Discrete Event Dyn Syst (2010) 20:409–439
DOI 10.1007/s10626-009-0080-9

Message Batching in Wireless Sensor Networks—A Perturbation Analysis Approach

Xu Ning · Christos G. Cassandras

Received: 16 May 2008 / Accepted: 29 July 2009 / Published online: 11 August 2009

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 consider a time-based batching approach. We first develop baseline analytical models based on Markovian assumptions, 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 message arrival processes and illustrate the effectiveness of the message batching scheme.

Keywords Wireless sensor network · Perturbation analysis · Batching · Queueing systems

The authors' work is supported in part by the National Science Foundation under grants DMI-0330171 and EFRI-0735974, by AFOSR under grants FA9550-07-1-0213 and FA9550-07-1-0361, and by DOE under grant DE-FG52-06NA27490. The research in this paper was conducted while Xu Ning was a student at Boston University.

X. Ning (✉)
Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA
e-mail: xuning@microsoft.com

C. G. Cassandras
Division of Systems Engineering, Boston University, 15 Saint Mary's St., Brookline, MA 02446, USA
e-mail: cgc@bu.edu

1 Introduction

A Wireless Sensor Network (WSN) consists of low-cost nodes which are mainly battery powered and have sensing and wireless communication capabilities (Megerian and Potkonjak 2003). Usually, the nodes in such a network share a common objective, such as environmental monitoring or event detection. Due to limited on-board power, nodes rely on short-range communication and form a multi-hop network to deliver information to a base station. Power consumption is a key issue in WSNs, since it directly impacts their lifespan 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 (Shnayder et al. 2004). In Ye et al. (2004), four major sources of energy waste in communication are identified: (1) a *collision* occurs when a packet is corrupted by other simultaneous transmissions and requires a retransmission; (2) *overhearing* arises when a node receives a packet which is not destined to it; (3) *control packet overhead* is the energy cost incurred during sending and receiving control packets instead of actual data payload; (4) *idle listening* occurs when the receiver staying awake during network idle time anticipating an incoming transmission. Among these four sources, idle listening accounts for the most significant waste of energy.

Energy waste due to idle listening can be reduced by adopting a Medium Access Control (MAC) scheme. MAC can be categorized into *scheduled* and *unscheduled* schemes. Scheduled schemes, such as TDMA (Sohrabi et al. 2000) and S-MAC (Ye et al. 2004), maintain a schedule among a small cluster of nodes such that nodes have coordinated transmission. Therefore, nodes can turn off their radio according to the schedule. Unscheduled schemes, on the other hand, try to emulate an "always-on" receiver by introducing additional ad-hoc synchronization. One way to achieve this is to use Low-Power Listening (LPL), which has been adopted previously, for instance, in radio paging systems (Mangione-Smith 1995). LPL uses a preamble-channel polling scheme to synchronize sender and receiver; detail will be given in Section 2. Unscheduled MACs using LPL include B-MAC (Polastre et al. 2004), WiseMAC (El-Hoiydi and Decotignie 2004) and X-MAC (Buettner et al. 2006), etc. One obvious advantage of unscheduled MAC is its universality, since all transmission controls are transparent to the applications to which it just appears to be an always-on radio. Another advantage is that it does not need in advance synchronization (it can, though, benefit from it since a preamble can be shortened when the transmission pair is roughly synchronized).

While LPL greatly reduces idle listening, large control packet overhead is incurred in the form of the preamble. To reduce control packet overhead, we consider the usage of *batching*. Batching approaches are widely used in all kinds of systems in order to reduce overhead or setup cost (Deb and Serfozo 1973; Deb 1984; Cassandras and Yu 2000). To the best of our knowledge, applying batching approaches in WSN communication has not been investigated. Moreover, 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 we explore in this paper 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 savings and performance, e.g., average delay?

In this paper, we study a *time-based* message batching approach in a discrete event system framework with the objective of reducing communication energy cost. Our contributions are twofold. First, we provide analytical stochastic models for time-based message batching, and solve it under Poisson process assumptions. This Markovian analysis gives us a baseline theoretical results. Second, we present an on-line control method to determine the optimal time for time-based message batching, in which no a priori statistical arrival information is needed. This involves performance sensitivity estimation based on sample path analysis using Perturbation Analysis (PA) techniques (Cassandras and Lafortune 1999) and a Stochastic Approximation algorithm (Kushner and George Yin 2003) to find the optimal batching time. We prove that the gradient estimators are unbiased and that this algorithm converges to the optimum.

This paper is organized as follows. The message batching problem and the discrete event model are introduced in Section 2. In Section 3, with exponential assumptions, the batching problem is solved analytically. In Section 4, we turn our focus to on-line control by using Smoothed Perturbation Analysis (SPA) to derive gradient estimators (Cassandras and Lafortune 1999; Fu and Hu 1997). Using these estimators, in Section 5 we provide simulation results of on-line control using a Stochastic Approximation (SA) algorithm. Finally, conclusions are given in Section 6.

2 Problem description and discrete event model

2.1 LPL and message batching problem

A sender node in a WSN detects a random event or receives a message from upstream (i.e., a node closer to a data source), 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 one form of LPL–Variable Preamble LPL at the link level. Some implementation details can be found in Joe and Ryu (2007), Mahlknecht and Bock (2004) and Buettner et al. (2006). Illustrated in Fig. 1, Variable Preamble LPL has the following main steps:

1. The receiver remains at a sleep state most of the time, and occasionally wakes up to sample the channel in order to determine whether it is busy or idle.
2. When the sender wants to send a message, it begins with an attached signal called the "preamble" to the receiver. The preamble can be viewed as a "wake up signal". After the preamble signal, the sender sends the message.
3. When the receiver wakes up and samples the channel, either of two cases may occur: (*i*) If the channel is idle, the receiver sets the next wake-up time and sleeps again, (*ii*) If the channel is busy (preamble detected), the receiver stays on until the message is received. After transmission, the receiver also sets its next wake-up time and sleeps again.

In this type of LPL, the preamble (or "wake up signal") is initiated when the sender is ready to transmit and it allows a variable sleep time on the receiver side. In

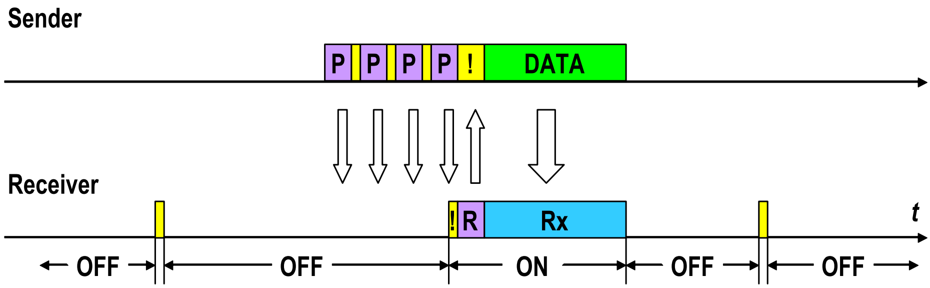


Fig. 1 Illustration of the *Variable Preamble* LPL (Ning and Cassandras 2006). The preamble is in the form of a sequence of packets interlaced with short listening periods. After each preamble packet (P) sent, the sender listens for possible receiver reply (R) which is sent upon the receiver detecting 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

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, receiver channel polling events take place randomly and are modeled through a point process as well. This captures the fact that (*a*) there may be multiple receivers for redundancy purposes, (*b*) random clock drift and time offset behaviors are possible, and (*c*) different sampling schedules may be adopted by different receivers.

Generally, upon detecting an event, the sender starts sending the preamble at once. Message batching is a mechanism to intentionally delay sending this preamble in anticipation of more events/messages to come, such that when the preamble meets with a polling event, as illustrated in Fig. 1, the *entire batch* of messages is transmitted. There are multiple ways to determine when to send the preamble, with the two most fundamental ways being (*i*) time-based message batching and (*ii*) queue-length-based batching. In time-based message batching, the sender 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 is sent immediately after the queue length reaches a threshold. 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 fixed, while in our analysis we aim to determine a suitable batching time or threshold so as to reduce the preamble cost.

In this paper we only consider time-based message batching because it is more suitable in the WSN environment. The implementation of time-based batching is a very simple time-out mechanism. It does not force the controller to wait, say K messages before sending out the preamble. When the traffic is bursty, e.g., an on-off type of traffic, if the queue length has not reached K before the exogenous arrival process "switches off", the sender has to wait for the whole "off" duration, incurring a large delay.

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 without any clock drift or offset or other randomness. However, the analysis is trivial because 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.

2.2 Discrete event model

To simplify the analysis, we make the following assumptions: (1) Generally, the interarrival time of the events is much larger than preamble durations, the time to transmit a message and the duration of a channel sampling activity. Therefore, we model channel samplings as points in time. (2) Although in the variable preamble LPL a sequence of discrete preamble packets is sent, we model the preamble as a continuous signal whose duration is a real positive number. In addition, we focus on the preamble and the channel sampling activities and ignore the energy cost incurred during the handshaking and the transmission of the data payload part of the message, since it is not controllable in the scope of our problem. A typical sample path of time-based message batching is shown in Fig. 2a. The upper timeline is for the sender side and the lower one is for the receiver. The jth event or upstream message arrival is denoted by A_j. The jth channel sampling event is denoted by S_j.

After an arrival finds the sender's buffer empty, e.g., A_1 in Fig. 2a, the sender waits for W units of time before sending a preamble which is indicated by a thicker 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 $\{S_j : j \geq 1\}$.

Certain arrival/sampling events are *critical events*. An arrival that finds an empty buffer is critical because it initiates a *Batching Period (BP)* at the sender node, i.e., a time interval where the sender's buffer is not empty, 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 BP. Clearly, the sample path consists of a sequence of BPs $\{BP_i, i \geq 1\}$. A batching 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. 2b. Once $\{A_j : j \geq 1\}$ and $\{S_j : j \geq 1\}$ are given, the index sets $C_I = \{I_i, i \geq 1\}$ and $C_R = \{R_i, i \geq 1\}$ can be recursively determined as follows:

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

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

$$I_{i+1} = \min_{j=1,2,\ldots} \{j : A_j > S_{R_i}\}, \quad i > 1 \tag{3}$$

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

$$D_j = \min_{i=1,2,\ldots} \{S_{R_i} : S_{R_i} > A_j\} - A_j \tag{4}$$

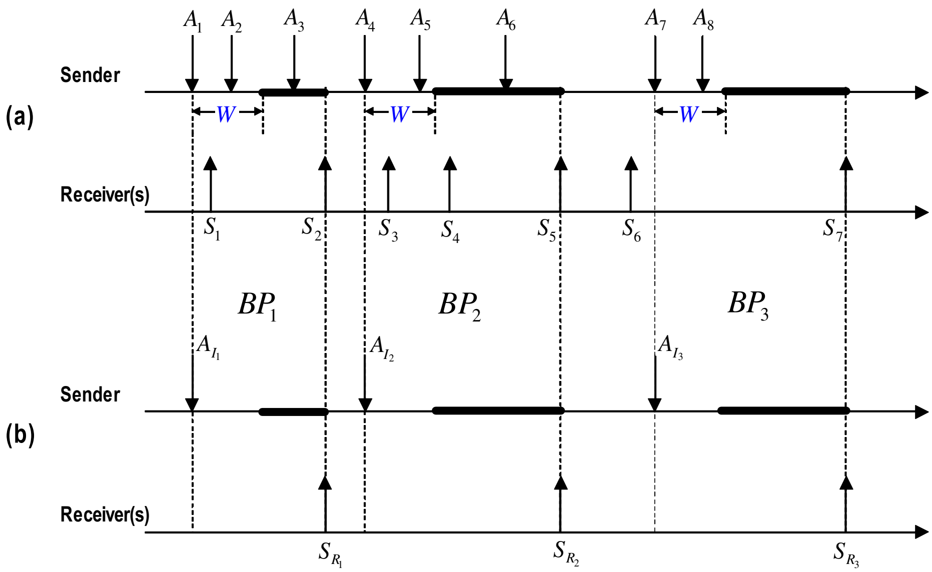


Fig. 2 **a** Typical sample path in time-based message batching. **b** Critical events divide the sample path into *batching periods*, i.e. time intervals where messages exist in the sender's queue

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:

$$P_i = S_{R_i} - A_{I_i} - W \tag{5}$$

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

$$\bar{D} = \lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} D_j, \quad \bar{Q} = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{B} P_i \tag{6}$$

which are the long term average delay and preamble length *per message*, respectively. Assuming ergodicity, $\bar{D}$ and $\bar{Q}$ are deterministic quantities dependent on W and on the statistics of $\{A_j : j \geq 1\}$ and $\{S_j : j \geq 1\}$. $\bar{D}$ and $\bar{Q}$ 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.

