in area A are uniformly distributed. 3) The maximum transmission range of the cluster head is denoted by R, and there are n sensor nodes in the cluster. The cluster covers a circular geographic area of πR^2 with the cluster head at the center. The cluster head plans to allow, on average, $n \cdot \beta$ $(\beta < 1)$ nodes to sleep in each cycle.

A. Energy Consumption

The energy consumption rate is defined as the energy consumed per second when the sensor is active and is generally a positive convex function of the distance between the sensor node and the head of cluster $E_{\text{active}}(x) = C(x) + K$, where K is a positive constant, and $C(x)$ is a nonnegative convex function. In [7], the authors used a power function as

$$E_{\text{active}}(x) = \lambda \cdot k_1 \cdot [\max(x_{\min}, x)]^{\gamma} + k_2 \tag{1}$$

where λ denotes the average packet transmission rate per second of each sensor node, x is the distance between the sensor node and the cluster head, k_1 is the constant corresponding to energy consumption due to transmission of each packet, k_2 is the idle/receive energy consumption per second, $x_{\min}$ is the minimum allowable transmission range corresponding to the minimum allowable transmission energy, and $\gamma \geq 2$ is the path-loss exponent. From [7], the expected energy consumption of each node during a second in the RS scheme is computed as

$$E = \int_0^R (1-\beta) \cdot f(x) \cdot E_{\text{active}}(x) \cdot dx \tag{2}$$

where $f(x)$ is the probability density function (pdf) of the distance x between a sensor and the cluster head. Because it is assumed that sensor nodes are uniformly distributed in the circular coverage area of the cluster, based on [7], $f(x)$ is

$$f(x) = \frac{\partial F}{\partial x} = \frac{\partial\left[\Pr(X \leq x)\right]}{\partial x} = \frac{\partial}{\partial x}\left[\frac{\pi x^2}{\pi R^2}\right] = \frac{2x}{R^2} \tag{3}$$

where $0 \leq x \leq R$. According to the assumption that the number of sensor nodes is distributed according to a 2-D Poisson point process with expected density ρ, from [7], the average E over all possible numbers of nodes in a cluster is

$$E^{(\text{overall})} = \sum_{n=0}^{\infty} nE \frac{(\rho\pi R^2)^n}{n!} \cdot e^{-\rho\pi R^2} = E \cdot \rho\pi R^2. \tag{4}$$

B. Network Lifetime

Based on [7], the network lifetime $T(\beta_d)$ is defined as the time when a fraction of sensor nodes β_d run out of energy. Let Ψ be the total battery energy that each sensor node carries when the sensor network is initialized. In the RS scheme, the time when β_d fraction of nodes run out of battery life is the time when sensor nodes with $x \geq x_d$ all run out of battery life. From [7], x_d satisfies

$$\beta_d = \int_{x_d}^{R} f(x)\, dx = \frac{R^2 - [x_d]^2}{R^2}. \tag{5}$$

The network lifetime of the RS scheme is

$$T(\beta_d) = \frac{\Psi}{E(x_d)} = \frac{\Psi}{(1-\beta)\left\{\lambda \cdot k_1 \cdot [\max(x_{\min}, x_d)]^{\gamma} + k_2\right\}}. \tag{6}$$

III. Discrepancy Analysis

System performance evaluations are always based on a certain set of assumptions. However, those assumptions may not exactly be held in real-world systems. For example, in the RS scheme [7], all the conclusions for the scheme are based on the assumption that sensor nodes are independently and uniformly distributed in each cluster. In fact, deployment of sensor nodes is impacted by many factors such as weather, terrain, and so on. Thus, locations of sensor nodes do not necessarily follow a uniform distribution or other distributions that researchers may choose. In this section, we will present the error analysis when the assumptions are different. For simplicity, we give some notations. Let $\tilde{E}^{(\text{overall})}$ and $\tilde{T}(\beta_d)$ denote the overall expected energy consumption and the network lifetime in a cluster derived from real-world sensor node distribution data, respectively. Thus, their discrepancy can be given by

$$E_{\text{error}} = \left| E^{(\text{overall})} - \tilde{E}^{(\text{overall})} \right| \tag{7}$$

$$T_{\text{error}} = \left| T(\beta_d) - \tilde{T}(\beta_d) \right|. \tag{8}$$

To show the discrepancy in system performance generated by assumptions, based on [7], we first assume that sensor nodes are still randomly distributed as a 2-D Poisson point process with density ρ. That is, the probability of finding n nodes in a region of area A is equal to $(\rho A)^n \exp(-\rho A)/n!$. However, n nodes in area A follow the 2-D Gaussian distribution. We assume that the deployment region of sensor nodes in a cluster is modeled in a 2-D Cartesian coordination system and the cluster head located at point (0, 0). Then, we present the performance derived from this new assumption. Finally, the performance from two assumptions is compared by several figures. In the following, we give the performance error analysis.

Authorized licensed use limited to: UNIV OF ALABAMA-TUSCALOOSA. Downloaded on November 17, 2009 at 14:06 from IEEE Xplore. Restrictions apply.

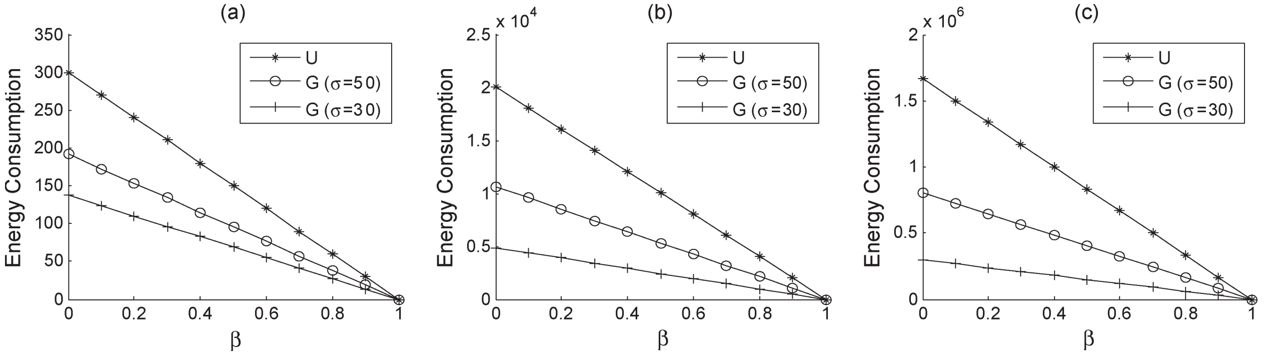


Fig. 1. Energy consumption comparison ($\beta =$ fraction of sensor nodes allowed to sleep; U $=$ uniform distribution; G $=$ Gaussian distribution). (a) $\gamma = 2$. (b) $\gamma = 3$. (c) $\gamma = 4$.

