Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Message Batching in Wireless Sensor Networks — A Perturbation Analysis Approach

Xu Ning and **Christos G. Cassandras**
Dept. of Manufacturing Engineering and Center for Information and Systems Engineering
Boston University, Brookline, MA 02446
nx@bu.edu, cgc@bu.edu

***Abstract*— We address the problem of batching messages generated at nodes of a sensor network for the purpose of reducing communication energy at the expense of added latency. We first develop a baseline analytical model, derive conditions under which batching is profitable, and explicitly determine a batching time that optimizes a performance metric capturing the trade-off between communication energy and message latency. We then provide an on-line performance optimization method based on Smoothed Perturbation Analysis (SPA) for estimating the performance sensitivity with respect to the controllable batching time. We prove that the SPA gradient estimator is unbiased and combine it with a Stochastic Approximation (SA) algorithm for on-line optimization. Numerical results are provided for Poisson and Markov modulated Poisson arrival processes and illustrate the effectiveness of the message batching scheme.**

***Index Terms*— Wireless Sensor Network, Perturbation Analysis, Batching**

I. INTRODUCTION

A Wireless Sensor Network (WSN) consists of low-cost nodes which are mainly battery powered and have sensing and wireless communication capabilities [1]. Power consumption is a key issue in WSNs, since it directly impacts their lifetime in the likely absence of human intervention for most applications of interest. Energy in WSN nodes is consumed by the CPU, by sensors, and by radio, with the latter consuming the most [2].

Several approaches for reducing communication energy cost have been proposed and implemented [3]. However, among existing approaches few consider exploiting network traffic statistics for further savings, even though a large class of applications involves irregular, random, event-driven traffic. Since an important characteristic of WSNs is the sporadic and bursty nature of traffic, a natural question is: Upon detecting an event, should the sender transmit the message immediately, or is it profitable to intentionally delay it for some time period knowing that more events might occur within a short interval? If so, how long should the sender wait and what is the trade-off between energy and network performance, say average delay? In this paper, we propose a time-based message batching approach which utilizes network statistics to reduce communication energy cost. The contribution is to: (i) provide an analytical stochastic model on time-based message batching, and solve it under exponential distribution assumptions; (ii) present gradient estimators of performance, in which no a priori statistical arrival information is needed, and proved their unbiasedness; (iii) present an on-line control method using a Stochastic Approximation algorithm.

The time-based message batching problem is formulated in Section II, where the stochastic model is introduced. In Section III, with exponential assumptions, the model is solved analytically and some quantitative results are derived. In Section IV, we turn our focus to on-line control by using Smoothed Perturbation Analysis (SPA) to derive gradient estimators [4]. Using these estimators, in Section V we provide simulation results of on-line control using a Stochastic Approximation (SA) algorithm [5]. Finally, conclusions are given in Section VI.

II. TIME-BASED MESSAGE BATCHING

Our approach focuses on the link layer of a WSN. A sender node in a WSN detects a random event or receives a message from upstream, and sends a message to a receiver, which either relays it to the next hop or processes it. Random events are modeled through a point process. We adopt the same low-power listening and *variable length preamble* technique presented in [6]. As illustrated in Fig. 1, the preamble is initiated when the sender is ready to transmit and it allows a variable sleep time on the receiver side. In other words, unlike many other approaches, the preamble need not be longer than the sleep time to ensure proper reception. Although (as shown in Fig. 1) the preamble consists of discrete packets, we assume a continuous preamble in order to simplify the analysis. From the sender's perspective, the receiver channel polling events take place randomly and are modeled through a point process as well. This captures the fact that there may be multiple receivers for redundancy purposes, that clock drift and time offset behavior are possible, and that different sampling schedules may be adopted by different receivers.

In our previous work [6], upon detecting an event, the sender starts sending the preamble at once. However, in *time-based* message batching, since the sender anticipates more events/messages to come, it intentionally postpones sending the preamble for W units of time, where W is a preset parameter independent of the buffer content; this is in contrast to *queue-length based* batching where the preamble

The authors' work is supported in part by the National Science Foundation under Grant DMI-0330171, by AFOSR under grants FA9550-04-1-0133 and FA9550-04-1-0208, and by DOE under grant DE-FG52-06NA27490.