3 Analytical solution with Poisson processes

To find the analytical expression of mean delay and preamble, 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 batching 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. As a first step, we obtain expressions for $\bar{Q}$ and $\bar{D}$ in terms of W, λ and μ:

Lemma 1 *In time-based message batching, the long-run average delay and long-run average preamble per message are given by:*

$$\bar{Q} = \frac{1/\mu}{1 + \lambda W + \dfrac{\lambda}{\mu}}$$

$$\bar{D} = \frac{W + \dfrac{1}{\mu} + \lambda W \left(\dfrac{W}{2} + \dfrac{1}{\mu} \right) + \dfrac{\lambda}{\mu^2}}{1 + \lambda W + \dfrac{\lambda}{\mu}}$$

Proof 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 P denote the length of the preamble time. Note that it is the time until the next sampling event, therefore, it is also exponentially distributed with rate μ, and $E[P] = 1/\mu$. Thus, the expected total number of arrivals during a BP, including the first arrival, is

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

Because of the regenerative structure of the system, the average preamble time per message is $\bar{Q} = E[P]/E[N_{BP}]$. To see this, consider a sample path with B batching periods. We have:

$$\bar{Q}(B) = \frac{\sum_{i=1}^{B} P_i}{\sum_{i=1}^{B} N_{BP,i}} = \frac{\frac{1}{B}\sum_{i=1}^{B} P_i}{\frac{1}{B}\sum_{i=1}^{B} N_{BP,i}}$$

where $N_{BP,i}$ is the number of messages in BP_i. Hence, because for $i \neq j$, P_i and P_j are independent, and $N_{BP,i}$ and $N_{BP,j}$ are independent because of the regenerative structure, by the law of large numbers (Gaivoronski and Messina 2006),

$$\bar{Q} = \lim_{B\to\infty} \bar{Q}(B) = \lim_{B\to\infty} \frac{\frac{1}{B}\sum_{i=1}^{B} P_i}{\frac{1}{B}\sum_{i=1}^{B} N_{BP,i}} = \frac{E[P]}{E[N_{BP}]}$$

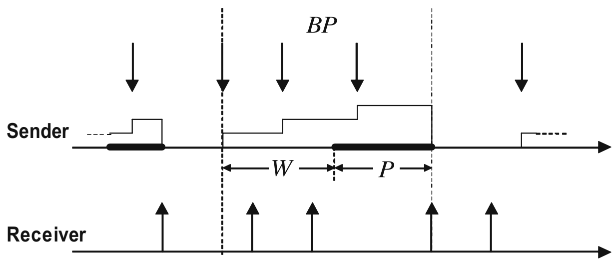


Fig. 3 A single batching period in time-based message batching

Therefore, we have

$$\bar{Q} = \frac{E[P]}{E[N_{BP}]} = \frac{1/\mu}{1 + \lambda W + \dfrac{\lambda}{\mu}} \tag{7}$$

Next, to obtain $\bar{D}$, there are three cases depending on when an arrival occurs within a BP:

Case 1 First arrival. This arrival initiates the BP and waits for $D_{11} = W + P$. Therefore,

$$E[D_{11}] = W + \frac{1}{\mu} \tag{8}$$

Case 2 Arrivals during the waiting period. By the Poisson assumption, if the number of arrivals is N_2, $E[N_2] = \lambda W$. Moreover, denoting by D_{2j} the delay experienced by the jth arrival in this period, the accumulated delay is $\sum_{j=1}^{N_2} D_{2j}$. By the property of the Poisson process that the unordered event times are uniformly distributed in the waiting period conditioned on the number of events in the waiting period (c. f. Chapter 2 in Ross 1995), the accumulated delay given N_2 is

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

It follows that

$$\begin{aligned} E\left[\sum_{j=1}^{N_2} D_{2j}\right] &= E\left[E\left[\sum_{j=1}^{N_2} D_{2j} \mid N_2\right]\right] = E\left[N_2\left(\frac{W}{2} + \frac{1}{\mu}\right)\right] \\ &= \lambda W\left(\frac{W}{2} + \frac{1}{\mu}\right) \end{aligned} \tag{9}$$

Case 3 Arrivals during the preamble period. First, if the number of arrivals during this period is N_3, then

$$E[N_3] = E[E[N_3|P]] = E[\lambda P] = \frac{\lambda}{\mu}$$

The accumulated delay is $\sum_{j=1}^{N_3} D_{3j}$. To calculate the expected value, we condition on T_P and N_3 and use a similar argument as in Case 2:

$$\begin{aligned} E\left[\sum_{j=1}^{N_3} D_{3j}\right] &= E\left[E\left[E\left(\sum_{j=1}^{N_3} D_{3j}|N_3, P\right)|P\right]\right] \\ &= E\left[E\left(\frac{N_3 P}{2}|P\right)\right] = E\left[\frac{\lambda P^2}{2}\right] \\ &= \frac{\lambda}{2}\int_0^{\infty} x^2 \mu e^{-\mu x} dx = \frac{\lambda}{\mu^2} \end{aligned} \tag{10}$$

Again, due to the regenerative structure and using Eqs. 8, 9, and 10, the long-run average delay is given by

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

□

Suppose our performance objective is chosen as a linear combination of the two metrics parametrized by W, $\bar{D}(W)$ and $\bar{Q}(W)$:

$$J(W) = \bar{D}(W) + \alpha\bar{Q}(W) \tag{12}$$

where $\alpha > 0$. Then, using Lemma 1, we can find the optimal parameter W^*, which is given in the following theorem:

Theorem 1 *In time-based message batching, let the performance be a linear combination of $\bar{D}(W)$ and $\bar{Q}(W)$:*

$$J(W) = \bar{D}(W) + \alpha\bar{Q}(W)$$

Then, it is only profitable to perform batching when

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

and the optimal W is given by

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

Proof Taking derivatives with respect to W in Eqs. 11 and 7 we get:

$$\frac{dJ}{dW} = \frac{\frac{\lambda^2 W^2}{2} + \lambda\left(1 + \frac{\lambda}{\mu}\right)W + \left[1 + (1-\alpha)\frac{\lambda}{\mu}\right]}{\left(1 + \lambda W + \frac{\lambda}{\mu}\right)^2} \tag{13}$$

The optimal waiting time W^* is obtained by solving $dJ/dW = 0$. This is equivalent to solving the quadratic equation in the numerator, which yields two roots:

$$\frac{-\left(1 + \frac{\lambda}{\mu}\right) \pm \sqrt{\left(1 + \frac{\lambda}{\mu}\right)^2 - 2\left[1 + (1-\alpha)\frac{\lambda}{\mu}\right]}}{\lambda} \tag{14}$$

The root we are interested in should be non-negative. Hence, J has a stationary point for $W \geq 0$ if and only if:

$$1 + (1-\alpha)\frac{\lambda}{\mu} \leq 0 \tag{15}$$

which implies that batching is profitable. This root corresponds to a local minimum of J since d^2J/dW^2 is positive. In fact, this local minimum is also *global* for $W > 0$ because the other root in Eq. 14 must be negative. So,

$$W^* = \frac{-\left(1+\frac{\lambda}{\mu}\right) + \sqrt{\left(1+\frac{\lambda}{\mu}\right)^2 - 2\left[1+(1-\alpha)\frac{\lambda}{\mu}\right]}}{\lambda} \tag{16}$$

$$J^* = \frac{W^* + \frac{1}{\mu} + \lambda W^*\left(\frac{W^*}{2}+\frac{1}{\mu}\right) + \frac{\lambda}{\mu^2} + \alpha\frac{1}{\mu}}{1+\lambda W^* + \frac{\lambda}{\mu}} \tag{17}$$

□

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

$$J_0 = \frac{1}{\mu} + \frac{\alpha/\mu}{1+\lambda/\mu} \tag{18}$$

which is a useful basis for performance comparison. To further explore the benefit of batching when Eq. 15 holds, let $\Delta = J_0 - J^*$. We normalize the parameters by setting $1/\mu = 1$, $1/\lambda = k$ and let

$$\tilde{\Delta}(k,\alpha) = \frac{J_0 - J^*}{J_0} \tag{19}$$

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

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, W^*, J^* through Eqs. 16–18, and finally $\tilde{\Delta}$ in Eq. 19. 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.

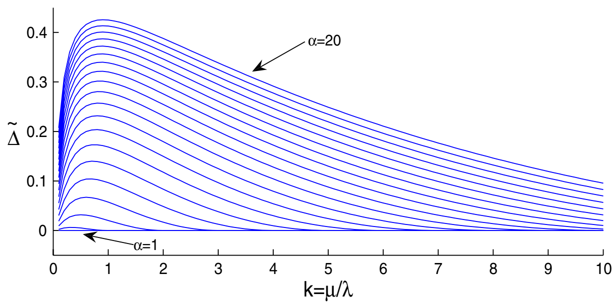


Fig. 4 Relative optimal message batching benefits under different k and α settings

Remark 1 For queue-length-based batching, we can use a similar analysis to derive the optimal batching size under Poisson process assumptions. Interested readers are referred to Ning and Cassandras (2008).

4 On-line gradient estimation and optimization

The analytical model breaks down when the Markovian assumptions of the previous section are relaxed. Moreover, in practice network statistics are not known in advance, which calls for a method designed to determine the optimal batching time parameter W without explicit knowledge of statistical information on the network traffic. Therefore, we propose an on-line gradient estimation method for time-based message batching based on Perturbation Analysis (PA) (Cassandras and Lafortune 1999; Fu and Hu 1997). In particular, we attempt to extract from an observed sample path not only performance data, but also sensitivity information with respect to the controllable parameter W. This gradient information will be used to obtain the optimal value W^* using a Stochastic Approximation algorithm (Kushner and George Yin 2003).

Consider the performance objective $J(W)$ in Eq. 12 and, for an appropriately defined probability space, let $L_D(W,\omega)$, $L_Q(W,\omega)$ be *sample functions* over an observed sample path denoted by ω and consisting of a fixed number of arrivals, N, and B batching periods (where B depends on N):

$$L_D(W,\omega) = \frac{1}{N}\sum_{j=1}^{N} D_j(W,\omega) \tag{20}$$

$$L_Q(W,\omega) = \frac{1}{N}\sum_{i=1}^{B} P_i(W,\omega) \tag{21}$$

In addition, let

$$L(W,\omega) = L_D(W,\omega) + \alpha L_Q(W,\omega) \tag{22}$$

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

$$\frac{dJ(W)}{dW} = \frac{dE[L(W,\omega)]}{dW} = \lim_{\Delta W\to 0}\frac{E[\Delta L(W,\omega)]}{\Delta W} \tag{23}$$

4.1 Smoothed perturbation analysis

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

$$\left[\frac{dJ(W)}{dW}\right]_{IPA} = \frac{dL(W,\omega)}{dW}$$

However, in our problem, IPA fails to produce an unbiased estimator since the sample function $L(W,\omega)$ is not continuous with respect to W for every possible ω (a detailed analysis of this can be found in Ning and Cassandras 2008.) Therefore, one cannot exchange the limit and expectation in Eq. 23. This motivates the use of Smoothed Perturbation Analysis (SPA) (Gong and Ho 1987).