A. Energy Consumption

According (1) and (2), the expected energy consumption of each node during a second is

$$\tilde{E} = \int_0^R (1-\beta) \cdot \tilde{f}(x) \cdot E_{\text{active}}(x) \cdot dx \qquad (9)$$

where $\tilde{f}(x)$ is the pdf of the distance X between a sensor and the cluster head. It is clear that X is a random variable, and $\Pr(X \leq x)$ denotes the probability that the distance between a sensor and the cluster head is less than or equal to x. Since nodes follow the 2-D Gaussian distribution in each cluster, we have

$$\tilde{F}(x) = \Pr(X \leq x) = \frac{1}{2\pi\sigma^2} \iint_{i^2+j^2 \leq x^2} e^{-\frac{i^2+j^2}{2\sigma^2}}\, di\, dj. \qquad (10)$$

Let $i = r \sin\theta$ and $j = r\cos\theta$, where $0 \leq r \leq x$, and $0 \leq \theta \leq 2\pi$. Thus, we have

$$\tilde{F}(x) = \Pr(X \leq x) = \frac{1}{2\pi\sigma^2} \int_0^{2\pi} \int_0^{x} e^{-\frac{r^2}{2\sigma^2}} |J|\, dr\, d\theta \qquad (11)$$

where

$$|J| = \begin{vmatrix} \frac{\partial i}{\partial r} & \frac{\partial i}{\partial \theta} \\ \frac{\partial j}{\partial r} & \frac{\partial j}{\partial \theta} \end{vmatrix} = r.$$

Thus, we have

$$\tilde{f}(x) = \frac{\partial \tilde{F}(x)}{\partial x} = \frac{\partial\left[\Pr(X \leq x)\right]}{\partial x} = e^{-\frac{x^2}{2\sigma^2}} \cdot \frac{x}{\sigma^2}. \qquad (12)$$

Therefore, for the expected energy consumption of each node during a second, we have

$$\tilde{E} = \lambda k_1 (1-\beta) \left[x_{\min}^{\gamma} \left(1 - e^{-\frac{x_{\min}^2}{2\sigma^2}}\right) + \int_{x_{\min}}^{R} e^{-\frac{x^2}{2\sigma^2}} \frac{x^{\gamma+1}}{\sigma^2} dx \right] + k_2(1-\beta)\left(1 - e^{-\frac{R^2}{2\sigma^2}}\right). \qquad (13)$$

From (4), we have

$$\tilde{E}^{(\text{overall})} = \tilde{E} \cdot \rho\pi R^2. \qquad (14)$$

To show the error analysis, we need to choose the same parameters in [7] for the sensor network. Thus, based on [7], in the sensor network, we assume that there are $n = 500$ sensor nodes in each cluster, where we have $k_1 = 10^{-6}$ J/(frame $\cdot$ m^2), $k_2 = 0.1$ J/s, $x_{\min} = 5$ m, and $\lambda = 100$ frames/s. The maximum transmission range of the cluster head is $R = 100$.

Fig. 1(a)–(c) shows the energy consumption versus fraction of sensor nodes allowed to sleep β for both Gaussian and uniform distributions, where the standard deviation of the Gaussian distribution is 50 or 30, namely, $\sigma = 50$ or $\sigma = 30$, respectively. As illustrated in the figures, the energy consumption decreases when β increases for both Gaussian and uniform distributions. When $\beta = 0$, the energy consumption achieves the maximum value. In addition, it is easy to see that, when the fraction of sensor nodes allowed to sleep become 1, the energy consumption becomes 0 for both distributions. When β increases, each sensor node in the cluster has a higher probability to be selected to sleep, and thus, the energy consumption decreases. When $\beta = 1$, this means that all sensor nodes are elected to sleep, and thus, the energy consumption is 0. In Fig. 1(a)–(c), we observe that, when the path-loss exponent increases, the energy consumption for both distributions quickly increases. When β increases, the discrepancy between two distributions decreases until it reaches zero, because when β increases, each sensor node in the cluster has a higher probability of being selected to sleep, and thus, the energy consumption decreases. When $\beta = 1$, this means that all sensor nodes are elected to sleep, and thus, for both distributions, the energy consumption is 0.

Comparing Fig. 1(a)–(c), we find that, under the same parameters, when the path-loss exponent increases, the discrepancy becomes larger. As illustrated in Fig. 1, when the standard deviation of the Gaussian distribution is 30, the discrepancy is larger than when the standard deviation of the Gaussian distribution is 50. This fact shows that the deployment of sensor nodes in [7] is more similar with the Gaussian distribution with $\sigma = 50$ than the Gaussian distribution with $\sigma = 30$. Fig. 1 also shows that the discrepancy of energy consumption is pretty large under different assumptions of deployment distributions of sensor nodes.

Authorized licensed use limited to: UNIV OF ALABAMA-TUSCALOOSA. Downloaded on November 17, 2009 at 14:06 from IEEE Xplore. Restrictions apply.

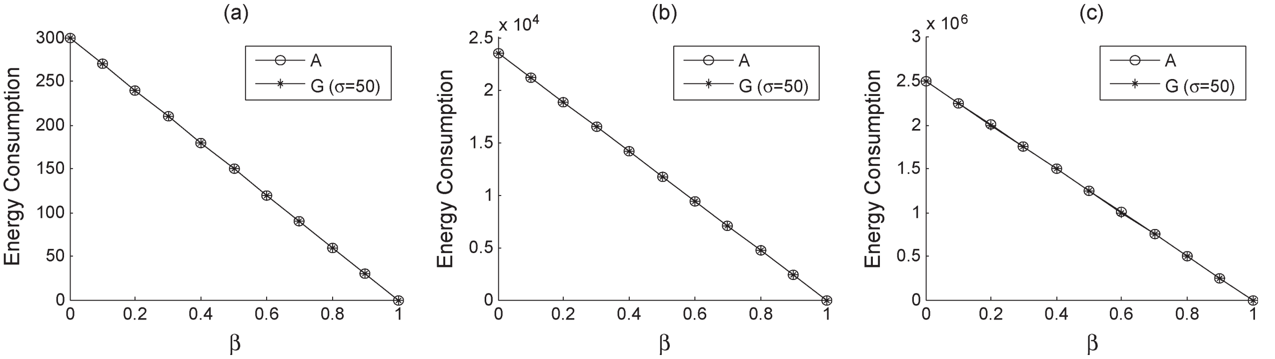


Fig. 2. Approximation of energy consumption ($\beta =$ fraction of sensor nodes allowed to sleep; A = approximation; G = Gaussian distribution). (a) $\gamma = 2$. (b) $\gamma = 3$. (c) $\gamma = 4$.