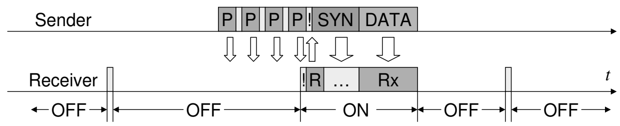

Fig. 1. Illustration of the variable preamble technique. After each preamble packet (P) sent, the sender listens for possible receiver reply (R) which is sent upon the receiver detecting P at polling events. When sender receives R, a synchronization packet (SYN) is sent, followed by the data payload (DATA). Thus the data are received.

is sent after a certain threshold in the queue is reached. We choose time-based batching because it ensures a bounded delay, whereas queue-length based batching may result in large delays especially when arrivals are very sporadic in WSNs. When the preamble meets with a polling event, as illustrated in Fig. 1, the entire batch of messages is transmitted. Note that the transmission of the message (DATA) and control packets (SYN) also consumes energy. However, we do not consider this cost since it is uncontrollable, while in our analysis we aim to determine a suitable batching time W so as to reduce the preamble cost (number of P packets). The fundamental trade-off in this problem is the following: If the waiting time W increases, the sender's energy consumption is reduced as more messages share a single preamble; on the other hand, all messages are further delayed by the increase in W. A special case arises when the polling process is deterministic. However, the analysis is much simpler as the sender can perfectly coordinate with receiver polling periods by varying W for each polling event. Therefore, in the following we focus on the more general stochastic case.

A typical sample path is shown in Fig. 2(a). The upper timeline is for the sender side and the lower one is for the receiver. The jth event or upstream message arrival time is denoted by A_j. The jth channel sampling event time is denoted by S_j.

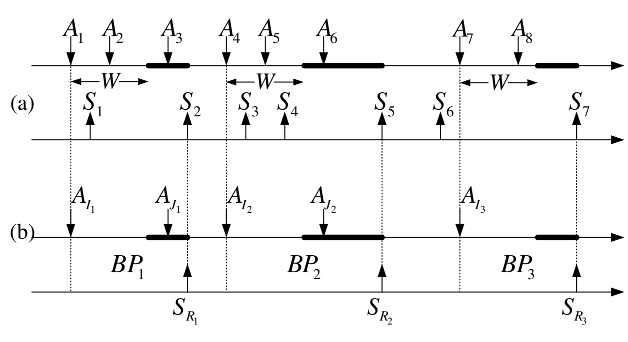

Fig. 2. (a) Typical sample path. (b) Critical events divide the sample path into busy periods.

After an arrival finds the sender's buffer empty, e.g., A_1 in Fig. 2(a), the sender waits for W units of time before sending a preamble which is indicated by a bold line segment. As a sampling event S_2 sees the preamble, messages A_1, A_2, A_3 are transmitted. Since the only randomness in the system lies in the arrival and sampling time epochs, the sample path is uniquely determined by two exogenous point processes: (i) The message arrival process $\{A_j : j \geq 1\}$, and (ii) The sampling process $\{S_j : j \geq 1\}$.

The sample path contains certain *critical events*. An arrival that finds an empty buffer is critical because it initiates a busy period at the sender node and also determines the starting time of a preamble. In addition, a particular sampling event that "downloads" the messages is critical as it ends a busy period. Clearly, the sample path consists of a sequence of busy periods $\{BP_i, i \geq 1\}$. A busy period BP_i starts with an arrival A_{I_i} which finds an empty buffer, and ends with a sampling event S_{R_i} which triggers the transmission of all the accumulated messages. These are critical events, as shown in Fig. 2(b). Once $\{A_j : j \geq 1\}$ and $\{S_j : j \geq 1\}$ are given, the index sets $\{I_i, i \geq 1\}$ and $\{R_i, i \geq 1\}$ can be recursively determined as follows:

$$I_1 = 1 \tag{1}$$

$$R_i = \min_j \{j : S_j > A_{I_i} + W\}, \ i \geq 1 \tag{2}$$

$$I_{i+1} = \min_j \{j : A_j > S_{R_i}\}, \quad i > 1 \tag{3}$$

An additional critical event is the first arrival during a preamble time within BP_i, denoted by A_{J_i}, where:

$$J_i = \min \{j : A_{I_i} + W < A_j < S_{R_i}\} \tag{4}$$

Note that J_i may not exist for some BP_i, e.g., BP_3 in Fig. 2(b).