The main idea in SPA is to replace the original sample function $L(W, \omega)$ by a conditional expectation $E[L(W, \omega)|\mathbf{Z}]$, i.e., set

$$J(W) = E[L(W, \omega)] = E[E[L(W, \omega)|\mathbf{Z}]]$$

where $\mathbf{Z}$ is called a *characterization* of a sample path (Gong and Ho 1987) of the process based on which $J(W)$ is evaluated. $\mathbf{Z}$ is defined as a set of random variables whose value is observed in a sample path, and it must be appropriately selected. Since, generally, $E[L(W, \omega)|\mathbf{Z}]$ is "smoother" than $L(W, \omega)$, we expect to be able to interchange the expectation and the limit in Eq. 23, obtaining an SPA estimator:

$$\left[\frac{dJ(W)}{dW}\right]_{SPA} = \lim_{\Delta W \to 0} \frac{E[\Delta L(W, \Delta W, \omega)|\mathbf{Z}]}{\Delta W} \tag{24}$$

where $\Delta L(W, \Delta W, \omega)$ is defined as the difference between the sample function values on the nominal and perturbed sample paths. In the following analysis, we only consider

$$\Delta L(W, \Delta W, \omega) = L(W, \omega) - L(W - \Delta W, \omega)$$

where $\Delta W > 0$, so we are interested in estimating the *left derivative* $dJ(W)/dW^-$.

Looking at Eq. 24, we see that the selection of $\mathbf{Z}$ is crucial, as a different $\mathbf{Z}$ results in a different form of the SPA estimator. In our case, we take advantage of the sample path structure described in previous sections. Specifically, a sample path is decomposed into batching periods based on which all subsequent analysis is performed. We define $\mathbf{Z}$ to include BP information as follows:

$$\mathbf{Z} = \left\{B, I_i, R_i, A_{I_i}, S_{R_i}, n_i, \quad i = 1, \ldots, B\right\} \tag{25}$$

In words, $\mathbf{Z}$ contains the information identifying when each BP starts and ends, as well as the total number of BPs, i.e., B, and the number of sampling events within each BP_i, i.e., n_i (but not their occurrence times).

Because in the perturbed sample path, the batching time W is reduced by ΔW, two cases can occur. If there does not exist a sampling event during the ΔW interval, all the message delivery time and the batching period structure remain intact. However, if there is a sampling event during the ΔW interval, the resulting perturbed sample path will have a different batching period structure, as shown in Fig. 5. Therefore, it is important to evaluate the probability of such event happening. The following lemma describes how this probability is computed. Before we proceed with the lemma, we introduce the following definitions. Define $g(\cdot)$ as the p.d.f. of the random inter-sampling time X, and $g_n(\cdot)$ is the n-fold convolution of $g(\cdot)$:

$$g_1(x) = g(x)$$
$$g_n(x) = (g * g_{n-1})(x), \quad n \geq 2$$

where the convolution operation $*$ is defined as:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(u)\, g(x - u)\, du$$

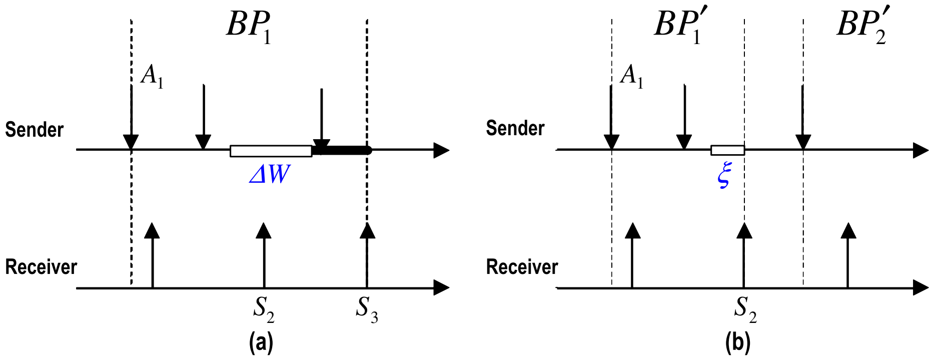


Fig. 5 If there exists a sampling event during the ΔW interval (**a**), one batching period is split into two (**b**)

Also define

$$H_n(U, P) = \frac{g(P)\, g_{n-1}(U-P)}{\int_P^U g(s)\, g_{n-1}(U-s)\, ds}, \quad n = 2, 3, \ldots \tag{26}$$

$$H_1(U, P) = 0$$

We can now proceed with the following lemma:

Lemma 2 *Assume that the sampling process is a general renewal process and let X be an inter-sampling time with c.d.f. $G(\cdot)$ and p.d.f. $g(\cdot)$. Assume that $g(x) < \infty$ for all x, and $E[X] < \infty$. Let e_i denote the event that for an arbitrary batching period BP_i, there exists a sampling event in the interval $[W - \Delta W, W]$. Then, $\Pr(e_i)$ is of order $O(\Delta W)$ and is given by*

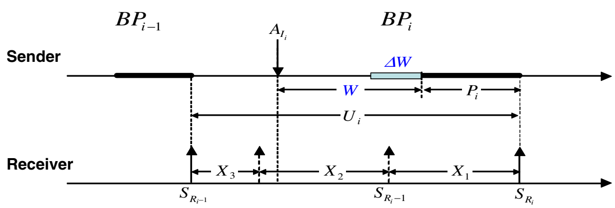
$$\Pr(e_i) = E\left[H_{n_i}(U_i, P_i)\right] \Delta W + o(\Delta W) \tag{27}$$

where U_i, P_i are defined as (see Fig. 6):

$$P_i = S_{R_i} - \left(A_{I_i} + W\right)$$

$$U_i = S_{R_i} - S_{R_{i-1}}$$

Fig. 6 Computing the probability that a sampling event occurs within the ΔW interval



Proof Figure 6 illustrates an occurrence of event e_i. Note that the evaluation of $\Pr(e_i)$ can be facilitated by conditioning on some random quantities in the sample path. In particular, for a $BP[A_{I_i}, S_{R_i}]$, we can condition on $S_{R_{i-1}}, S_{R_i}, A_{I_i}$, as well as n_i defined to be the number of sampling events in the interval $(S_{R_{i-1}}, S_{R_i}]$; for example, in Fig. 6 we can see that $n_i = 3$. Then, $\Pr(e_i)$ can be expressed as

$$\Pr(e_i) = E\left[\Pr\left(e_i | S_{R_{i-1}}, S_{R_i}, A_{I_i}, n_i\right)\right] \tag{28}$$

In simple terms, we fix two sampling events $S_{R_{i-1}}$ and S_{R_i} which delimits the half-open interval $(S_{R_{i-1}}, S_{R_i}]$. Due to the renewal property of the sampling process, sampling events outside this interval are independent. In addition to $S_{R_{i-1}}$ and S_{R_i}, there are $n_i - 1$ additional sampling events, whose occurrence time is random within this interval. We first notice that if $n_i = 1$, i.e., there is no such "movable" sampling event, so that

$$\Pr\left(e_i | S_{R_{i-1}}, S_{R_i}, A_{I_i}, n_i = 1\right) = 0$$

If $n_i > 1$, given ΔW, there is a non-zero probability that one such event will be within the ΔW interval. To compute this conditional probability, we notice that no additional sampling event can occur during P_i, otherwise the batching period would have ended earlier than S_{R_i}. Let $X_1 = S_{R_i} - S_{R_i - 1}$ and let X_j, $j = 2, \ldots, n_i$, be the remaining inter-sampling intervals. Therefore, when $n_i > 1$,

$$\begin{aligned}
\Pr\left(e_i | S_{R_{i-1}}, S_{R_i}, A_{I_i}, n_i\right) &= \Pr\left(X_1 \leq P_i + \Delta W | X_1 > P_i, \sum_{j=1}^{n_i} X_j = U_i\right) \\
&= \lim_{\Delta U \downarrow 0} \Pr\left(X_1 \leq P_i + \Delta W | X_1 > P_i, U_i < \sum_{j=1}^{n_i} X_j \leq U_i + \Delta U\right) \\
&= \lim_{\Delta U \downarrow 0} \frac{\Pr\left(P_i < X_1 \leq P_i + \Delta W, U_i - X_1 < \sum_{j=2}^{n_i} X_j \leq U_i - X_1 + \Delta U\right)}{\Pr\left(X_1 > P_i, U_i - X_1 < \sum_{j=2}^{n_i} X_j \leq U_i - X_1 + \Delta U\right)}
\end{aligned}$$

Because X_1 and X_j, $j = 2, \ldots, n_i$, are all mutually independent, using the total probability theorem on both denominator and numerator,

$$\Pr\left(e_i | S_{R_{i-1}}, S_{R_i}, A_{I_i}, n_i\right) = \lim_{\Delta U \downarrow 0} \frac{\int_{P_i}^{P_i + \Delta W} g(s) \int_{U_i - s}^{U_i - s + \Delta U} g_{n_i - 1}(t)\, dtds}{\int_{P_i}^{U_i} g(s) \int_{U_i - s}^{U_i - s + \Delta U} g_{n_i - 1}(t)\, dtds}$$

When $\Delta U \downarrow 0$, the mean-value theorem yields

$$\begin{aligned}
&\Pr\left(e_i | S_{R_{i-1}}, S_{R_i}, A_{I_i}, n_i\right) \\
&\quad = \lim_{\Delta U \downarrow 0} \frac{g(P_i)\, g_{n_i - 1}(U_i - P_i)\, \Delta W \Delta U + o(\Delta W)\, \Delta U + o(\Delta U)\, \Delta W}{\int_{P_i}^{U_i} g(s)\, g_{n_i - 1}(U_i - s)\, \Delta U ds} \\
&\quad = \frac{g(P_i)\, g_{n_i - 1}(U_i - P_i)\, \Delta W}{\int_{P_i}^{U_i} g(s)\, g_{n_i - 1}(U_i - s)\, ds} + o(\Delta W)
\end{aligned}$$

Since U_i, P_i, n_i do not depend on ΔW and recalling Eq. 28, we get

$$\Pr(e_i) = E\left[H_{n_i}(U_i, P_i)\right]\Delta W + o(\Delta W) \sim O(\Delta W)$$

$\square$

In Lemma 2 we found that $\Pr(e_i) \sim O(\Delta W)$. Let $\bar{e}_i$ be e_i's complement. Since we have assumed that the inter-sampling times are mutually independent, e_i and e_j, $j \neq i$, are independent, hence the probability that two or more such events occurring is of higher order than $O(\Delta W)$, so we set $\Pr\left(e_i e_j \cdots\right) = o(\Delta W)$. By conditioning on all possible combinations of e_i and $\bar{e}_i$, $i = 1, \ldots, B$, and using the total probability theorem, we obtain:

$$\begin{aligned}
&E\left[\Delta L(W, \Delta W, \omega) | \mathbf{Z}\right] \\
&\quad= E\left[\Delta L(W, \Delta W, \omega) | \bar{e}_i, i = 1, \ldots, B, \mathbf{Z}\right] \Pr\left(\bar{e}_i, i = 1, \ldots, B | \mathbf{Z}\right) \\
&\qquad+ \sum_{i=1}^{B} E\left[\Delta L(W, \Delta W, \omega) | e_i, \bar{e}_j, j \neq i, \mathbf{Z}\right] \Pr\left(e_i, \bar{e}_j, j \neq i | \mathbf{Z}\right) \\
&\qquad+ \sum_{i=1}^{B} \sum_{j=1, j\neq i}^{B} E\left[\Delta L(W, \Delta W, \omega) | e_i e_j, \bar{e}_k, k \neq i, k \neq j, \mathbf{Z}\right] \Pr\left(e_i e_j, \bar{e}_k, k \neq i, k \neq j | \mathbf{Z}\right) \\
&\qquad+ \sum_{i=1}^{B} \sum_{j=1, j\neq i}^{B} \sum_{k=1, k\neq i, k\neq j}^{B} \ldots + E\left[\Delta L(W, \Delta W, \omega) | e_i, i = 1 \ldots B, \mathbf{Z}\right] \Pr\left(e_1 e_2 \ldots e_B | \mathbf{Z}\right) \\
&\quad= E\left[\Delta L(W, \Delta W, \omega) | \bar{e}_i, i = 1, \ldots, B, \mathbf{Z}\right] \Pr\left(\bar{e}_i, i = 1, \ldots, B | \mathbf{Z}\right) \\
&\qquad+ \sum_{i=1}^{B} E\left[\Delta L(W, \Delta W, \omega) | e_i, \bar{e}_j, j = 1, \ldots, B, j \neq i, \mathbf{Z}\right] \\
&\qquad\times \Pr\left(e_i, \bar{e}_j, j = 1, \ldots, B, j \neq i | \mathbf{Z}\right) + o(\Delta W)
\end{aligned} \tag{29}$$