B. Approximation

As aforementioned, we can obtain the expected energy consumption of each node during a second by (13). When the coverage radius of the cluster R is large and the minimum transmission range $x_{\min}$ is small, we can get an approximation expression of the expected energy consumption of each node in a second. First, when R is large and $x_{\min}$ is small, we have

$$\int_{x_{\min}}^{R} e^{-\frac{x^2}{2\sigma^2}} \cdot \frac{x^{\gamma+1}}{\sigma^2}\, dx \approx \int_{0}^{\infty} e^{-\frac{x^2}{2\sigma^2}} \cdot \frac{x^{\gamma+1}}{\sigma^2}\, dx. \qquad (15)$$

Then, we do some mathematical transformations for (15) as follows:

$$\begin{aligned}\int_0^\infty e^{-\frac{x^2}{2\sigma^2}} \frac{x^{\gamma+1}}{\sigma^2}\, dx &= \frac{1}{2\sigma^2}\int_0^\infty e^{-\frac{x^2}{2\sigma^2}} (x^2)^{\frac{\gamma}{2}}\, dx^2 \\ &= \frac{1}{2\sigma^2}\int_0^\infty e^{-\frac{y}{2\sigma^2}} (y)^{\frac{\gamma}{2}}\, dy \\ &= \frac{1}{2\sigma^2}(2\sigma^2)^{\frac{\gamma+2}{2}}\Gamma\left(\frac{\gamma+2}{2}\right) \\ &\quad \times \int_0^\infty \frac{\left(\frac{1}{2\sigma^2}\right)^{\frac{\gamma+2}{2}}}{\Gamma\left(\frac{\gamma+2}{2}\right)} (y)^{\frac{\gamma+2}{2}-1} e^{-\frac{1}{2\sigma^2}y}\, dy.\end{aligned}$$

Note that the pdf of the gamma distribution is

$$f(x) = \begin{cases} \frac{\beta^\alpha}{\Gamma(\alpha)} \cdot x^{\alpha-1} \cdot e^{-\beta x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

where the gamma function is defined as $\Gamma(z) = \int_0^\infty e^{-z} x^{z-1} dx$. Thus, we have

$$\begin{aligned}\int_0^\infty e^{-\frac{x^2}{2\sigma^2}} \frac{x^{\gamma+1}}{\sigma^2} dx &= \frac{1}{2\sigma^2}(2\sigma^2)^{\frac{\gamma+2}{2}}\Gamma\left(\frac{\gamma}{2}+1\right) \\ &= 2^{\frac{\gamma}{2}-1}\sigma^\gamma \gamma \Gamma(\gamma/2).\end{aligned}$$

Thus, (13) can be approximated by a closed-form expression, i.e.,

$$\tilde{E} \approx (1-\beta)\left(k_1 \lambda 2^{\gamma/2-1}\sigma^\gamma \gamma \Gamma(\gamma/2) + k_2\right). \qquad (16)$$

Fig. 2(a)–(c) shows the approximation of energy consumption where the coverage radius of the cluster R is 250, and the minimum transmission range $x_{\min}$ is 5. As illustrated in the figures, the approximation is effective and accurate for large R. Thus, when the coverage radius R is greater than 250, we can estimate the energy consumption by (16) instead of (13). This simplifies the calculation to avoid integration.

C. Network Lifetime

From [7], the network lifetime parameter is β_d, which represent the fraction of sensor nodes running out of energy.

Based on (5), we have $\beta_d = \int_{x_d}^{R} \tilde{f}(x) dx$, where $\tilde{f}(x) = e^{-(x^2/2\sigma^2)} \cdot x/\sigma^2$. Thus, $\beta_d = \exp(-[x_d]^2/2\sigma^2) - \exp(-R^2/2\sigma^2)$, leading to $x_d = \sqrt{-2\sigma^2 \cdot \ln(\beta_d + \exp(-R^2/2\sigma^2))}$. Based on (6), we have

$$\tilde{T}(\beta_d) = \frac{\Psi}{\tilde{E}(x_d)} = \frac{\Psi}{(1-\beta)\left\{\lambda \cdot k_1 \cdot [\max(x_{\min}, x_d)]^\gamma + k_2\right\}}.$$

Fig. 3(a)–(c) shows the network lifetime versus fraction of sensor nodes allowed to sleep β for both uniform and Gaussian distributions, where the standard deviation of the Gaussian distribution is 50, namely, $\sigma = 50$, and the battery energy is $\Psi = 10^3$ J. In addition, we assume that, when a fraction of sensor nodes $\beta_d = 0.5$ run out of energy, the network will die. As illustrated in the figures, the network lifetime increases when β increases for both Gaussian and uniform distributions. Because when β increases, each sensor node has a higher probability to be selected to sleep, the energy consumption rate of each sensor node decreases, which means that the network lifetime increases. When $\beta = 0$, each sensor node works all the time, and its energy consumption rate is the largest, and thus, the network lifetime achieves the minimum value. In addition, it is easy to see that, when the fraction of sensors allowed to sleep become 1, the network lifetime becomes high for both distributions. In Fig. 3(a)–(c), we can see that, when the path-loss exponent increases, the network lifetime for both distributions quickly decreases. When β increases, the discrepancy between two distributions quickly increases. Comparing Fig. 3(a)–(c), we find that, under the same parameters, when the path-loss exponent increases, the discrepancy becomes smaller.

Authorized licensed use limited to: UNIV OF ALABAMA-TUSCALOOSA. Downloaded on November 17, 2009 at 14:06 from IEEE Xplore. Restrictions apply.

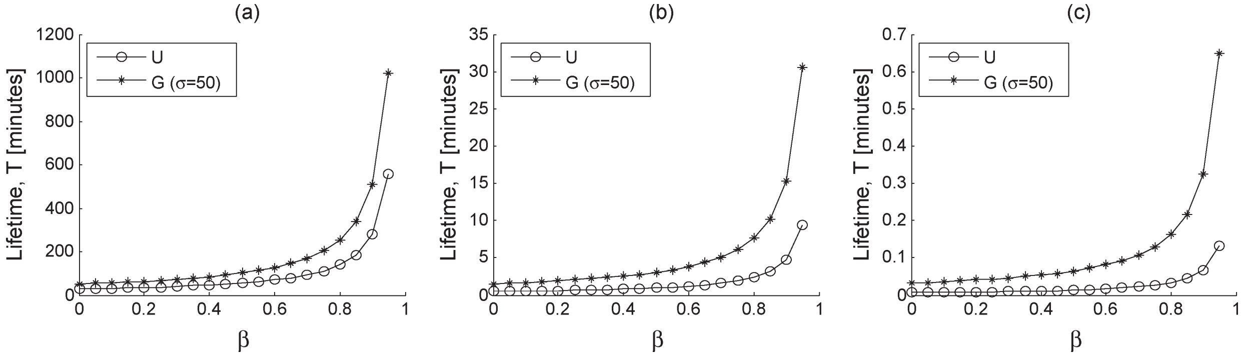


Fig. 3. Network lifetime comparison ($\beta =$ fraction of sensor nodes allowed to sleep; U = uniform distribution; G = Gaussian distribution). (a) $\gamma = 2$. (b) $\gamma = 3$. (c) $\gamma = 4$.