The system performance depends on the message delays and the preamble length. For a message arriving at A_j, its delay is given by:

$$D_j = \min_i \{S_{R_i} : S_{R_i} > A_j\} - A_j \tag{5}$$

In each BP_i there is only one preamble, beginning after a delay W initiated by A_{I_i} and ending with the critical sampling event S_{R_i}. The length of this preamble is:

$$P_i = S_{R_i} - A_{I_i} - W \tag{6}$$

Denote by N the number of arrivals in a sample path and by B the number of busy periods (depending on N, obviously) and define:

$$\bar{D} = \lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} D_j, \quad \bar{P} = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{B} P_i \tag{7}$$

which are the long term average delay and preamble length *per message*, respectively. Assuming ergodicity, they are deterministic values determined by W and the statistics of $\{A_j : j \geq 1\}$ and $\{S_j : j \geq 1\}$. $\bar{D}$ and $\bar{P}$ reflect the key trade-off, since the goal of the batching mechanism is to delay sending a preamble so that a single preamble is shared by a batch of messages and energy consumption is reduced at the expense of message delay.

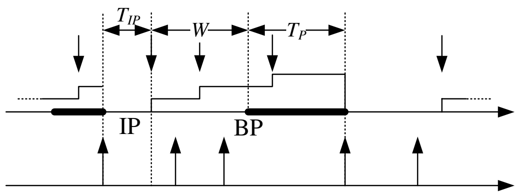

Fig. 3. A single busy period.

III. Analytical Solution with Poisson Processes

In this section, we assume that the arrival and sampling processes are Poisson with rate λ and μ, respectively. Due to the Markovian structure of the system, the analysis can be performed in terms of a single busy period BP, which begins when an arrival finds the system empty, and ends when a sampling event takes away the stored messages, as illustrated in Fig. 3. Denoting the average delay and preamble length per message in a BP by D and P respectively and assuming ergodicity, we have $\bar{D} = E[D]$ and $\bar{P} = E[P]$. Conversely, an idle period IP lies between two consecutive BPs, where there is no message in the sender's queue. Let T_{IP} denote the length of an IP and note that it is exponentially distributed with rate λ and $E[T_{IP}] = 1/\lambda$.

A BP consists of two phases. Phase 1 lasts for W time units, when the source is waiting with its radio off. Phase 2 occurs when the source continuously sends the preamble. Let T_P denote the length of phase 2. Note that it is the time until the next sampling event, therefore, it is also exponentially distributed with rate μ, and $E[T_P] = 1/\mu$. Thus, the expected total number of arrivals during a BP, including the first arrival, is $E[N_{BP}] = 1 + \lambda W + \lambda/\mu$, hence the average preamble time per message is: $E[P] = E[T_P/N_{BP}]$. Due to the regenerative structure of the system, we have

$$E[P] = \frac{E[T_P]}{E[N_{BP}]} = \frac{1}{\lambda + \lambda\mu W + \mu} \tag{8}$$

Next, to obtain $E[D]$, there are three cases corresponding to arrivals occurring at different times within a busy period: (i) the first arrival which initiates the BP; (ii) arrivals during the waiting period. (iii) arrivals during the preamble period. Denote by D_{ij} the jth arrival of the ith case, and by N_i the number of arrivals during case i. Hence, $E[D_{11}] = W + 1/\mu$, $E[N_2] = \lambda W$, $E[N_3] = \lambda/\mu$. Some straightforward calculations lead to:

$$E\left[\sum_{j=1}^{N_2} D_{2j}\right] = \lambda W\left(\frac{W}{2} + \frac{1}{\mu}\right),\ E\left[\sum_{j=1}^{N_3} D_{3j}\right] = \frac{\lambda}{\mu^2}$$

Due to the regenerative structure, we have

$$E[D] = \frac{E\left[D_{11} + \sum_{j=1}^{N_2} D_{2j} + \sum_{j=1}^{N_3} D_{3j}\right]}{E[1 + N_2 + N_3]} = \frac{W + \frac{1}{\mu} + \lambda W\left(\frac{W}{2} + \frac{1}{\mu}\right) + \frac{\lambda}{\mu^2}}{1 + \lambda W + \frac{\lambda}{\mu}} \tag{9}$$

Suppose our performance objective is chosen as a linear combination of the two metrics, $\bar{D}$ and $\bar{P}$:

$$J = \bar{D} + \alpha\bar{P} = E[D] + \alpha E[P] \tag{10}$$

where $\alpha > 0$. Taking derivatives with respect to W in (9) and (8) we get

$$\frac{dJ}{dW} = \frac{\frac{\lambda^2 W^2}{2} + \lambda\left(1 + \frac{\lambda}{\mu}\right)W + \left[1 + (1-\alpha)\frac{\lambda}{\mu}\right]}{\left(1 + \lambda W + \frac{\lambda}{\mu}\right)^2}$$