This long equation simply states that, conditioned on $\mathbf{Z}$, $\Delta L(W, \Delta W, \omega)$ can be decomposed by BPs and events e_i for all i. In the first term

$$E\left[\Delta L(W, \Delta W, \omega) | \bar{e}_i, i = 1 \ldots B, \mathbf{Z}\right]$$

is the contribution to ΔL when no e_i event occurs. The second term is a sum over all BPs, where

$$E\left[\Delta L(W, \Delta W, \omega) | e_i, \bar{e}_j, j = 1, \ldots, B, j \neq i, \mathbf{Z}\right]$$

is the contribution when only e_i occurs in the ith BP but no other e_j, $j \neq i$ occurs. The third term is a double sum of the contributions when only two events, e_i and e_j, occur, and so on. By the mutual independence of the e_i events, terms other than the first two are of $o(\Delta W)$, which results in Eq. 29. Now, since we are only interested in $B + 1$ conditions, i.e., no e_i ever occurs and exactly one e_i occurs over B BPs, we define for notational simplicity:

$$\begin{aligned}
&\bar{e} \triangleq \{\bar{e}_i, i = 1 \ldots B\} && \text{No } e_i \text{ occurs in any } BP_i \\
&\hat{e}_i \triangleq \left\{e_i, \bar{e}_j, j = 1, \ldots, B, j \neq i\right\} && \text{Only } e_i \text{ occurs in some } BP_i
\end{aligned}$$

Then, to determine the SPA estimator, we need to evaluate the following four quantities: (*i*) $\Pr\left(\hat{e}_i|\mathbf{Z}\right)$, (*ii*) $\Pr\left(\bar{e}|\mathbf{Z}\right)$, (*iii*) $E\left[\Delta L\left(W, \Delta W, \omega\right)|\hat{e}_i, \mathbf{Z}\right]$, and (*iv*) $E\left[\Delta L\left(W, \Delta W, \omega\right)|\bar{e}, \mathbf{Z}\right]$. To compute $\Pr\left(\hat{e}_i|\mathbf{Z}\right)$ we can directly use the result obtained in Lemma 2:

$$\Pr\left(\hat{e}_i|\mathbf{Z}\right) = H_{n_i}\left(U_i, P_i\right)\Delta W + o\left(\Delta W\right) \tag{30}$$

where U_i, P_i and n_i are all known given $\mathbf{Z}$. Since the probability of two or more e_i events occurring is of $o\left(\Delta W\right)$, we conclude that

$$\begin{aligned}\Pr\left(\bar{e}|\mathbf{Z}\right) &= 1 - \sum_{i=1}^{B}\Pr\left(\hat{e}_i|\mathbf{Z}\right) + o\left(\Delta W\right)\\ &= 1 - \sum_{i=1}^{B} H_{n_i}\left(U_i, P_i\right)\Delta W + o\left(\Delta W\right)\end{aligned}$$

The following two sections will derive $E\left[\Delta L\left(W, \Delta W, \omega\right)|\hat{e}_i, \mathbf{Z}\right]$ and $E\left[\Delta L\left(W, \Delta W, \omega\right)|\bar{e}, \mathbf{Z}\right]$ for both average delay and average preamble metrics.

4.1.1 SPA for the average delay metric

Define $\Delta L_D\left(W, \Delta W, \omega\right)$ as the difference in the delay metric defined in Eq. 20:

$$\Delta L_D\left(W, \Delta W, \omega\right) = L_D\left(W, \omega\right) - L_D\left(W - \Delta W, \omega\right)$$

We first focus on evaluating $E\left[\Delta L_D\left(W, \Delta W, \omega\right)|\bar{e}, \mathbf{Z}\right]$. First, we know that if event e_i does not occur in BP_i, there can be no change in the delay of any message in BP_i. Therefore,

$$E\left[\Delta L_D\left(W, \Delta W, \omega\right)|\bar{e}, \mathbf{Z}\right] = 0 \tag{31}$$

The more interesting case lies in the evaluation of $E\left[\Delta L_D\left(W, \Delta W, \omega\right)|\hat{e}_i, \mathbf{Z}\right]$. We illustrate this case through an example as shown in Fig. 7. Clearly, batching periods prior to BP_i are not affected by event e_i. However, because sampling event S_{R_i-1} (not measurable w.r.t. $\mathbf{Z}$) occurs in the interval $[W - \Delta W, W]$, BP_i ends prematurely at S_{R_i-1} resulting in a shortened $BP'_{i,1}$ (see Fig. 7b). Then, a new batching period $BP'_{i,2}$ is initiated by a message $A_{I'_{i,2}}$ which originally lies within BP_i in the nominal sample path. We can see that a new batching period structure evolves in the perturbed sample path with arrival and sampling events that are not necessarily the critical events of the nominal sample path. Finally, notice that in the perturbed sample path, at $BP'_{i,4}$, the initializing event $A_{I'_{i,4}}$ coincides with $A_{I_{i+2}}$, which is a critical arrival event in the nominal sample path. This concludes the effect of the perturbation due to e_i because from $BP'_{i,4}$ (BP_{i+2}) onward the perturbed sample path is identical to the nominal sample path; therefore, the delays of all messages for BP_{i+2} onward are unaffected. To summarize:

1. Event $\hat{e}_i$ has no effect on the BP structure of the sample path prior to BP_i.
2. Event $\hat{e}_i$ results in a different BP structure from BP_i onward.
3. When in the perturbed sample path, an arrival event initializing some BP is also a critical arrival in the nominal sample path, the perturbed and nominal sample path BPs are *synchronized* and no further difference exists.

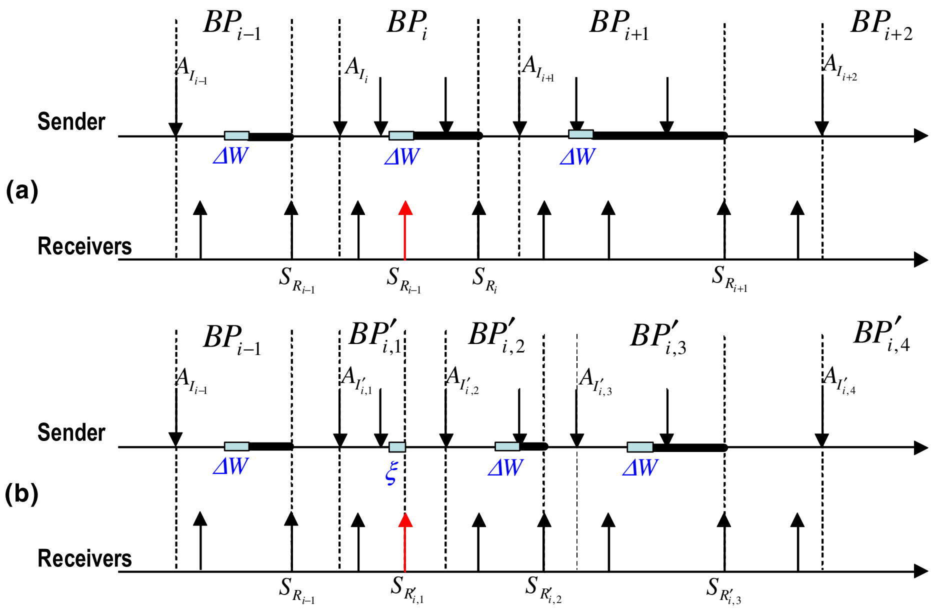


Fig. 7 **a** An example of a sample path with ΔW perturbation intervals overlaid. Event e_i occurs in BP_i. **b** The perturbed sample path of **a**. Notice the change in busy period structure

Therefore, the key to evaluating $\Delta L_D(W, \Delta W, \omega)$ given $\hat{e}_i$ is to identify the new BP structure until synchronization. We can see that this evaluation will require the following quantities: $\{A_j,\ j \geq I_i\}$ and $\{S_j,\ j \geq R_i\}$. Note that we may not need all these quantities, depending on whether synchronization takes place or not before the end of the observed sample path. Now define a set $\mathbf{Z}_i$ as follows:

$$\mathbf{Z}_i = \left\{A_j,\ j \geq I_i\right\} \cup \left\{S_j,\ j \geq R_i\right\} \tag{32}$$

which is called a *sub-characterization*. Therefore, by conditioning on $\mathbf{Z}_i$, we have:

$$E\left[\Delta L_D(W, \Delta W, \omega)\,|\hat{e}_i, \mathbf{Z}\right] = E\left[E\left[\Delta L_D(W, \Delta W, \omega)\,|\mathbf{Z}_i\right]|\hat{e}_i, \mathbf{Z}\right] \tag{33}$$

With $\mathbf{Z}_i$ given, we can now *reconstruct* the BP structure resulting from event e_i and hence evaluate $\Delta L_D(W, \Delta W, \omega)$ above. Let $\{I'_{i,k},\ k \geq 1\}$ and $\{R'_{i,k},\ k \geq 1\}$ be the index sets of critical events in the new batching periods on the perturbed path. Their values can be obtained in a recursive fashion similar to Eqs. 1–3. To initialize we have:

$$I'_{i,1} = I_i \tag{34}$$

$$S_{R'_{i,1}} = A_{I_i} + (W - \Delta W) + \xi \tag{35}$$

for some $\xi \in [0, \Delta W]$. Moreover, the indices of the sampling event ending the kth BP and of the arrival event starting the $(k+1)$th BP in the perturbed path are given by

$$R'_{i,k} = \min_j \left\{ j : S_j > A_{I'_k} + W \right\}, \; k \geq 2 \tag{36}$$

$$I'_{i,k+1} = \min_j \left\{ A_j : A_j > S_{R'_{i,k}} \right\}, \quad k \geq 1 \tag{37}$$

This procedure is carried out through the $\left[l(i) - 1\right]$th batching period, where $BP'_{i,l(i)}$ is the BP synchronized to the nominal sample path (if one exists). Therefore, $A_{I'_{i,l(i)}}$ coincides with some A_{I_j} in the nominal sample path, i.e.,

$$I'_{i,l(i)} \in \left\{ I_j : j > i \right\} \tag{38}$$

as illustrated in Fig. 7 with $I_{i+2} = I'_{i,4}$. The affected messages are thus $\{A_j : I_i \leq j < I'_{i,l(i)}\}$. Denote by $D'_j(W, \Delta W, \omega)$ the new delay of the jth message. Similar to Eq. 4, it is easy to see that

$$D'_j(W, \Delta W, \omega) = \min_k \left\{ S_{R'_{i,k}} : S_{R'_{i,k}} > A_j \right\} - A_j, \quad I_i \leq j < I'_{i,l(i)} \tag{39}$$

Note that, $D'_j(W, \Delta W, \omega)$ depends on ΔW. However, it is easy to see that only messages in $BP'_{i,1}$ are affected because $S_{R'_{i,1}}$ contains the effect of ΔW and ξ. Since we are interested in the limiting behavior as $\Delta W \to 0$ in Eq. 24, hence $\xi \to 0$, in the limit we have

$$S_{R'_{i,1}} = A_{I_i} + W \tag{40}$$

Therefore, we define

$$D'_j(W, \omega) = \lim_{\Delta W \downarrow 0} D'_j(W, \Delta W, \omega), \quad I_i \leq j < I'_{l(i)} \tag{41}$$

whose value in the estimator is actually exactly the same as $D'_j(W, \Delta W, \omega)$ but using Eq. 40 for $S_{R'_{i,1}}$ instead. Therefore, given $e_i, \bar{e}_j, \; j = 1, \ldots, B, \; j \neq i, \mathbf{Z}_i$, the difference in the delay metric is

$$\begin{aligned} E\left[\Delta L_D(W, \Delta W, \omega) \mid \hat{e}_i, \mathbf{Z}_i, \mathbf{Z}\right] &= \frac{1}{N} \sum_{j=1}^{N} \left[D_j(W, \omega) - D'_j(W, \Delta W, \omega) \right] \\ &= \frac{1}{N} \sum_{I_i \leq j < I'_{i,l(i)}} \left[D_j(W, \omega) - D'_j(W, \Delta W, \omega) \right] \end{aligned} \tag{42}$$

where $D'_j(W, \Delta W, \omega)$ is computable from the observed sample path data through Eqs. 39 and 34–38.