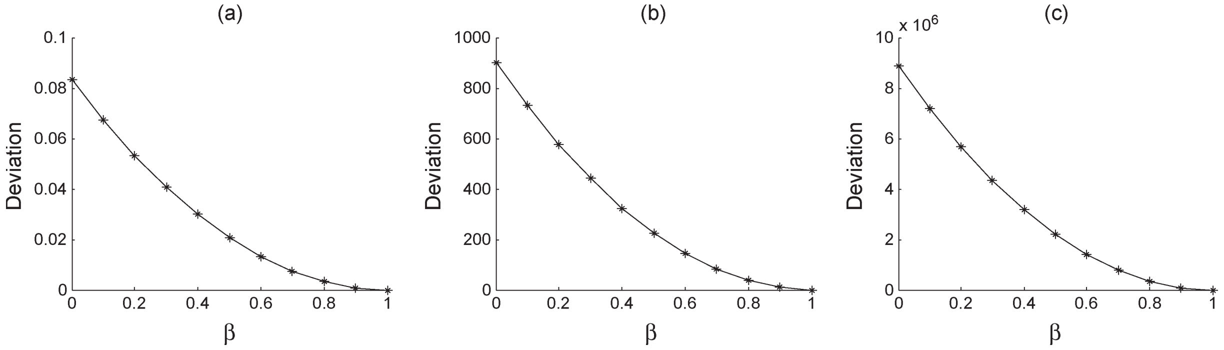


Fig. 4. Deviation of energy consumption ($\beta =$ fraction of sensor nodes allowed to sleep). (a) $\gamma = 2$. (b) $\gamma = 3$. (c) $\gamma = 4$.

In Fig. 3, we can observe that the discrepancy of network lifetime between two distributions is not large. This is because, compared with the battery energy $\Psi = 10^3$ J, the energy consumption rate of each sensor node is too large. For example, in Fig. 1(c), the order of magnitude of the energy consumption rate is 10^6, and thus, the order of magnitude of the network lifetime is $\Psi/10^6 = 10^{-3}$. Thus, even if the discrepancy of the energy consumption rates between two distributions is large, the discrepancy of the network lifetime between two distributions can be small. However, if we set a proper value for battery energy, we can still see the significant discrepancy in network lifetime between two distributions.

IV. Studies of the Standard Deviation of Energy Levels and Network Lifetime

A. Standard Deviation

When we discuss the system performance, we usually consider the expected value of system performance. For example, in [7] or in the previous sections of this paper, when referring to the energy consumption rate, we actually computed the expected value of the energy consumption rate of one sensor node in a cluster. The energy consumption of each node during a second is a random variable, which is denoted by $E(X)$. In the RS scheme, the energy consumption $E(X)$ can be expressed as

$$\begin{aligned} E(X) &= (1-\beta)E_{\text{active}}(X) \\ &= (1-\beta)\left(\lambda \cdot k_1 \cdot [\max(x_{\min}, X)]^{\gamma} + k_2\right) \end{aligned}$$

where X is random variable, which denotes the distance between a sensor and the cluster head. Thus, the standard deviation of the energy consumption rate can be computed as $\text{Var}(E) = \int_0^R [E(x) - E]^2 f(x)dx$, where $f(x)$ is the pdf of X, and $E = \int_0^R (1-\beta)f(x)E_{\text{active}}(x)dx$, as given in (2), i.e.,

$$\text{Var}(E) = \int\limits_0^R \left((1-\beta)E_{\text{active}}(x)\right)^2 f(x)\,dx - \left(\int\limits_0^R (1-\beta)f(x)E_{\text{active}}(x)\,dx\right)^2. \quad (17)$$

Thus, we can compute the standard deviation of energy consumption whether the deployment distribution is uniform or Gaussian. Based on the results of the standard deviation, we can further estimate the energy consumption of the network.

Fig. 4 shows the deviation of energy consumption versus fraction of sensor nodes allowed to sleep β when the deployment of sensor nodes is a uniform distribution. The variance of the energy-consumption rate decreases when β increases for all path-loss parameters. When the path-loss parameter increases, the variance sharply increases.

B. Study of Network Lifetime Under Random Battery Energy

In [7], when referring to network lifetime, it actually implied an assumption that the battery energy that each sensor node carries when the sensor network is initialized is the same. However, in the real world, we know that the quality of the battery

Authorized licensed use limited to: UNIV OF ALABAMA-TUSCALOOSA. Downloaded on November 17, 2009 at 14:06 from IEEE Xplore. Restrictions apply.

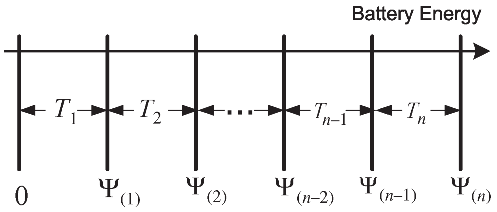


Fig. 5. Order statistics of battery energy.

is different and that the energy that each battery contains is random. In this section, we will reconsider the network lifetime where the impact of random battery energy is included. We still assume that n sensor nodes are deployed in each cluster. Thus, the total battery energy of n sensor nodes can be denoted by $\Psi_1, \Psi_2, \ldots, \Psi_n$, where Ψ_i $(i = 1, 2, \ldots, n)$ are random variables. The order statistics $\Psi_{(1)}, \Psi_{(2)}, \ldots, \Psi_{(n)}$ are defined by sorting the realizations of $\Psi_1, \Psi_2, \ldots, \Psi_n$ in increasing order. Let $f(x)$ and $F(x)$ be the pdf and the cumulative distribution function of Ψ_i, $i = 1, 2, \ldots, n$, respectively. According to [14], the probability density of the kth statistic can be given as follows:

$$\begin{aligned} f_{\Psi_{(k)}}(x) &= \frac{d}{dx} F_{\Psi_{(k)}}(x) = \frac{d}{dx} P(\Psi_{(k)} \leq x) \\ &= \frac{d}{dx} P \text{ (at least } k \text{ batteries whose energy} \leq x) \\ &= \frac{d}{dx} \sum_{j=k}^{n} \binom{n}{j} P(\Psi_1 \leq x)^j \left(1 - P(\Psi_1 \leq x)\right)^{n-j} \\ &= \frac{d}{dx} \sum_{j=k}^{n} \binom{n}{j} F(x)^j \left(1 - F(x)\right)^{n-j} \\ &= \frac{n!}{(k-1)!(n-k)!} F(x)^{k-1} \left(1 - F(x)\right)^{n-k} f(x). \end{aligned}$$

Based on [7], the network lifetime $T(\beta_d)$ is defined as the time when a fraction of sensor nodes β_d run out of energy. Thus, in this method, the time when a β_d fraction of nodes run out of battery life is the time when the sensor who carries $\Psi_{(k)}$ when the sensor network is initialized runs out of battery life, where $k = \lfloor n\beta_d \rfloor$. Here, we assume that each sensor in the cluster has the same energy consumption rate E. Thus, the expected lifetime of the node that carries $\Psi_{(k)}$ energy is given as follows:

$$T(\beta_d) = \int_0^{\infty} x f_{\Psi_{(k)}}(x) \Big/ E \cdot dx \tag{18}$$

where E is given by (2).

In this section, we assume that the initial battery energy levels of sensor nodes follow an exponential distribution. In other words, $\Psi_1, \Psi_2, \ldots, \Psi_n$ are independent identically distributed random variables following an exponential distribution with mean value Ψ. An important property of an exponential distribution is that it is memoryless: $P\{X > s + t | X > t\} = P\{X > s\}$, $s, t \geq 0$. Based on this property, we can transform the general expected lifetime equation (18) to a simple form.