To determine the optimal waiting time W^*, we solve the equation $dJ/dW = 0$. This is equivalent to solving the quadratic equation in the numerator. Hence, J has a stationary point for $W \geq 0$ if and only if:

$$1 + (1-\alpha)\frac{\lambda}{\mu} \leq 0 \tag{11}$$

which implies that batching is profitable. This positive root corresponds to a local minimum of J since d^2J/dW^2 is positive. So the optimal batching time and corresponding cost are:

$$W^* = \frac{-\left(1 + \frac{\lambda}{\mu}\right) + \sqrt{\left(1 + \frac{\lambda}{\mu}\right)^2 - 2\left[1 + (1-\alpha)\frac{\lambda}{\mu}\right]}}{\lambda}$$

$$J^* = \frac{W^* + \frac{1}{\mu} + \lambda W^*\left(\frac{W^*}{2} + \frac{1}{\mu}\right) + \frac{\lambda}{\mu^2} + \alpha\frac{1}{\mu}}{1 + \lambda W^* + \frac{\lambda}{\mu}}$$

In the case where $W = 0$, i.e., no batching is carried out, the performance is

$$J_0 = \frac{1}{\mu} + \frac{\alpha/\mu}{1 + \lambda/\mu}$$

To further explore the benefit of batching when (11) holds, let $\Delta = J_0 - J^*$. We normalize the parameters by setting $1/\mu = 1, 1/\lambda = k$ and let

$$\tilde{\Delta}(k, \alpha) = \frac{J_0 - J^*}{J_0}$$

so that $\tilde{\Delta}(k, \alpha)$ is a function of k and α which characterizes the relative optimal batching benefits.

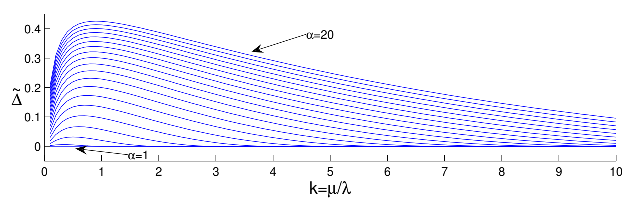

Fig. 4. Relative optimal message batching benefits under different k and α settings.

Figure 4 shows the relative benefit under different k and α settings. The curves are obtained by choosing different k and

α, then calculating J_0, J^* and $\tilde{\Delta}$. An interesting observation is that all curves attain their maximum at around $k = 1$, which implies that under the setting $\mu/\lambda = 1$ the batching scheme performs the best. This observation can be used as a guideline for tuning the receivers, although the problem itself focuses on the sender. Meanwhile, as α increases, the benefit is obviously larger since more emphasis is put on the power side of the objective.

IV. On-line Gradient Estimation and Optimization

In practice, network statistics are not known in advance, which calls for an adaptive control method. The analytical model breaks down when the Markovian assumptions of the previous section are relaxed. Therefore, we propose an on-line gradient estimation method based on Perturbation Analysis (PA) [4]. Thus, we attempt to extract from an observation of the sample path not only performance data, but also sensitivity information with respect to the controllable parameter W. This gradient information will be used in optimization, i.e., seeking the optimal value W^*, using a Stochastic Approximation (SA) algorithm.

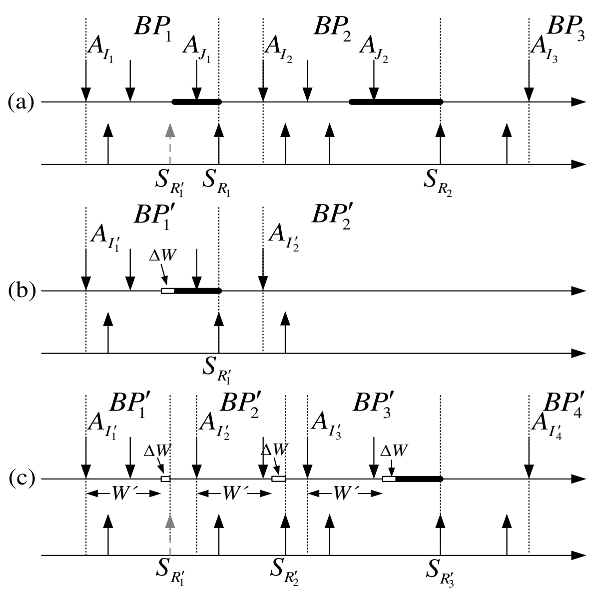

Fig. 5. (a) is the nominal sample path of two cases: without/with sampling event $S_{R_1'}$. (b), (c) are the perturbed sample paths without/with $S_{R_1'}$, respectively. $S_{R_1'}$ combined with A_{J_1} results in a drastically different perturbed sample path and discontinuous sample functions.