We are now in a position to prove the following theorem:

Theorem 2 *Let $J_D(W) = E[L_D(W, \omega)]$ and assume that (i) $E[X] < \infty$ where X is the inter-sampling time, and (ii) $g(\cdot) < \infty$ within its whole support region. An unbiased SPA estimator of the left derivative $dJ_D(W)/dW^-$ is given by:*

$$\left[\frac{dJ_D(W)}{dW^-} \right]_{SPA} = \frac{1}{N} \sum_{i=1}^{B} \sum_{I_i \leq j < I'_{l(i)}} \left[D_j(W, \omega) - D'_j(W, \omega) \right] H_{n_i}(U_i, P_i) \tag{43}$$

Proof By definition and from Eq. 29,

$$\frac{dJ_D(W)}{dW^-} = \lim_{\Delta W \downarrow 0} \frac{E[\Delta L_D(W, \Delta W, \omega)]}{\Delta W} = \lim_{\Delta W \downarrow 0} \frac{E[E[\Delta L_D(W, \Delta W, \omega)|\mathbf{Z}]]}{\Delta W}$$

where $\mathbf{Z}$ is the characterization defined in Eq. 25: $\mathbf{Z} = \{B, I_i, R_i, A_{I_i}, S_{R_i}, n_i,\ i = 1, \ldots, B\}$. Given $\mathbf{Z}$, the BP structure is known. Recalling Eq. 29, we have:

$$E[\Delta L_D(W, \Delta W, \omega)|\mathbf{Z}] = E[\Delta L_D(W, \Delta W, \omega)|\bar{e}, \mathbf{Z}] \Pr(\bar{e}|\mathbf{Z}) + \sum_{i=1}^{B} E\left[\Delta L_D(W, \Delta W, \omega)|\hat{e}_i, \mathbf{Z}\right] \Pr\left(\hat{e}_i, \mathbf{Z}\right) + o(\Delta W)$$

Since the first term is zero due to Eq. 31,

$$E[\Delta L_D(W, \Delta W, \omega)|\mathbf{Z}] = \sum_{i=1}^{B} E\left[\Delta L_D(W, \Delta W, \omega)|\hat{e}_i, \mathbf{Z}\right] \Pr\left(\hat{e}_i|\mathbf{Z}\right) + o(\Delta W)$$

and we have

$$\frac{dJ_D(W)}{dW^-} = \lim_{\Delta W \downarrow 0} \frac{1}{\Delta W} E\left[\sum_{i=1}^{B} E\left[\Delta L_D(W, \Delta W, \omega)|\hat{e}_i, \mathbf{Z}\right] \Pr\left(\hat{e}_i|\mathbf{Z}\right) + o(\Delta W)\right] \quad (44)$$

To derive the SPA estimator, we need to exchange the order of limit and expectation. Since expectation is an integral, Lebesgue's dominated convergence theorem provides a sufficient condition that allows the two limit processes to commute. Therefore, we need to show that there exists $\varepsilon > 0$ such that

$$E\left\{\sup_{0 \le \Delta W \le \varepsilon} \left|\frac{1}{\Delta W} \sum_{i=1}^{B} E\left[\Delta L_D(W, \Delta W, \omega)|\hat{e}_i, \mathbf{Z}\right] \Pr\left(\hat{e}_i|\mathbf{Z}\right) + o(\Delta W)\right|\right\} < \infty \quad (45)$$

In Lemma 2 we have shown that $\Pr\left(\hat{e}_i|\mathbf{Z}\right) \sim O(\Delta W)$, that is, there exists $K_1 > 0$ and $\varepsilon_0 > 0$ such that when $\Delta W < \varepsilon_0$, $\Pr\left(\hat{e}_i|\mathbf{Z}\right) \le K_1 \Delta W$. From Eq. 33 we know that a sufficient condition for

$$\left|E\left[\Delta L_D(W, \Delta W, \omega)|\hat{e}_i, \mathbf{Z}\right]\right| < \infty$$

is that, for all possible $\mathbf{Z}_i$,

$$\left|E\left[\Delta L_D(W, \Delta W, \omega)|\hat{e}_i, \mathbf{Z}_i, \mathbf{Z}\right]\right| < \infty$$

where, recalling Eq. 42,

$$E\left[\Delta L_D(W, \Delta W, \omega)|\hat{e}_i, \mathbf{Z}_i, \mathbf{Z}\right] = \frac{1}{N} \sum_{I_i \le j < I'_{i,l(i)}} \left[D_j(W, \omega) - D'_j(W, \Delta W, \omega)\right]$$

Since a sample path contains a *finite* number N of messages and B batching periods, we have

$$\left|D_j(W, \omega) - D'_j(W, \Delta W, \omega)\right| < S_{R_B}$$

where $E\left[S_{R_B}\right] = BE[X] < \infty$ and does not depend on ΔW. Let $E\left[S_{R_B}\right] = K_2(N) < \infty$. In addition,

$$I'_{i,l(i)} - I_i \le N$$

Therefore,

$$\left| E\left[\Delta L_D\left(W, \Delta W, \omega\right)|\hat{e}_i, \mathbf{Z}\right] \right| \leq K_2\left(N\right) < \infty$$

and it follows that

$$\frac{1}{\Delta W}\left| E\left[\Delta L_D\left(W, \Delta W, \omega\right)|\hat{e}_i, \mathbf{Z}\right] \Pr\left(\hat{e}_i|\mathbf{Z}\right) \right| \leq K_1 K_2\left(N\right)$$

Let $R\left(\Delta W\right)$ be a ΔW-dependent term of order $o\left(\Delta W\right)$. There exists some ε_i and constant $C < \infty$ such that for all $\Delta W < \varepsilon_i$, $|R\left(\Delta W\right)| \leq C\Delta W$. Let $\varepsilon = \min_i \varepsilon_i$, and since $B \leq N$, we obtain,

$$\begin{aligned} &E\left\{ \sup_{0 \leq \Delta W \leq \varepsilon} \left| \sum_{i=1}^{B} E\left[\Delta L_D\left(W, \Delta W, \omega\right)|\hat{e}_i, \mathbf{Z}\right] H\left(U_i, P_i\right) + \frac{o\left(\Delta W\right)}{\Delta W} \right| \right\} \\ &\quad \leq N K_1 K_2\left(N\right) + C < \infty \end{aligned}$$

Using the dominated convergence theorem, we have from Eqs. 44 and 42:

$$\begin{aligned} \frac{dJ_D\left(W\right)}{dW^-} &= E\left[\sum_{i=1}^{B} \lim_{\Delta W \downarrow 0} E\left[\Delta L_D\left(W, \Delta W, \omega\right)|\hat{e}_i, \mathbf{Z}\right] \Pr\left(\hat{e}_i|\mathbf{Z}\right)\right] \\ &= E\left[\sum_{i=1}^{B} \lim_{\Delta W \downarrow 0} E_{\mathbf{Z}_i}\left[\frac{1}{N} \sum_{I_i \leq j < I'_{i,I(i)}} \left[D_j\left(W, \omega\right) - D'_j\left(W, \Delta W, \omega\right)\right]\right] H_{n_i}\left(U_i, P_i\right)\right] \end{aligned}$$

where we use the notation $E_{\mathbf{Z}_i}$ to remind ourselves that each expectation within the outer sum is over the sub-characterization $\mathbf{Z}_i$. Using the dominated convergence theorem once again, we can move the limit inside this expectation and make use of Eq. 41 to get

$$\frac{dJ_D\left(W\right)}{dW^-} = E\left\{\sum_{i=1}^{B} E_{\mathbf{Z}_i}\left[\frac{1}{N} \sum_{I_i \leq j < I'_{i,I(i)}} \left[D_j\left(W, \omega\right) - D'_j\left(W, \omega\right)\right]\right] H_{n_i}\left(U_i, P_i\right)\right\} \tag{46}$$

Note that $H_{n_i}\left(U_i, P_i\right)$ does not depend on $\mathbf{Z}_i$ (which includes arrival and sampling events *after* BP_i). Moreover, $E\left[\Delta L_D\left(W, \Delta W, \omega\right)|e_i, \mathbf{Z}_i, \mathbf{Z}\right]$ does not depend on events in $\mathbf{Z}_i \backslash \mathbf{Z}$ (i.e., the arrival and sampling events before BP_i). Therefore, we can rewrite the expectation in Eq. 46 as

$$\frac{dJ_D\left(W\right)}{dW^-} = E\left\{\frac{1}{N} \sum_{i=1}^{B} \sum_{I_i \leq j < I'_{i,I(i)}} \left[D_j\left(W, \omega\right) - D'_j\left(W, \omega\right)\right] H_{n_i}\left(U_i, P_i\right)\right\}$$

where the expectation is over all random variables in $\mathbf{Z}$ and $\mathbf{Z}_1, \ldots, \mathbf{Z}_B$. The quantity inside the expectation is the SPA estimator defined in Eq. 43, which is, therefore, unbiased. □

4.1.2 SPA for the average preamble metric

Define $\Delta L_Q\left(W, \Delta W, \omega\right)$ as the difference in the preamble metric defined in Eq. 21:

$$\Delta L_Q\left(W, \Delta W, \omega\right) = L_Q\left(W, \omega\right) - L_Q\left(W - \Delta W, \omega\right)$$

Recall from Eq. 29 that:

$$E\left[\Delta L_Q(W,\Delta W,\omega)|\mathbf{Z}\right] = E\left[\Delta L_Q(W,\Delta W,\omega)|\bar{e},\mathbf{Z}\right]\Pr(\bar{e}|\mathbf{Z}) \\ +\sum_{i=1}^{B} E\left[\Delta L_Q(W,\Delta W,\omega)|\hat{e}_i,\mathbf{Z}\right]\Pr\left(\hat{e}_i|\mathbf{Z}\right)+o(\Delta W) \tag{47}$$

Again, we first focus on evaluating $E\left[\Delta L_Q(W,\Delta W,\omega)|\bar{e},\mathbf{Z}\right]$. First, we know that if event e_i does not occur in BP_i, there will be no change in the BP structure. Therefore, the only difference is that in each of the B batching periods, the preamble time is extended by ΔW, i.e.,

$$E\left[\Delta L_Q(W,\Delta W,\omega)|\bar{e},\mathbf{Z}\right] = -\frac{B\Delta W}{N} \tag{48}$$