As illustrated in Fig. 5, $T_i = \Psi_{(i)} - \Psi_{(i-1)}$, where $\Psi_{(1)}, \Psi_{(2)}, \ldots, \Psi_{(n)}$ are the order statistics of $\Psi_1, \Psi_2, \ldots, \Psi_n$.

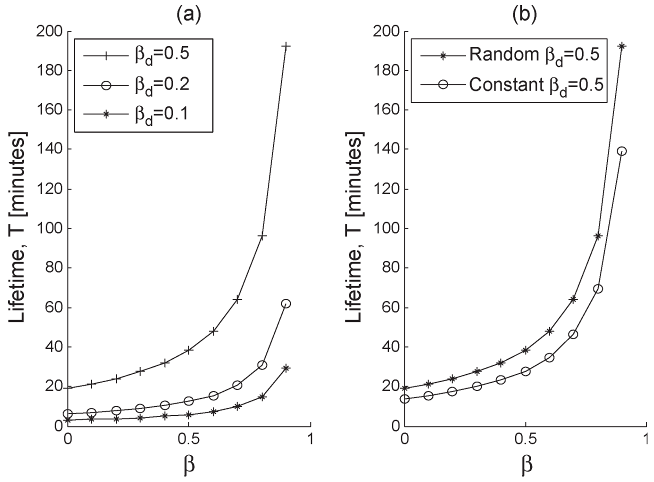


Fig. 6. Network lifetime.

Because of the memoryless property of the exponential distribution, we have the following facts [15]: The random variable T_i follows the exponential distribution $\text{Exp}((n - i + 1)/\Psi)$, where $i = 1, 2, \ldots, n$. Thus, the expected value of T_i is $\Psi/(n - i + 1)$. Based on the equation that $\Psi_{(i)} = \sum_{j=1}^{i} T_j$, the expected value of $\Psi_{(i)}$ is given by $\sum_{j=1}^{i} (\Psi/(n - j + 1))$. Thus, the expected lifetime of the network is given by

$$T(\beta_d) = \left(\sum_{j=1}^{k} \frac{\Psi}{(n - j + 1)} \right) \Bigg/ E \tag{19}$$

where $k = \lfloor n\beta_d \rfloor$.

Fig. 6(a) shows the network lifetime versus β (fraction of sensor nodes allowed to sleep) for a uniform distribution. We adopt the same parameters of the network model in Section III-C. The network lifetime improves as β increases for all the values of the parameter β_d due to energy saving by increasing the portion of sleep sensor nodes.

Fig. 6(b) compares two network lifetime cases: One is under the assumption that every sensor in the network carries the same battery energy, and the other is under the assumption that every sensor in the network carries random battery energy, which follows an exponential distribution. As illustrated in Fig. 6(b), when sensor nodes carry random battery energy, the network lifetime is longer than the case where sensors carry the same energy. In addition, the network lifetime with random battery energy can better reflect the network lifetime in the real world.

V. Distribution Free

From the discussions in previous sections, we concluded that, when the assumed sensor deployment distribution is far from reality, the discrepancy of the system performance is great and cannot be neglected. In this section, we propose a statistical approach, which is called kernel density estimation (KDE), to estimate the performance of the network where no assumption on the deployment distribution is made.

Authorized licensed use limited to: UNIV OF ALABAMA-TUSCALOOSA. Downloaded on November 17, 2009 at 14:06 from IEEE Xplore. Restrictions apply.

We first introduce KDE and then present a mathematical analysis of estimated energy consumption based on the KDE method. KDE belongs to a class of estimation, which is called the *nonparametric* density estimator. In comparison with parametric estimators, where the estimator has assumed a known distribution function and the parameters of this function (e.g., mean and variance) are the only information that we need to explore, nonparametric estimation has no assumed known distribution function and depends on all the data points to reach an estimate.

Suppose that we have random samples $X_1, \ldots, X_n$ from observations. From [8], the estimated density at any point x is

$$\hat{f}_n(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

where $K(\,)$ is the kernel density function, and h is called the window width.

In this paper, we consider a 2-D case, where $X_i = (X_{i1}, X_{i2})^T$. Thus, the estimation function is expressed as

$$\hat{f}_n(x, y) = \frac{1}{nh_1h_2}\sum_{i=1}^{n} K\left(\frac{x - X_{i1}}{h_1}, \frac{y - X_{i2}}{h_2}\right).$$

Now, we focus on a cluster to give the performance analysis. When a cluster head is found, a coordinate frame is established, where the location of the cluster head is (0, 0). Thus, each location in this cluster can be given by a math coordinate. First, we collect n_0 sensor samples from the same cluster, and their coordinates are denoted by $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{n_0}, Y_{n_0})$. Then, the estimated pdf of deployment of sensor nodes in a cluster is given by

$$\hat{f}_{n_0}(k, l) = \frac{1}{n_0h_1h_2}\sum_{i=1}^{n_0} K\left(\frac{k - X_i}{h_1}, \frac{l - Y_i}{h_2}\right) \tag{20}$$

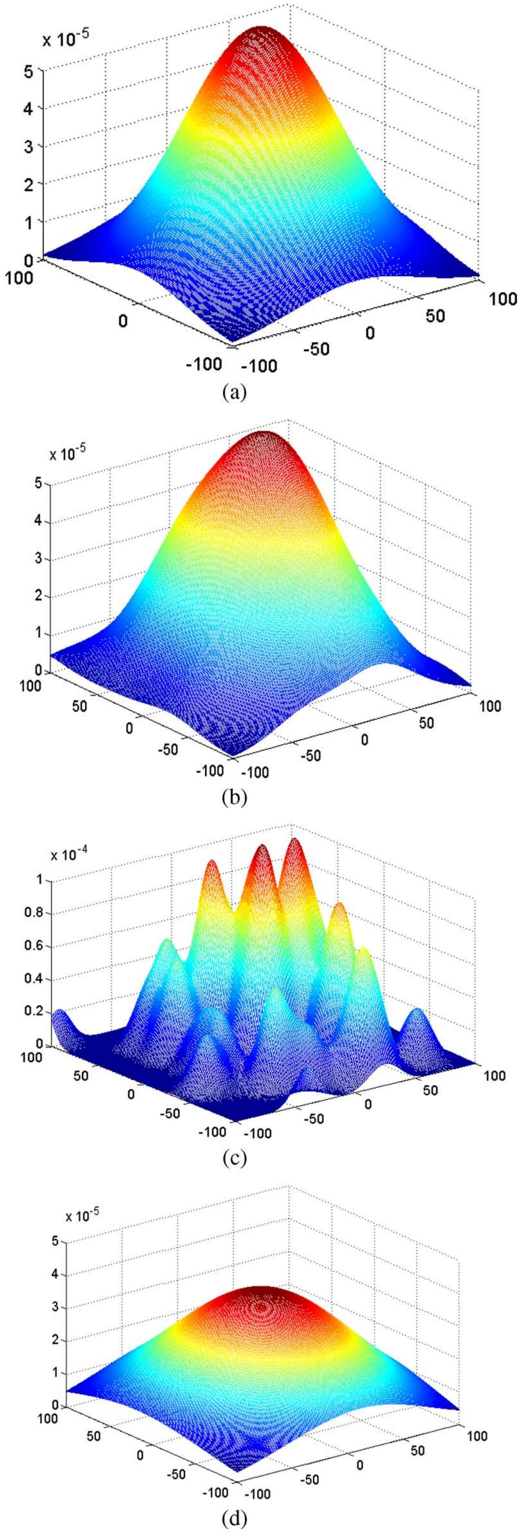
where the kernel density function is chosen as the 2-D Gaussian density function, namely, $K(u, v) = (1/2\pi)e^{-(1/2)(u^2+v^2)}$, because the Gaussian kernel function is the mostly used and powerful kernel function in the KDE method. Thus, we have