Consider the performance objective J in (10) and let $L_D(W,\omega)$, $L_P(W,\omega)$ be *sample functions* defined on an observed sample path denoted by ω and consisting of N arrivals and B busy periods:

$$L_D(W,\omega) = \frac{1}{N}\sum_{j=1}^{N} D_j(W,\omega) \quad (12)$$

$$L_P(W,\omega) = \frac{1}{N}\sum_{i=1}^{B} P_i(W,\omega) \quad (13)$$

and let

$$L(W,\omega) = L_D(W,\omega) + \alpha L_P(W,\omega) \quad (14)$$

By the ergodicity assumption, $J(W) = E[L(W,\omega)]$, hence

$$\frac{dJ(W)}{dW} = \frac{dE[L(W,\omega)]}{dW} = \lim_{\Delta W\to 0}\frac{E[\Delta L(W,\omega)]}{\Delta W} \quad (15)$$

In the most basic form of PA, known as Infinitesimal Perturbation Analysis (IPA), an unbiased estimator of dJ/dW is obtained through the sample derivative of $L(W,\omega)$ provided it exists and we are able to change the order of expectation and limit above. Thus, the IPA estimator is

$$\left[\frac{dJ(W)}{dW}\right]_{IPA} = \frac{d[L(W,\omega)]}{dW} \quad (16)$$

However, in this problem, there exists some ω such that $L(W,\omega)$ is discontinuous with respect to W (see Fig. 5). Due to this discontinuity, IPA will not produce an unbiased estimator, a key condition needed in the convergence of the SA algorithm. The observation in Fig. 5 that the existence of some events leads to large sample path changes motivates the use of Smoothed Perturbation Analysis (SPA) [7]. The main idea in SPA is to replace the original sample function $L(W,\omega)$ by a conditional expectation $E[L(W,\omega)|z]$, i.e., set

$$J(W) = E[L(W,\omega)] = E[E[L(W,\omega)|z]] \quad (17)$$

where z is called a "*characterization*" of the sample path which must be appropriately selected. Since $E[L(W,\omega)|z]$ is generally "smoother" than $L(W,\omega)$, we expect to be able to interchange expectation and limit and obtain the SPA estimator:

$$\left[\frac{dJ(W)}{dW}\right]_{SPA} = \lim_{\Delta W\to 0}\frac{E[\Delta L(W,\Delta W,\omega)|z]}{\Delta W} \quad (18)$$

where $\Delta L(W,\Delta W,\omega)$ is defined as the difference between the sample function values on the nominal and perturbed sample paths. Let us select the characterization z to be all the arrival events $\{A_j\}$ and sampling events $\{S_j\}$:

$$z = \{A_j : j \geq 1\} \cup \{S_j : j \geq 1\}$$

Let e_i denote the event (in the measure theoretic sense) that a sampling event such as $S_{R_1'}$ in Figure 5(c) above takes place, i.e., $e_i \equiv [A_{I_1} + W - \Delta W < S_k \leq A_{I_1} + W]$ for some S_k. Let $\bar{e}_i$ denote its complement. Assume that $\Pr(e_i)$ is an analytic function of the time $t_i = A_{I_i} + W$ and of ΔW around 0, i.e.,

$$\begin{aligned}\Pr(e_i) &= F(t_i,\Delta W) = F(t_i,0) + a_1(t_i)\Delta W + o(\Delta W)\\ &= a_1(t_i)\Delta W + o(\Delta W)\end{aligned} \quad (19)$$

which is of order $O(\Delta W)$. We further assume that inter-sampling times are mutually independent, so that the probability that, from (19), two or more such events occur is of higher order than $O(\Delta W)$: $\Pr(e_i e_j \ldots) = o(\Delta W)$. Define ΔL_i to be the perturbation in the sample function generated within BP_i. Given z, the total expected perturbation over all BP's is:

$$E\left[\Delta L\left(W,\Delta W,\omega\right)|z\right]= \sum_{i=1}^{B}\left[E\left[\Delta L_i|\bar{e}_i,z\right]\Pr\left(\bar{e}_i\right)+E\left[\Delta L_i|e_i,z\right]\Pr\left(e_i\right)\right] + \sum_{i=1}^{B}\sum_{j=1,j\neq i}^{B}E\left[\Delta L_{i,j}|e_ie_j,z\right]\Pr\left(e_ie_j\right)+... = \sum_{i=1}^{B}\left[E\left[\Delta L_i|\bar{e}_i,z\right]\Pr\left(\bar{e}_i\right)+E\left[\Delta L_i|e_i,z\right]\Pr\left(e_i\right)\right] + o\left(\Delta W\right) \quad (20)$$

Thus, the evaluation of the SPA sample function $E\left[\Delta L\left(W,\Delta W,\omega\right)|z\right]$ boils down to two cases corresponding to $\bar{e}_i$ and e_i, where $\Pr\left(e_i\right) = a_1\left(t_i\right)\Delta W+o\left(\Delta W\right)$, $\Pr\left(\bar{e}_i\right)=1-\Pr\left(e_i\right)$.

First, under $\bar{e}_i$ the only difference in the sample function is an increase in the preamble length in BP_i since the waiting time is reduced by $\Delta W>0$ and this effect does not propagate to any further busy periods. All arrivals must still wait until S_{R_i} to be transmitted, therefore, $E\left[\Delta D_j|\bar{e}_i,z\right]= 0$, $j=1,\dots,N$, $E\left[\Delta P_i|\bar{e}_i,z\right]=\Delta W$, and the total perturbation in this case is: $E\left[\Delta L_{D,i}|\bar{e}_i,z\right]=0$, $E\left[\Delta L_{P,i}|\bar{e}_i,z\right]= \Delta W/N$. Combining (18) and the first part of (20), we get

$$\lim_{\Delta W\to 0}\frac{\Pr\left(\bar{e}_i\right)}{\Delta W}E\left[\Delta L_{D,i}|\bar{e}_i,z\right]=0 \quad (21)$$

$$\lim_{\Delta W\to 0}\frac{\Pr\left(\bar{e}_i\right)}{\Delta W}E\left[\Delta L_{P,i}|\bar{e}_i,z\right] = \lim_{\Delta W\to 0}\frac{1}{N}\frac{1-a_1\left(t_i\right)\Delta W+o\left(\Delta W\right)}{\Delta W}\Delta W=\frac{1}{N} \quad (22)$$

On the other hand, under e_i there will be major changes in the delays of messages, as well as preamble lengths. Combining (18) and the second part of (20) we get

$$\lim_{\Delta W\to 0}\frac{\Pr\left(e_i\right)}{\Delta W}\left[\sum_{i=1}^{B}E\left[\Delta L_i|e_i,z\right]\right] \quad (23)$$

$$= a_1\left(t_i\right)\lim_{\Delta W\to 0}\sum_{i=1}^{B}E\left[\Delta L_i|e_i,z\right] \quad (24)$$

so that the remaining task is to evaluate $E\left[\Delta L_i|e_i,z\right]$ and then take the limit $\Delta W\to 0$. To accomplish this, we need to partially *reconstruct* the perturbed sample path. First, referring to Fig. 5, if A_{J_i} does not exist, we can see that the message batch will be transmitted at $S_{R'_i}$ and no propagation to future busy periods exists. However, if A_{J_i} exists, the perturbation caused by ΔW will propagate because the critical events in BP_i are affected. Let $\{I'_k,k\geq 1\}$ and $\{R'_k,k\geq 1\}$ be the index sets of critical events in the perturbed sample path and observe that

$$I'_1 = I_i \quad (25)$$

$$S_{R'_1} = A_{I_i}+W-\xi$$

where $\xi\in[0,\Delta W]$. As $\Delta W\to 0$ in (24), $\xi\to 0$ so we can omit it and obtain the following recursive expressions:

$$S_{R'_1} = A_{I_i}+W \quad (26)$$

$$R'_k = \min_j\left\{j:S_j>A_{I'_k}+W\right\},\ k\geq 2 \quad (27)$$

$$I'_{k+1} = \min_j\left\{A_j:A_j>S_{R'_k}\right\},\quad k\geq 1 \quad (28)$$

This procedure is carried out through the $[l(i)-1]$th busy period with $l(i)$ such that $A_{I'_{l(i)}}$ coincides with some A_{I_j} in the nominal sample path, i.e.,