Next, we evaluate $E\left[\Delta L_Q(W,\Delta W,\omega)|\hat{e}_i,\mathbf{Z}\right]$ for each BP_i. We illustrate this case through the example shown in Fig. 7. When e_i occurs, we need to reconstruct a segment of new BPs until synchronization, and then calculate the difference. This reconstruction is the same as in the previous subsection. Therefore, we choose the same sub-characterization $\mathbf{Z}_i$:

$$\mathbf{Z}_i = \left\{A_j,\ j\geq I_i\right\}\cup\left\{S_j,\ j\geq R_i\right\}$$

and by conditioning on $\mathbf{Z}_i$ we get:

$$E\left[\Delta L_Q(W,\Delta W,\omega)|\hat{e}_i,\mathbf{Z}\right] = E\left\{E\left[\Delta L_Q(W,\Delta W,\omega)\ |\ \hat{e}_i,\mathbf{Z}_i\right]\ |\ \mathbf{Z}\right\} \tag{49}$$

With $\mathbf{Z}_i$ given, the reconstruction is done using Eqs. 34–38. Next, we evaluate $\Delta L_Q(W,\Delta W,\omega)$ in Eq. 49. First, notice that e_i only affects BPs beginning with BP_i and until synchronization takes place at $BP'_{i,l(i)}$ in the perturbed sample path when it coincides with $BP_{m(i)}$ in the nominal sample path. From BP_i to $BP_{m(i)}$ (corresponding to from $BP'_{i,1}$ to $BP'_{i,l(i)}$ in the perturbed sample path), due to the reconstruction, we need to evaluate the new preamble time as follows:

$$P'_{i,j}(W,\Delta W,\omega) = S_{R'_{i,j}} - A_{I'_{i,j}} + W - \Delta W,\quad 1\leq j< l(i)$$

In the limit when $\Delta W\to 0$, we have

$$P'_{i,j}(W,\omega) = S_{R'_{i,j}} - A_{I'_{i,j}} + W,\quad 1\leq j< l(i) \tag{50}$$

Therefore, from BP_i to $BP_{m(i)}$ the difference in preamble time is:

$$\sum_{k=i}^{m-1} P_i - \sum_{k=1}^{l(i)-1} P'_{i,k}(W,\Delta W,\omega)$$

while from BP_1 to BP_{i-1} and from $BP_{m(i)}$ to BP_B, the difference in the preamble time is ΔW for each BP, totaling:

$$-(i+B-m(i))\Delta W$$

so that

$$E\left[\Delta L_Q(W,\Delta W,\omega)\ |\ \hat{e}_i,\ \mathbf{Z}_i,\mathbf{Z}\right]$$
$$=\frac{1}{N}\left[-(i+B-m(i))\,\Delta W+\sum_{k=i}^{m(i)-1}P_i(W,\omega)-\sum_{k=1}^{l(i)-1}P'_{i,k}(W,\Delta W,\omega)\right] \quad (51)$$

Theorem 3 *Let $J_Q(W)=E\left[L_Q(W,\omega)\right]$ and assume that (i) $E[X]<\infty$ where X is the inter-sampling time, and (ii) $g(\cdot)<\infty$ within its whole support region. An unbiased SPA estimator of the left derivative $dJ_Q(W)/dW^-$ is given by:*

$$\left[\frac{dJ_Q(W)}{dW^-}\right]_{SPA}=\frac{1}{N}\sum_{i=1}^{B}\left\{\left[\sum_{k=i}^{m(i)-1}P_i(W,\omega)-\sum_{k=1}^{l(i)-1}P'_{i,k}(W,\omega)\right]H_{n_i}(U_i,P_i)-1\right\} \quad (52)$$

Proof By definition and from Eq. 29,

$$\frac{dJ_Q(W)}{dW^-}=\lim_{\Delta W\downarrow 0}\frac{E\left[\Delta L_Q(W,\Delta W,\omega)\right]}{\Delta W}=\lim_{\Delta W\downarrow 0}\frac{E\left[E\left[\Delta L_Q(W,\Delta W,\omega)\,|\mathbf{Z}\right]\right]}{\Delta W}$$

where $\mathbf{Z}$ is the characterization defined in Eq. 25 $\mathbf{Z}=\left\{B,I_i,R_i,A_{I_i},S_{R_i},n_i,\ i=1,\ldots,B\right\}$. Similar to Theorem 2, we have to show that

$$E\left\{\sup_{0\le\Delta W\le\varepsilon}\left|\frac{E\left[E\left[\Delta L_Q(W,\Delta W,\omega)\,|\mathbf{Z}\right]\right]}{\Delta W}\right|\right\}<\infty$$

in order to switch the order of limit and expectation. Since given $\mathbf{Z}$, the BP structure is known, we have:

$$E\left[\Delta L_Q(W,\Delta W,\omega)\,|\mathbf{Z}\right]=E\left[\Delta L_Q(W,\Delta W,\omega)\,|\bar{e},\mathbf{Z}\right]\Pr(\bar{e}|\mathbf{Z})$$
$$+\sum_{i=1}^{B}E\left[\Delta L_Q(W,\Delta W,\omega)\,|\hat{e}_i,\mathbf{Z}\right]\Pr\left(\hat{e}_i|\mathbf{Z}\right)+o(\Delta W)$$

In Lemma 2 we have shown that $\Pr\left(\hat{e}_i|\mathbf{Z}\right)\sim O(\Delta W)$, that is, there exists $K_0>0$ and $\varepsilon_0>0$, such that when $\Delta W<\varepsilon_0$, $\Pr\left(\hat{e}_i|\mathbf{Z}\right)\le K_0\Delta W$. The first term inside the supremum is:

$$E\left[\Delta L_Q(W,\Delta W,\omega)\,|\bar{e},\mathbf{Z}\right]\Pr(\bar{e}|\mathbf{Z})=-\frac{B\Delta W}{N}\left[1-\sum_{i=1}^{B}\Pr\left(\hat{e}_i|\mathbf{Z}\right)\right]$$
$$=-\frac{B\Delta W}{N}+o(\Delta W)$$

Clearly,

$$\frac{1}{\Delta W}E\left[\Delta L_Q(W,\Delta W,\omega)\,|\bar{e},\mathbf{Z}\right]\Pr(\bar{e}|\mathbf{Z})=-\frac{B}{N}+\frac{o(\Delta W)}{\Delta W}$$

is bounded as $\Delta W \to 0$ since $B \leq N$. Utilizing the sub-characterization $\mathbf{Z}_i$, the second term becomes

$$\sum_{i=1}^{B} E\left[\Delta L_Q(W, \Delta W, \omega) \,|\hat{e}_i, \mathbf{Z}\right] \Pr\left(\hat{e}_i|\mathbf{Z}\right)$$

$$= \sum_{i=1}^{B} E\left[E\left[\Delta L_Q(W, \Delta W, \omega) \,|\hat{e}_i, \mathbf{Z}\right] |\mathbf{Z}_i\right] \Pr\left(\hat{e}_i|\mathbf{Z}\right)$$

where, from Eq. 51,

$$E\left[\Delta L_Q(W, \Delta W, \omega) \;|\; \hat{e}_i,\; \mathbf{Z}_i, \mathbf{Z}\right]$$
$$= \frac{1}{N}\left[-(i + B - m(i))\,\Delta W + \sum_{k=i}^{m(i)-1} P_i(W, \omega) - \sum_{k=1}^{l(i)-1} P'_{i,k}(W, \Delta W, \omega)\right]$$

For any $\mathbf{Z}_i$, this is bounded in a sample path defined by a *finite* number N of messages. To see this, consider an arbitrary $P_i(W, \omega)$ which cannot exceed S_{R_B} where $E\left[S_{R_B}\right] = BE[X] < \infty$ and does not depend on ΔW. Let $E\left[S_{R_B}\right] = K_2(N) < \infty$, so that

$$\frac{1}{N}\left|\sum_{k=i}^{m(i)-1} P_i(W, \omega) - \sum_{k=1}^{l(i)-1} P'_{i,k}(W, \Delta W, \omega)\right| \leq \frac{1}{N} N K_2(N) = K_2(N)$$

Also, since $1 < i < m(i) \leq B \leq N$, we have $B - (m(i) - i) < B \leq N$. Therefore,

$$\frac{1}{N}\left|(i + B - m(i))\,\Delta W\right| \leq \frac{1}{N}(N\Delta W) = \Delta W$$

for all possible $\mathbf{Z}_i$, hence,

$$E\left[\Delta L_Q(W, \Delta W, \omega) \;|\; \hat{e}_i, \mathbf{Z}\right] \leq -\Delta W + K_2(N)$$

Let $R(\Delta W)$ be a ΔW-dependent term of order $o(\Delta W)$. There exists some ε_i and a constant $C < \infty$ such that for all $\Delta W < \varepsilon_i$, $R(\Delta W) \leq C\Delta W$. Let $\varepsilon = \min_i \varepsilon_i$, hence,

$$E\left\{\sup_{0\leq\Delta W\leq\varepsilon} \frac{1}{\Delta W}\left| E\left[\Delta L_Q(W, \Delta W, \omega) \,|\bar{e}, \mathbf{Z}\right] \Pr(\bar{e}|\mathbf{Z}) \right.\right.$$
$$\left.\left. + \sum_{i=1}^{B} E\left[\Delta L_Q(W, \Delta W, \omega) \,|\hat{e}_i, \mathbf{Z}\right] \Pr\left(\hat{e}_i|\mathbf{Z}\right) + o(\Delta W)\right|\right\}$$
$$\leq E\left\{\sup_{0\leq\Delta W\leq\varepsilon}\left| -\frac{B}{N} + NK_1\left[K_2(N) - \Delta W\right] + \frac{o(\Delta W)}{\Delta W}\right|\right\}$$
$$\leq -1 + NK_1\left[K_2(N) + \varepsilon\right] + C < \infty$$

Invoking the dominated convergence theorem twice similar to Theorem 2, we have

$$\frac{dJ_Q(W)}{dW^-} = E\left\{-\frac{B}{N} + \frac{1}{N}\sum_{i=1}^{B} E_{\mathbf{Z}_i}\left[\sum_{k=i}^{m(i)-1} P_i(W, \omega) - \sum_{k=1}^{l(i)-1} P'_{i,k}(W, \Delta W, \omega)\right] H_{n_i}(U_i, P_i)\right\}$$

As in Theorem 2, note that $H_{n_i}(U_i, P_i)$ does not depend on $\mathbf{Z}_i$ (i.e., the arrival and sampling events after BP_i), and $E\left[\Delta L_Q(W, \Delta W, \omega)\,|e_i, \mathbf{Z}_i, \mathbf{Z}\right]$ does not depend on events in $\mathbf{Z}_i \backslash \mathbf{Z}$ (i.e., the arrival and sampling events before BP_i). Thus, we can rewrite the expectation in Eq. 46 as:

$$\frac{dJ_Q(W)}{dW^-} = E\left\{\frac{1}{N}\sum_{i=1}^{B}\left[\left(\sum_{k=i}^{m(i)-1} P_i(W, \omega) - \sum_{k=1}^{l(i)-1} P'_{i,k}(W, \Delta W, \omega)\right) H_{n_i}(U_i, P_i) - 1\right]\right\}$$

where the quantity inside the expectation is the SPA estimator (52) which is, therefore, unbiased. □

4.2 Implementation of SPA

The SPA gradient estimator algorithm is described as follows:

1. Observe a sample path containing a fixed number N of arrivals and B batching periods (B depends on N).
2. For each BP_i, initialize $I'_{i,1}$ with Eq. 34 and $S'_{R_{i,1}}$ with Eq. 40, and record n_i.
   - (a) Compute $H_{n_i}(U_i, P_i)$ using Eq. 26, where $U_i = S_{R_i} - S_{R_{i-1}}$ (except for $U_1 = S_{R_1}$) and $P_i = S_{R_i} - A_{I_i} + W$.
   - (b) Use Eqs. 36–37 to partially reconstruct the perturbed sample path until synchronization (38) is met.
   - (c) Use Eq. 41 to calculate the perturbed delay D'_j, $j = I_i \le j < I'_{l(i)}$.
   - (d) Use Eq. 50 to calculate the perturbed preamble $P'_{i,k}$, $k = 1, \ldots, l(i)$.
3. Obtain the SPA derivative estimate through Eqs. 43, 52.