$$\hat{f}_{n_0}(k, l) = \frac{1}{2\pi n_0h_1h_2}\sum_{i=1}^{n_0} \exp\left(-\frac{(k - X_i)^2}{2h_1^2} - \frac{(l - Y_i)^2}{2h_2^2}\right). \tag{21}$$

Next, we need to derive the pdf $\hat{f}(x)$ of the distance x between a sensor and the cluster head under the estimated sensor distribution. Based on (10), the probability distribution function of the distance x between a sensor and the cluster head is

$$\begin{aligned}
&\widehat{F}(x)\\
&= \Pr(\widehat{X} \le x) = \iint\limits_{k^2+l^2\le x^2} \hat{f}_{n_0}(k, l)\,dk\,dl\\
&= \iint\limits_{k^2+l^2\le x^2} \frac{1}{2\pi n_0h_1h_2}\sum_{i=1}^{n_0}\exp\left(-\frac{(k - X_i)^2}{2h_1^2} - \frac{(l - Y_i)^2}{2h_2^2}\right)dkdl\\
&= \frac{1}{2\pi nh_1h_2}\sum_{i=1}^{n_0}\iint\limits_{k^2+l^2\le x^2}\exp\left(-\frac{(k - X_i)^2}{2h_1^2} - \frac{(l - Y_i)^2}{2h_2^2}\right)dkdl.
\end{aligned}$$

We still make use of integral transformation to solve our problem. Let $i = r\,\sin\,\theta$ and $j = r\,\cos\,\theta$, where $0 \le r \le x$,



(a)

(b)

(c)

(d)

Fig. 7. Impact of the window width h in the kernel distribution estimation. (a) Gaussian distribution. (b) Estimate distribution ($h = (27, 27)$). (c) Estimate distribution ($h = (10, 10)$). (d) Estimate distribution ($h = (50, 50)$).

Authorized licensed use limited to: UNIV OF ALABAMA-TUSCALOOSA. Downloaded on November 17, 2009 at 14:06 from IEEE Xplore. Restrictions apply.

and $0 \le \theta \le 2\pi$. Thus, we have

$$\begin{aligned}\tilde{F}(x) &\\ &= \Pr(\widehat{X} \le x)\\ &= \frac{1}{2\pi n_0 h_1 h_2}\sum_{i=1}^{n_0}\int_0^{2\pi}\int_0^{x}\exp\left(-\frac{(r\sin\theta - X_i)^2}{2h_1^2} - \frac{(r\cos\theta - Y_i)^2}{2h_2^2}\right)\\ &\quad \times |J| dr\, d\theta\end{aligned}$$

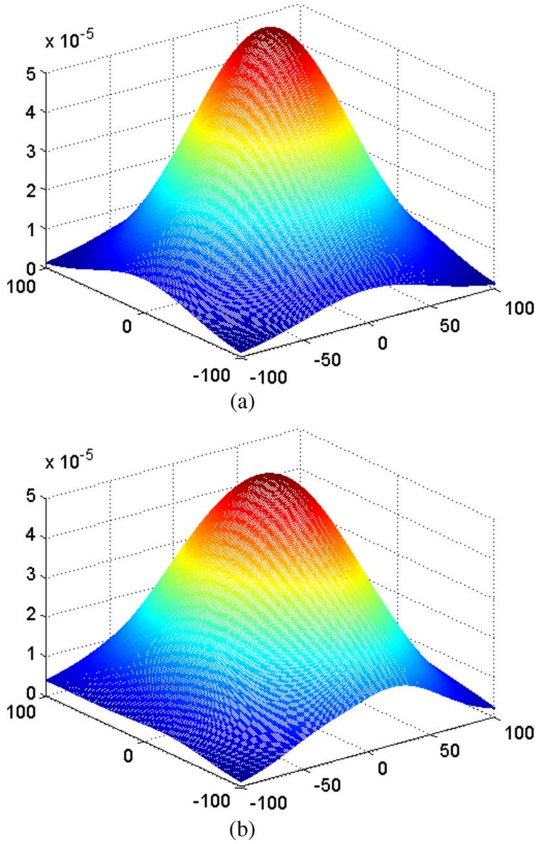
where

$$|J| = \begin{vmatrix} \frac{\partial i}{\partial r} & \frac{\partial i}{\partial \theta} \\ \frac{\partial j}{\partial r} & \frac{\partial j}{\partial \theta}\end{vmatrix} = r.$$

Thus, the pdf $\hat{f}(x)$ is given by (22), shown at the bottom of the page.

Based on (2), the expected energy consumption of each node during a second derived from KDE is as in (23), shown at the bottom of the page.

After giving the mathematical formula of energy consumption, we present how to implement the KDE method to estimate the system performance in the real world. Here, to test our method, we assume that, after deployment, the sensor locations follow the 2-D Gaussian distribution in the real world, rather than the uniform distribution, which the author assumed in [7]. Thus, according to the Gaussian distribution, sample location data can randomly be generated. Then, the estimation distribution can be obtained based on sample data. Finally,



(a)

(b)

Fig. 8. Automatic generation of the window width h. (a) Gaussian distribution. (b) Estimate distribution ($h = (35, 30)$).

$$\begin{aligned}\tilde{f}(x) = \frac{\partial \tilde{F}(x)}{\partial x} &= \frac{\partial\left[\Pr(\widehat{X} \le x)\right]}{\partial x}\\ &= \frac{1}{2\pi n_0 h_1 h_2}\sum_{i=1}^{n_0}\int_0^{2\pi}\exp\left(-\frac{(x\,\sin\,\theta - X_i)^2}{2h_1^2} - \frac{(x\,\cos\,\theta - Y_i)^2}{2h_2^2}\right)\cdot x\,d\theta\\ &= \frac{1}{2\pi n_0 h_1 h_2}\sum_{i=1}^{n_0}\int_0^{2\pi}\exp\left(-\frac{(x\,\sin\,\theta)^2 - 2x(X_i\sin\,\theta) + X_i^2}{2h_1^2} - \frac{(x\,\cos\,\theta)^2 - 2x(Y_i\cos\,\theta) + Y_i^2}{2h_2^2}\right)\cdot x\,d\theta \qquad (22)\end{aligned}$$