$$I'_{l(i)} = I_{j_0(i)}\in\{I_j:j>i\} \quad (29)$$

as illustrated in Fig. 5 with $A_{I_3}=A_{I'_4}$. Denote by D'_j the new delay of the jth arrival, and by P'_k the new preamble in the kth busy period beginning with BP_i. It is easy to see that

$$D'_j = \min_i\left\{S_{R'_i}:S_{R'_i}>A_j\right\}-A_j \quad (30)$$

$$P'_1 = 0 \quad (31)$$

$$P'_k = S_{R'_k}-A_{I'_k}-W,\quad k\geq 2 \quad (32)$$

Therefore,

$$\lim_{\Delta W\to 0}E\left[\Delta L_{D,i}|e_i,z\right]=\frac{1}{N}\sum_{I_i\leq j<I'_{l(i)}}\left(D'_j-D_j\right) \quad (33)$$

$$\lim_{\Delta W\to 0}E\left[\Delta L_{P,i}|e_i,z\right]=\frac{1}{N}\left(\sum_{j=2}^{l(i)-1}P'_j-\sum_{j=i}^{j_0(i)-1}P_j\right) \quad (34)$$

Recalling (14) and (18), we obtain the left-hand SPA gradient estimator for $\bar{D}$ using (21), (24), (33):

$$\left[\frac{d\bar{D}}{dW}\right]^{-}_{SPA}=-\frac{1}{N}\sum_{i=1}^{B}\sum_{I_i\leq j<I'_{l(i)}}a_1\left(t_i\right)\left(D'_j-D_j\right) \quad (35)$$

Similarly, using (22) and (34) we obtain

$$\left[\frac{d\bar{P}}{dW}\right]^{-}_{SPA}= -\frac{1}{N}\sum_{i=1}^{B}\left[a_1\left(t_i\right)\left(\sum_{j=2}^{l(i)-1}P'_j-\sum_{j=i}^{j_0(i)-1}P_j\right)+1\right] \quad (36)$$

Finally, combining the two and using (14),

$$\left[\frac{dJ\left(W\right)}{dW}\right]^{-}_{SPA}=\left[\frac{d\bar{D}}{dW}\right]^{-}_{SPA}+\alpha\left[\frac{d\bar{P}}{dW}\right]^{-}_{SPA} \quad (37)$$

Note that $a_1\left(t_i\right)$ depends on the sampling event distributions and must be separately evaluated. To summarize, the SPA gradient estimator algorithm is described as follows:

1) For each BP_i: (1) initialize I'_1 through (25); (2) use (26)-(29) to determine the critical events in the perturbed sample path; (3) use (30) to calculate the perturbed delay D'_j, $j=1,\dots,N$; (4) use (31)-(32) to calculate the perturbed preamble P'_i, $i=1,\dots,B$.

2) Obtain the SPA derivative estimate through (35), (36), and (37).

While (35)-(37) are left derivative estimators, right derivative estimators can be derived using similar analysis. Assuming mild conditions: (i) $E[A_N] < \infty$; (ii) $E[S_{j+1} - S_j] < \infty, j = 1, 2, ...$; (iii) $F(t_i, \Delta W)$ is analytic for all i around $\Delta W = 0$, we have the following theorem to establish the unbiasedness of (35)-(37):

Theorem 1: The SPA gradient estimator given by (35)-(37) is unbiased.

Proof: See [8]. ■

Using the gradient estimator (37), we use a Stochastic Approximation (SA) algorithm of the form:

$$W_{k+1} = \Pi_{[a,b]}\left[W_k - \frac{\beta}{k^\delta}\left[\frac{dJ(W_k)}{dW_k}\right]_{SPA}\right], \quad k \geq 0 \tag{38}$$

with W_0 being an initial point and $\Pi_{[a,b]}[x]$ a projection of x onto interval $[a, b]$. The parameters β and δ in (38) need to be carefully chosen to ensure convergence and regulate convergence speed. The guidelines are: (i) the algorithm converges for $\beta > 0$ and $0.5 < \delta \leq 1$; (ii) larger β and smaller δ will result in fast response but also higher variance, while smaller β and larger δ will have a slower response. In each step of the algorithm, we observe a sample path with N messages, obtain $[dJ(W_k)/dW_k]_{SPA}$ by the SPA algorithm, and use (38) to obtain W_{k+1}. By the fact that (37) is unbiased, W_k converges to W^* where $dJ(W^*)/dW^* = 0$. A similar proof of convergence can be found in [9].