This SPA estimation algorithm requires information from the whole observed sample path, which may require substantial memory and may not be feasible for sensor nodes. Therefore, we propose a *serialized* version which uses less storage:

1. Initialization: $n := 0$, $i := 1$, $j := 1$, $D = 0$, $Q = 0$. System buffer is empty.
2. Recording events $\{A_k\}$ and $\{S_k\}$ as time proceeds. When one BP's events are recorded, $j := j + 1$;
3. For BP_i, reconstruct the partial sample path. If $m(i) = j$, that is, the synchronization occurs at BP_j, then:
   - (a) Compute $H_{n_i}(U_i, P_i)$ using Eq. 26.
   - (b) Calculate the perturbed delay D'_j, $j = I_i \le j < I'_{l(i)}$.
   - (c) Calculate the perturbed preamble $P'_{i,k}$, $k = 1, \ldots, l(i)$.
   - (d) Accumulate: $D := D + \frac{1}{N} H_{n_i}(U_i, P_i)\left\{\sum_{I_i \le j < I'_{l(i)}}\left(D_j(W, \omega) - D'_j(W, \omega)\right)\right\}$
   - (e) Accumulate: $Q := Q + \frac{1}{N}\left\{\left[\sum_{k=i}^{m-1} P_i - \sum_{k=1}^{l(i)-1} P'_{i,k}(W, \omega)\right] H_{n_i}(U_i, P_i) - 1\right\}$
   - (f) $i := i + 1$ and remove BP_i from the buffer (delete the corresponding events from memory).
4. If N messages are observed, output $\left[\frac{dJ_Q(W)}{dW}\right]^-_{SPA} = Q$, $\left[\frac{dJ_D(W)}{dW}\right]^-_{SPA} = D$.

In this serialized algorithm, instead of recording the whole sample path in memory, accumulator variables D and Q are used, allowing the algorithm to "forget" the BPs that have been processed. Hence only data from a few BPs are kept in the system memory instead of the whole N-message sample path.

In practice, both sender and receiver partially observe the sample path on their own. However, whichever party carries out the computation of the SPA derivative estimates needs complete sample path information. We assume that each message in the sender's buffer is time-stamped by the sender. After the transmission of a batch, the receiver obtains all information needed to compute the estimate. The result can be transmitted back to sender, i.e. piggy-backed in hand-shake during the next transmission. How the sample path information is exchanged between sender and receiver does not affect the computation and the result of the SPA estimates.

4.3 Optimization using stochastic approximation algorithm

Since the combined performance index is Eq. 12, obviously, an unbiased SPA estimator of $dJ(W)/dW^-$ is given by

$$\left[\frac{dJ\left(W\right)}{dW^-}\right]_{SPA} = \left[\frac{dJ_D\left(W\right)}{dW^-}\right]_{SPA} + \alpha\left[\frac{dJ_Q\left(W\right)}{dW^-}\right]_{SPA} \tag{53}$$

Assuming $J(W)$ is a differentiable function with respect to W, we have the derivative of $J\left(W\right)$ equal its left derivative:

$$\frac{dJ\left(W\right)}{dW} = \frac{dJ\left(W\right)}{dW^-}$$

so Eq. 53 is also an unbiased estimator of $dJ\left(W\right)/dW$. Using the gradient estimator (53), we use a Stochastic Approximation (SA) algorithm (Kushner and George Yin 2003) of the form

$$W_{k+1} = \Pi_{[a,b]}\left[W_k - \frac{\beta}{k^\delta}\left[\frac{dJ\left(W_k\right)}{dW_k^-}\right]_{SPA}\right], \quad k \geq 0 \tag{54}$$

with W_0 being an initial point and $\Pi_{[a,b]}\left[x\right]$ is a projection of x onto interval $\left[a,b\right]$. The parameters β and δ in Eq. 54 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 $\left[dJ\left(W_k\right)/dW_k^-\right]_{SPA}$ through the SPA algorithm, and use Eq. 54 to obtain W_{k+1}. By the fact that Eq. 53 is unbiased, W_k converges to W^* where $dJ\left(W^*\right)/dW^* = 0$.

Theorem 4 *Assume that there exists a unique $W^* \in \left[a,b\right]$ where $dJ\left(W^*\right)/dW^* = 0$. Then, W_k converges to W^*.*

Proof The convergence of the Stochastic Approximation algorithm has been studied extensively in Kushner and George Yin (2003). In our case, we invoke a simpler convergence theorem which is proven in Gokbayrak and Cassandras (2001). The theorem (Theorem 6.1 in Gokbayrak and Cassandras 2001) states that, if all six convergence criteria, denoted by H1, H2, A1, A2, E1, E2 are satisfied, the Stochastic Approximation algorithm converges. These criteria are:

H1 There exists a unique optimal in $[a, b]$.
H2 Define $h(W) = dJ(W)/dW$ which is the true derivative. $\sup_{[a,b]} \|h(W)\| < \infty$
A1 Define $\eta_k = \beta/k^\delta$, $\sum_{n=1}^{\infty} \eta_k = \infty$ a.s.
A2 $\sum_{n=1}^{\infty} \eta_k^2 < \infty$ a.s.
E1 Define $\varepsilon_k = \left[dJ(W_k)/dW_k^-\right]_{SPA} - h(W_k)$ which is the estimation error. $\sum_{n=1}^{\infty} \eta_k \left\| E\left[\varepsilon_{k+1} | \Im_k\right]\right\| < \infty$ where $\{\Im_k\}$ is a filtration generated by the sample path.
E2 $E\left[\varepsilon_{k+1}^2 | \Im_k\right] < \sigma_k^2$ where $\{\sigma_k\}$ is a sequence adapted to $\{\Im_k\}$ such that $\sum_{n=1}^{\infty} \eta_k^2 \sigma_k^2 < \infty$ a.s.

The assumption of the existence of a stationary point is H1 and thus is already established. Using notations $h(W)$ and ε_k, Eq. 54 is thus rewritten as:

$$W_{k+1} = \Pi_{[a,b]}\left[W_k - \frac{\beta}{k^\delta}\left(h(W_k) + \varepsilon_k\right)\right], \quad k \geq 0$$

Based on the proof of Theorem 2 and 3, we know that $h(W_k)$ is bounded (H2). For simplicity, we also define $\eta_k = \beta/k^\delta$. Since $0.5 < \delta \leq 1$, we have $\sum_{k=1}^{\infty} \eta_k = \infty$ (A1) and $\sum_{k=1}^{\infty} \eta_k^2 < \infty$ (A2). Further, since our SPA estimator is unbiased, $E[\varepsilon_k] = 0$ (E1). The error variance $E\left[\varepsilon_k^2\right]$ is also bounded so that $\sum_{k=1}^{\infty} \eta_k^2 E\left[\varepsilon_k^2\right] < \infty$ (E2). Define a Lyapunov function $U(W) = \|W - W^*\|^2$, with all criteria H1, H2, A1, A2, E1, E2 all satisfied. Then, Theorem 6.1 in Gokbayrak and Cassandras (2001) states that $W_k \to W^*$ w. p. 1. □

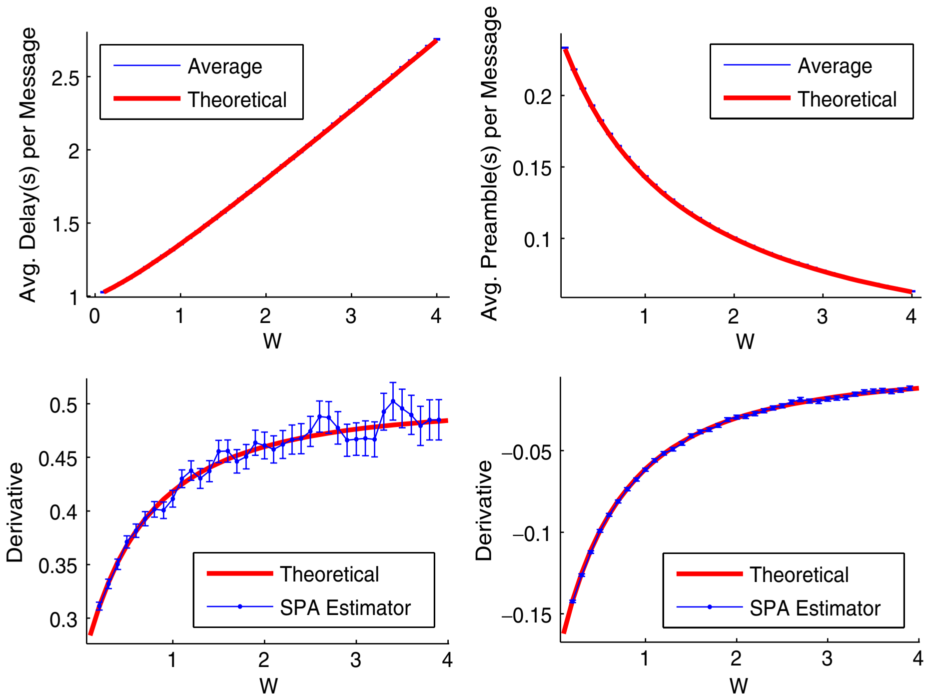
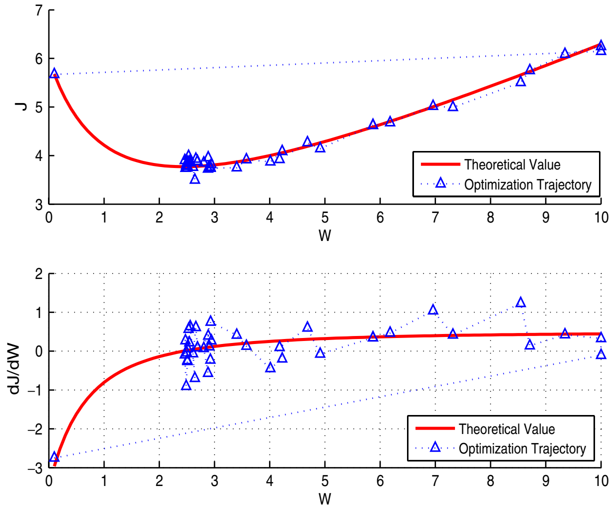


Fig. 8 SPA results with $\lambda = 3$, $\mu = 1$, $0 \leq W \leq 4$

Fig. 9 Optimization using Stochastic Approximation Algorithms. In the trajectory, not all iteration points are shown. The first 10 iteration points are shown. After that, points are shown every 10 iterations



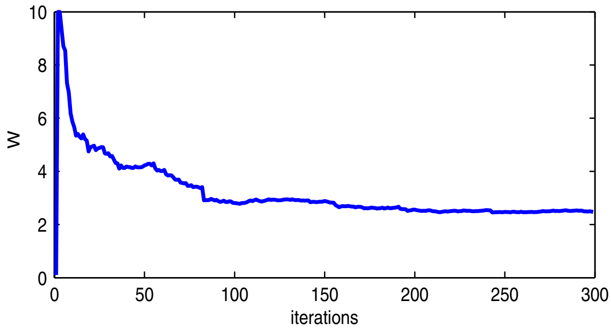
5 Numerical results

5.1 Poisson arrival and sampling 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 3. Figure 8 compares the performance metrics estimated with the analytical expressions (7), (11) and the SPA estimates obtained through Eqs. 43 and 52 with the derivatives obtained in Eq. 13, as well as with brute-force finite difference estimates, including the 95% confidence intervals. It is worth pointing out that the average delay derivative has a large variance but closely matches the theoretical value.