$$\begin{aligned}\widehat{E} &= \int_0^R (1-\beta)\cdot\tilde{f}(x)\cdot E_{\text{active}}(x)\cdot dx = \int_0^R (1-\beta)\cdot\tilde{f}(x)\cdot\left(\lambda k_1\left[\max(x_{\min}, x)\right]^\gamma + k_2\right)dx\\ &= \frac{(1-\beta)\left(\lambda k_1 x_{\min}^\gamma + k_2\right)}{2\pi n h_1 h_2}\sum_{i=1}^{n_0}\int_0^{x_{\min}}\int_0^{2\pi}\exp\left(-\frac{(x\,\sin\,\theta)^2 - 2x(X_i\sin\,\theta) + X_i^2}{2h_1^2} - \frac{(x\,\cos\,\theta)^2 - 2x(Y_i\cos\,\theta) + Y_i^2}{2h_2^2}\right)\cdot x\,d\theta\,dx\\ &\quad + \frac{(1-\beta)\lambda k_1}{2\pi n h_1 h_2}\sum_{i=1}^{n}\int_{x_{\min}}^{R}\int_0^{2\pi}\exp\left(-\frac{(x\,\sin\,\theta)^2 - 2x(X_i\sin\,\theta) + X_i^2}{2h_1^2} - \frac{(x\,\cos\,\theta)^2 - 2x(Y_i\cos\,\theta) + Y_i^2}{2h_2^2}\right)\cdot x^{\gamma+1}d\theta\,dx\\ &\quad + \frac{(1-\beta)k_2}{2\pi n h_1 h_2}\sum_{i=1}^{n_0}\int_{x_{\min}}^{R}\int_0^{2\pi}\exp\left(-\frac{(x\,\sin\,\theta)^2 - 2x(X_i\sin\,\theta) + X_i^2}{2h_1^2} - \frac{(x\,\cos\,\theta)^2 - 2x(Y_i\cos\,\theta) + Y_i^2}{2h_2^2}\right)\cdot x\,d\theta\,dx \qquad (23)\end{aligned}$$

Authorized licensed use limited to: UNIV OF ALABAMA-TUSCALOOSA. Downloaded on November 17, 2009 at 14:06 from IEEE Xplore. Restrictions apply.

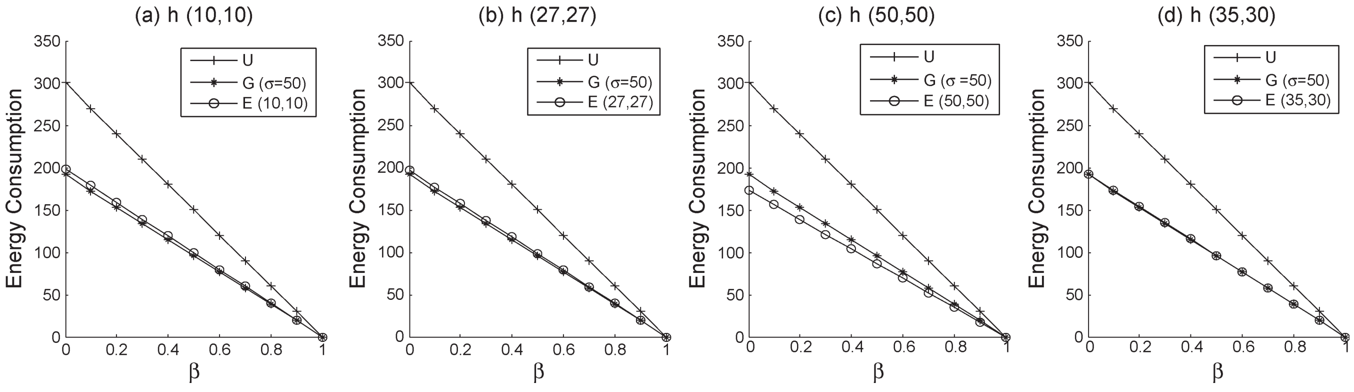


Fig. 9. Performance estimations. (a) Window width $h = (10, 10)$. (b) Window width $h = (27, 27)$. (c) Window width $h = (50, 50)$. (d) Window width $h = (35, 30)$.

by comparing the energy consumption derived from assumed (uniform), real-world (Gaussian), and estimation distributions, we show that our approach is more efficient for reflecting the real system performance in the real world.

First, in our experiment, the sample sensor-location knowledge is important to us. According to our knowledge, many research studies have been done in sensor-location technology, such as the geolocation approach [12]. In this geolocation approach, a few of the sensor nodes, called beacons, know their coordinates after deployment from satellite information (Global Positioning System). Thus, before deployment, we randomly elect some sensor nodes from the whole by deploying sensor nodes as beacons. Then, after deployment, we can regard these beacons as sample sensor nodes and obtain their location information. As aforementioned, the random sample location coordinate can be denoted by $(X_1, Y_1), (X_2, Y_2), \ldots (X_{n_0}, Y_{n_0})$, where n_0 is the size of the sample.

Second, the window width h plays an important role in KDE. Through Fig. 7, we show the impact of the window width h for distribution estimation. Fig. 7(a) shows the 2-D Gaussian distribution function in a sensor cluster. Fig. 7(c) shows the estimated distribution when the window width h is chosen as $h = (h_1, h_2) = (10, 10)$. As illustrated in Fig. 7(c), a large amount of random interferences exist, and thus, the curve face is rough. From Fig. 7(d), where the window width h is chosen as $h = (h_1, h_2) = (50, 50)$, we can see that the curve face is too flat because we ignore too much random local interference. As illustrated in Fig. 7(b), where the window width h is chosen as $h = (h_1, h_2) = (27, 27)$, the approximated effect is best among the aforementioned three estimated distributions. Thus, the selection of the window width is important for the effect of the KDE method. In the statistics field, many numerical methods have been developed for the decision of the window width. In this paper, we adopt the fast and accurate state-of-the-art bivariate kernel density estimator approach provided by Botev [13]. The value of the window width h can be generated by MATLAB based on the sample data. Fig. 8 shows the comparison between the Gaussian distribution and its estimation, whose window width h, i.e., $h = (h_1, h_2) = (35, 30)$, is generated by the fast and accurate state-of-the-art bivariate kernel density estimator approach. As illustrated in Fig. 8, the approximated effect is good.

After obtaining the window width h, we can study the system performance. In our experiment, the expected energy consumption can be studied based on (23). Fig. 9(a)–(d) shows the energy consumption versus fraction of sensor nodes allowed to sleep β for uniform, Gaussian, and estimated Gaussian distributions, where the window widths h are (27, 27), (10, 10), (50, 50), and (35, 30), respectively. In the experiment, we use the same parameters in Fig. 1(a)–(c) for the sensor network, and we collect $n_0 = 50$ location information of sensor nodes from a cluster. In Fig. 9(a)–(c), we can see that, when $h = (27, 27)$, the estimated performance is best. This result is consistent with the distribution estimation in Fig. 7(b)–(d). In Fig. 9(a)–(c), we can observe that, although the election of the window width is not very good, the estimated performance is still better than the performance from an inaccurate distribution assumption (uniform distribution). In our experiment, we adopt the fast and accurate state-of-the-art bivariate kernel density-estimator approach to generate the window width h, which is (35, 30). As illustrated in Fig. 9(d), the estimated performance is almost the same as the performance in the real world. Thus, we can make use of the KDE method to estimate the system performance in the real world and reduce the error caused by an incorrect assumption.