V. Numerical Results

Exponential Arrival Processes. First, we consider the case where the arrival and sampling processes are Poisson with rate λ and μ, respectively so as to compare our results with the analysis of Section III. Therefore, $a_1(t_i)$ in (19) is simply a constant μ. Fig. 6(a) shows a sample trajectory of the optimization process using the SA algorithm. Here, we use the objective function (10) with $\alpha = 20$ and choose $\beta = 6$ and $\delta = 1$ as the parameter values in (38). We performed 200 iterations and we note that W is already at the vicinity of the optimal point after approximately 100 iterations.

Markov Modulated Poisson Arrival Process. One feature of the SPA gradient estimator is that it does not depend on the arrival process distribution, which allows the controller to adapt to different network traffic patterns. In this section, we use a Markov Modulated Poisson Process (MMPP) to model bursty data traffic in a WSN. A MMPP consists of an underlying continuous time Markov process $\{X_t; t \geq 0\}$ and a function $\lambda(X_t)$ which specifies the arrival rate in each state. One simple MMPP example is a Poisson process with exponentially distributed on-off durations, where the states of a continuous time Markov chain are $\{0, 1\}$ and arrivals occur only in state 1 with rate λ_{ON}. In this example we still assume the same Poisson process for receiver polling epochs so $a_1(t_i) = \mu$. Fig. 6(b) show the derivative estimates obtained and a sample trajectory of the optimization process (38). Unfortunately, under MMPP arrivals, there are no analytical results for comparison purposes. We can still see, however, that the SA algorithm readily converges to the optimal W value in Fig. 6(b).

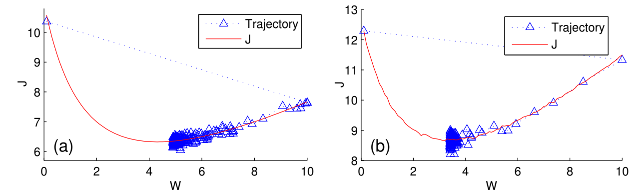

Fig. 6. (a) On-line optimization using the SA algorithm with exponential arrival process. Arrival and channel polling rates are $\lambda = \mu = 1$. (b) Optimization with arrival process being a Markov Modulated Poisson Process. Transition rates are $\lambda_{10} = 0.2$, $\lambda_{01} = 0.022$. Arrival rate and channel polling rate is also $\lambda_{ON} = \mu = 1$.

VI. Conclusions

We have proposed a time-based message batching approach for reducing transmission energy consumption in WSNs. When no analytical model is available, we have developed a gradient estimator using Smooth Perturbation Analysis (SPA) and proved its unbiasedness. Since the SPA gradient estimator does not depend on the arrival distribution, it can be used in conjunction with a Stochastic Approximation (SA) algorithm allowing the controllable parameter W to adapt to possibly changing network traffic conditions. Future work is directed at extending this approach to multi-hop sensor networks and relaxing the requirement that the probability $P(t_i, \Delta W)$ of an e_i event taking place is known.

References

[1] S. Megerian and M. Potkonjak, *Wireless Sensor Networks*, ser. Wiley Encyclopedia of Telecommunications. New York, NY: Wiley-Interscience, January 2003.

[2] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2004, pp. 188–200.

[3] K. Kredo II and P. Mohapatra, "Medium access control in wireless sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 961–994, March 14 2007.

[4] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Norwell, MA, USA: Kluver Academic Publishers, 1999.

[5] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications, Second Edition*. New York, NY: Springer-Verlag New York, Inc, 2003.

[6] X. Ning and C. G. Cassandras, "Dynamic sleep time control in event-driven wireless sensor networks," in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, December, 2006, pp. 2722–2727.

[7] W. Gong and Y.-C. Ho, "Smoothed (conditional) perturbation analysis of discrete event dynamical systems," *IEEE Transactions on Automatic Control*, vol. 32, no. 10, pp. 858–866, October 1987.

[8] X. Ning and C. G. Cassandras, "Message batching in wireless sensor networks — a perturbation analysis approach," Boston University, Tech. Rep., 2007, http://people.bu.edu/nx/MessageBatchingTR.pdf.

[9] K. Gokbayrak and C. G. Cassandras, "Online surrogate problem methodology for stochastic discrete resource allocation problems," *Journal of Optimization Theory and Applications*, vol. 108, no. 2, pp. 349–376, February 2001.