Figure 9 shows a typical trajectory of the optimization process using the SA algorithm. Here, we use the objective function (12) with $\alpha = 20$ and choose $\beta = 6$ and $\delta = 1$ as the parameter values in Eq. 54. We performed 300 iterations and we note that W is already in the vicinity of the optimal point after approximately 100 iterations, as shown in Fig. 10.

Fig. 10 Convergence of W



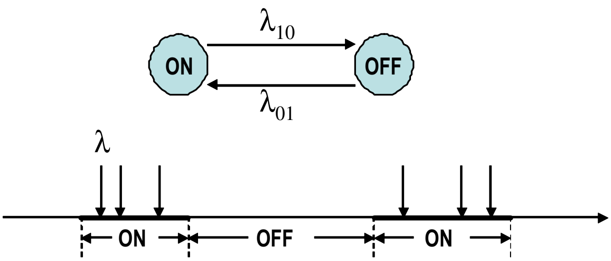


Fig. 11 A Markov Modulated Poisson Process with two states: ON/OFF

5.2 Markov modulated poisson arrival process and uniformly distributed sampling 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.

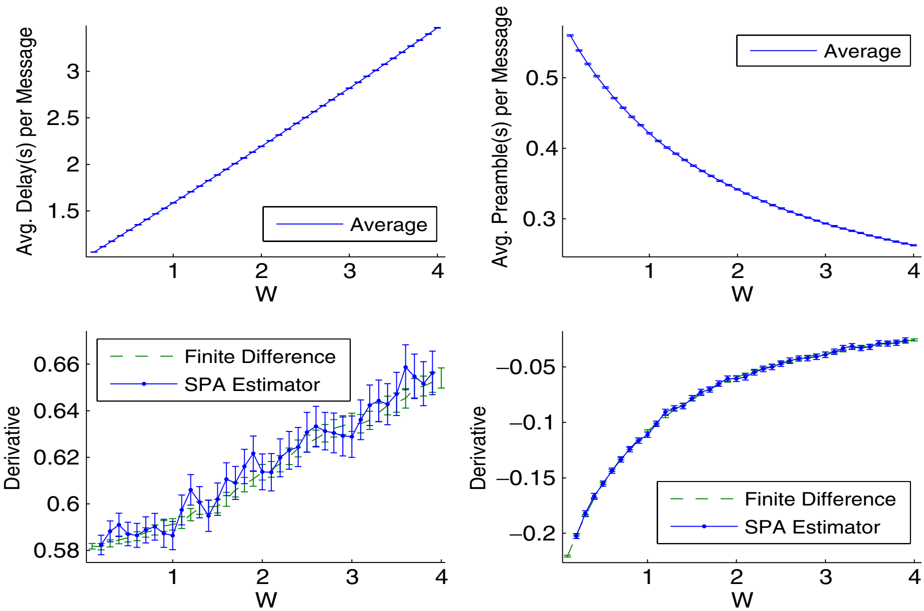


Fig. 12 SPA results with MMPP arrival process and Gamma sampling process

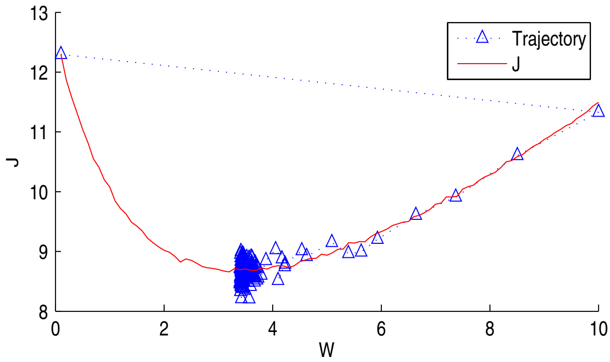


Fig. 13 On-line optimization with arrival process being a Markov Modulated Poisson Process and the sampling process having a Gamma distribution for inter-sampling time. The *red curve* of $J(W)$ is generated by exhaustive Monte-Carlo simulation

Figure 11 is an illustration of the on-off MMPP. Denote the transition rate of the Markov chain by λ_{10} and λ_{11} as shown in Fig. 11, and by λ the arrival rate during ON period. The steady state distribution of the Markov chain is $\pi_{ON} = \lambda_{01}\,(\lambda_{10} + \lambda_{01})^{-1}$ and $\pi_{OFF} = \lambda_{10}\,(\lambda_{10} + \lambda_{01})^{-1}$. It is then straightforward to generate a sample path of this process. In this example we set $\lambda = 1$, $\lambda_{10} = 0.2$, $\lambda_{01} = 0.022$ so the ON-OFF duty cycle is 10% and arrival rate during the ON period is 1. We also set that inter-polling time distribution of the receiver polling epochs is a Gamma distribution with parameter $(5, 2)$.

Figures 12 and 13 show the derivative estimates obtained and a sample trajectory of the optimization process Eq. 54. Unfortunately, under MMPP arrivals and Gamma distribution for the sampling process, there are no analytical results for comparison purposes. Therefore, we used Monte Carlo simulation to approximate the response curve $J(W)$, which is the red curve in Fig. 13. We can still see that the SA algorithm readily converges to the optimal W value in Fig. 13.

6 Conclusions

In this paper, we have proposed message batching approaches for reducing the transmission energy consumption in WSNs. Under Markovian assumptions, we derived analytical expressions for a performance objective as a function of the controllable batching time parameter and included an analysis quantifying the benefit of the batching scheme. When no analytical model is available, we have developed a gradient estimator for time-based batching using Smooth Perturbation Analysis (SPA) and proved its unbiasedness. The SPA estimators we have developed are for left derivatives. SPA estimators for right derivatives can be derived in a similar analysis, but will have a different form. 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. Our numerical results show that the analytical and simulation results are consistent, and that the SA algorithm indeed finds the optimal batching time, for both Poisson and MMPP arrival process models.

Future work is directed at extending this approach to multi-hop sensor networks. Also, finding a good hybrid policy, namely, a batching policy using both time and queue-length information is also a topic of interest.

References

Buettner M, Yee GV, Anderson E, Han R (2006) X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In: SenSys '06: proceedings of the 4th international conference on embedded networked sensor systems, New York, 1–3 November 2006, pp 307–320

Cassandras CG, Lafortune S (1999) Introduction to discrete event systems. Kluver, Norwell

Cassandras CG, Yu R (2000) A 'surrogate problem' approach for lot size optimization in manufacturing systems. In: Proceedings of the American control conference, Chicago, pp 3279–3283

Deb RK (1984) Optimal control of bulk queues with compound poisson arrivals and batch service. Opsearch 21(4):227–245

Deb RK, Serfozo RF (1973) Optimal control of batch service queues. Advanced Applied Probability 5:340–361

El-Hoiydi A, Decotignie J-D (2004) WiseMAC: an ultra low power mac protocol for multi-hop wireless sensor networks. In: Lecture notes in computer science, vol 3121. Springer, New York, pp 18–31

Fu M, Hu J-Q (1997) Conditional Monte Carlo: gradient estimation and optimization applications. Kluver, Norwell

Gaivoronski AA, Messina E (2006) Stochastic optimization algorithms for regenerative DEDS. In: Lecture notes in control and information sciences, vol 197. Springer, Berlin, pp 320–331

Gokbayrak K, Cassandras CG (2001) Online surrogate problem methodology for stochastic discrete resource allocation problems. J Optim Theory Appl 108(2):349–376

Gong W, Ho Y-C (1987) Smoothed (conditional) perturbation analysis of discrete event dynamical systems. IEEE Trans Automat Contr 32(10):858–866

Joe I, Ryu H (2007) A patterned preamble MAC protocal for wireless sensor networks. In: Proceedings of 16th international conference on computer communications and networks, ICCCN 2007, Honolulu, 13–16 August 2007, pp 1285–1290

Kushner HJ, George Yin G (2003) Stochastic approximation and recursive algorithms and applications, 2nd edn. Springer, New York

Mahlknecht S, Bock M (2004) CSMA-MPS: a minimum preamble sampling mac protocol for low power wireless sensor networks. In: Proceedings of IEEE international workshop on factory communication systems, pp 73–80

Mangione-Smith B (1995) Low power communications protocols: paging and beyond. In: Proceedings of IEEE symposium on low power electronics, San Jose, 9–11 October 1995, pp 8–11

Megerian S, Potkonjak M (2003) Wireless sensor networks. Wiley Encyclopedia of Telecommunications. Wiley, New York

Ning X, Cassandras CG (2006) Dynamic sleep time control in event-driven wireless sensor networks. In: Proceedings of the 45th IEEE conference on decision and control. San Diego, 13–15 December 2006, pp 2722–2727

Ning X, Cassandras CG (2008) Message batching in wireless sensor networks—a perturbation analysis approach. Technical report, Division of Systems Engineering, Boston University, http://people.bu.edu/nx/MessageBatchingTR.pdf

Polastre J, Hill J, Culler D (2004) Versatile low power media access for wireless sensor networks. In: SenSys '04: proceedings of the 2nd international conference on embedded networked sensor systems. ACM, New York, pp 95–107

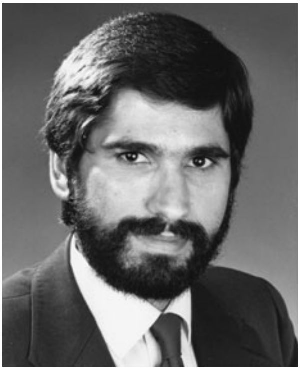
Ross SM (1995) Stochastic processes. Wiley, New York

Shnayder V, Hempstead M, Chen B, Allen GW, Welsh M (2004) Simulating the power consumption of large-scale sensor network applications. In: SenSys '04: proceedings of the 2nd international conference on embedded networked sensor systems. ACM, New York, pp 188–200

Sohrabi K, Gao J, Ailawadhi V, Pottie GJ (2000) Protocols for self-organization of a wireless sensor network. IEEE Pers Commun 7(5):16–27

Ye W, Heidemann J, Estrin D (2004) Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Trans Netw 12(3):493–506

Xu Ning was born in Guangzhou, Guangdong, China. He received the B.Eng. degree from Tsinghua University, Beijing, China in 2004, M.S. and Ph.D. degree from Boston University, Boston, Massachusetts, US in 2008 and 2009, respectively. He specializes in the areas of stochastic control, optimization and statistical learning, sensor network and robotics. He has briefly worked at ExxonMobil Corporate Strategic Research, Annandale, New Jersey, US, and Siemens Corporate Research, Princeton, New Jersey, US, in 2006 and 2008, respectively. Currently he is with Microsoft Corporation, focusing on the Internet search technology. He is a member of the IEEE and the INFORMS.



Christos G. Cassandras received the B.S. degree from Yale University, New Haven, CT, the M.S.E.E degree from Stanford University, Stanford, CA, and the S.M. and Ph.D. degrees from Harvard University, Cambridge, MA, in 1977, 1978, 1979, and 1982, respectively. From 1982 to 1984 he was with ITP Boston, Inc. where he worked on the design of automated manufacturing systems. From 1984 to 1996 he was a Faculty Member at the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst. Currently, he is Head of the Division of Systems Engineering and Professor of Electrical and Computer Engineering at Boston University, Boston, MA and a founding member of the Center for Information and Systems Engineering (CISE). He specializes in the areas of discrete event and hybrid systems, stochastic optimization, and computer simulation, with applications to computer networks, sensor networks, manufacturing systems, transportation systems, and command/control systems. He has published over 250 papers in these areas, and four books. Dr. Cassandras is currently Editor-in-Chief of the IEEE Transactions on Automatic Control and has served on several editorial boards and as Guest Editor for various journals. He has also served on the IEEE Control Systems Society Board of Governors. He is the recipient of several awards, including the Distinguished Member Award of the IEEE Control Systems Society (2006), the 1999 Harold Chestnut Prize, and a 1991 Lilly Fellowship. He is a member of Phi Beta Kappa and Tau Beta Pi, a Fellow of the IEEE and a Fellow of the IFAC.