VI. Conclusion

In this paper, we have used energy consumption and lifetime issues as an example to study the impact of assumptions. Previous works are based on assumed pdf's that govern the distribution of sensor nodes in the sensing field. However, the actual distribution of sensor nodes may not easily be assumed. Our analytical study shows that, when a wrong assumption is used, the introduced error on the network energy consumption is very large and cannot be neglected.

In this paper, we have proposed a distribution-free method for estimating network energy consumption. In our proposed method, no assumption on the sensor node distribution is required. Instead, we take a small sample of the actual deployment sensor nodes and carry on a statistical analysis to capture the distribution function of deployment. We use the kernel density estimator to estimate the deployment distribution. Based on the obtained knowledge, the energy consumption in the network can be calculated.

Authorized licensed use limited to: UNIV OF ALABAMA-TUSCALOOSA. Downloaded on November 17, 2009 at 14:06 from IEEE Xplore. Restrictions apply.

The results show that a small sample of sensor nodes yields fairly good estimations on the distribution used. Compared with the case that the wrong assumption (the uniform distribution) is used and the case that the knowledge of the deployment distribution (Gaussian distribution) is completely known, our estimates give far better results.

Furthermore, we provide a mathematical approximation and a standard deviation study for energy consumption, as well as a more in-depth study for network lifetime.

References

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd HICSS*, 2000, p. 8020.

[2] M. Younis, M. Youssef, and K. Arisha, "Energy-aware routing in cluster-based sensor networks," in *Proc. MASCOT*, 2002, pp. 129–136.

[3] A. Sankar and Z. Liu, "Maximum lifetime routing in wireless ad hoc networks," in *Proc. IEEE INFOCOM*, 2004, pp. 1089–1097.

[4] J. H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 4, pp. 609–619, Aug. 2004.

[5] R. Madan, Z. Q. Luo, and S. Lall, "A distributed algorithm with linear convergence for maximum lifetime routing in wireless sensor networks," in *Proc. Annu. Allerton Conf. Commun., Control Comput.*, 2005.

[6] Y. Xu, J. Heideman, and D. Estrin, "Adaptive energy-conserving routing for multihop ad hoc networks," USC/ISI, Arlington, VA, Res. Rep. 527, Oct. 2000.

[7] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney, "Scheduling sleeping nodes in high density cluster-based sensor networks," *Mob. Netw. Appl.*, vol. 10, no. 6, pp. 825–835, Dec. 2005.

[8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer Series in Statistics. Berlin, Germany: Springer-Verlag, 2001.

[9] K. Wu, Y. Gao, F. Li, and Y. Xiao, "Lightweight deployment-aware scheduling for wireless sensor networks," *Mob. Netw. Appl. (MONET)*, vol. 10, no. 6, pp. 837–852, Dec. 2005.

[10] L. Wang and Y. Xiao, "A survey of energy-efficient scheduling mechanisms in sensor networks," *Mob. Netw. Appl. (MONET)*, vol. 11, no. 5, pp. 723–740, Oct. 2006.

[11] C. Liu, K. Wu, Y. Xiao, and B. Sun, "Random coverage with guaranteed connectivity: Joint scheduling for wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 6, pp. 562–575, Jun. 2006.

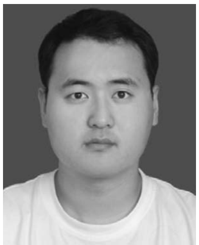
[12] A. Savvides, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, *Location discovery in ad-hoc wireless sensor networks*, Depts. Elect. Eng. and Comput. Sci., Univ. Calif. Los Angeles.

[13] [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/authors/27236

[14] K. Keith, *Mathematical Statistics*. London, U.K.: Chapman & Hall, 2000.

[15] S. M. Ross, *Stochastic Processes*. New York: Wiley, 1995.

[16] Y. Xiao, H. Li, Y. Pan, K. Wu, and J. Li, "On optimizing energy consumption for mobile handsets," *IEEE Trans. Veh. Technol.*, vol. 53, no. 6, pp. 1927–1941, Nov. 2004.



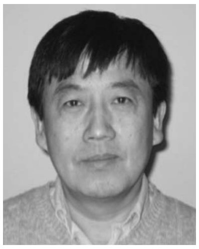
Miao Peng received the B.S. degree in applied mathematics from Dalian University of Technology, Dalian, China, in 2004 and the M.S. degree in mathematical statistics from Jilin University, Changchun, China, in 2007. He is currently working toward the Ph.D. degree in computer science with The University of Alabama, Tuscaloosa.

He is currently a Research Assistant with The University of Alabama. His research interests include wireless sensor networks, wireless network security, and energy-efficient wireless networks. In particular, he is interested in mathematical modeling in wireless and sensor networks.

Yang Xiao (SM'04) received the B.S. and M.S. degrees from Jilin University, Changchun, China, and the M.S. and Ph.D. degrees in computer science and engineering from Wright State University, Dayton, OH.

He is currently with the Department of Computer Science, The University of Alabama, Tuscaloosa. He is a Guest Professor with Jilin University and an Adjunct Professor with Zhejiang University, Hangzhou, China. He currently serves as the Editor-in-Chief of the *International Journal of Security and Networks*, the *International Journal of Sensor Networks*, and the *International Journal of Telemedicine and Applications*. His research areas are security, telemedicine, robot, sensor networks, and wireless networks. He has published more than 300 papers in major journals, refereed conference proceedings, and book chapters related to these research areas. His research has been supported by the U.S. National Science Foundation (NSF), the U.S. Army Research Office, etc.

Dr. Xiao was a voting member of the IEEE 802.11 Working Group from 2001 to 2004. He serves as an Associate Editor of several journals, e.g., the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He serves as a panelist for the NSF, the Canada Foundation for Innovation's Telecommunications Expert Committee, and the American Institute of Biological Sciences, as well as a referee/reviewer for many national and international funding agencies. He serves on the Technical Program Committee for more than 100 conferences such as the IEEE Conference on Computer Communications, the International Conference on Distributed Computing Systems, the ACM International Symposium on Mobile Ad Hoc Networking and Computing, the International Conference on Communications, the IEEE Global Telecommunications Conference, the Wireless Communications and Networks Conference, etc.



Pu Patrick Wang received the Ph.D. degree from Lehigh University, Bethlehem, PA, in 1990.

He is currently a Professor with the Department of Mathematics, The University of Alabama, Tuscaloosa. His research interests are theory and applications of stochastic processes. Some of the applications involved in his research include queuing theory and networks, telecommunication networks, and mathematical finance.

Authorized licensed use limited to: UNIV OF ALABAMA-TUSCALOOSA. Downloaded on November 17, 2009 at 14:06 from IEEE Xplore. Restrictions apply.