Optimal Stochastic Policies for Distributed Data Aggregation in Wireless Sensor Networks

Zhenzhen Ye, Alhussein A. Abouzeid and Jing Ai

***Abstract*—The scenario of *distributed data aggregation* in wireless sensor networks is considered, where sensors can obtain and estimate the information of the whole sensing field through local data exchange and aggregation. An intrinsic trade-off between energy and aggregation delay is identified, where nodes must decide optimal instants for forwarding samples. The samples could be from a node's own sensor readings or an aggregation with samples forwarded from neighboring nodes. By considering the randomness of the sample arrival instants and the uncertainty of the availability of the multi-access communication channel, a *sequential decision process* model is proposed to analyze this problem and determine optimal decision policies with local information. It is shown that, once the statistics of the sample arrival and the availability of the channel satisfy certain conditions, there exist optimal *control-limit* type policies which are easy to implement in practice. In the case that the required conditions are not satisfied, the performance loss of using the proposed control-limit type policies is characterized. In general cases, a finite-state approximation is proposed and two on-line algorithms are provided to solve it. Practical distributed data aggregation simulations demonstrate the effectiveness of the developed policies, which also achieve a desired energy-delay tradeoff.**

***Index Terms*—Data aggregation, energy-delay tradeoff, semi-Markov decision processes, wireless sensor networks.**

I. Introduction

Data aggregation is recognized as one of the basic distributed data processing procedures in sensor networks for saving energy and reducing contentions for communication bandwidth. We consider the scenario of *distributed data aggregation* where sensors can obtain and estimate the information of the whole sensing field through data exchange and aggregation with their neighboring nodes. Such fully decentralized aggregation schemes eliminate the need for fixed tree structures and the role of sink nodes, i.e., each node can obtain global estimates of the measure of interest via local information exchange and propagation, and an end-user can enquire an arbitrary node to obtain the information of the whole sensing field. Because of its robustness and flexibility in face of network uncertainties, such as topology change and nodes failure, it stimulated a lot of research interests recently, e.g., [1], [2], [3], [4]. In [1], the authors present the motivation and a good example of distributed, periodic data aggregation.

The local information exchange in distributed data aggregation generally is asynchronous and thus the arrival of samples at a node is random. For energy saving purpose, a node prefers to aggregate as much as possible information before sending out a sample with the aggregated information. The aggregation operation is also helpful in reducing the contention for communication resources. However, delay due to waiting for aggregation should also be taken into account as it is directly related to the accuracy of the information represented by certain temporal distortion [5]. This is especially true for certain time-sensitive applications in large-scale wireless sensor networks, such as environment monitoring, disaster relief and target tracking. Therefore, *a fundamental trade-off exists between energy and delay in aggregation, which imposes a decision-making problem in aggregation operations*. A node should decide when is the optimal time instant for sending out the aggregated information, given any available local knowledge of the randomness of sample arrival as well as the channel contention. In general, the exact optimal time instants might not be easy to find. However, since computation is much cheaper than communication [6], [7], exploiting the on-board computation capabilities of sensor nodes to discover near-optimal time instants is worthwhile.

In this paper, we propose a semi-Markov decision process (SMDP) model to analyze the decision problem and determine the optimal policies at nodes with local information. The decision problem is formulated as an *optimal stopping* problem with an infinite decision horizon and the expected total discounted reward optimality criterion is used to take into account the effect of delay. In the proposed formulation, instead of directly characterizing the complicated interaction between energy consumption and delay in the aggregation (i.e., how much delay can tradeoff how much energy), the proposed reward structure (see Section II-A) addresses a much more natural objective in data aggregation - *a lower energy consumption and a lower delay are better*. With this objective, the intrinsic energy-delay tradeoff achieves one of its equilibria when the maximal reward is obtained. With this formulation, we show that[1], once the statistics of sample arrival and the availability of the multi-access channel approximately satisfy certain conditions as described in Section IV, there exists simple *control-limit* type policies which are optimal and easy to implement in practice. In the case that the required conditions are not satisfied, the control-limit policies are low-complexity alternatives to the optimal policy and the performance loss can be bounded. We also propose a finite-state approximation of the original decision problem to provide near-optimal policies which do not require any assumption on the random processes of sample arrival and channel availability. For implementation, we provide two on-line algorithms, adaptive real-time dynamic

This work was supported in part by National Science Foundation (NSF) grant 0546402. Some preliminary results of this work is reported in the proceedings of INFOCOM 2007. Z. Ye, A. A. Abouzeid and J. Ai are with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute Troy, NY 12180-3590, USA; Email: yez2@rpi.edu, abouzeid@ecse.rpi.edu, aij@rpi.edu.

[1]Except for most major theorems, we skip the technical proofs of the results due to the space limit, and refer the interested readers to [8].

programming (ARTDP) and real-time Q-learning (RTQ), to solve the finite-state approximation. These algorithms are practically useful on current wireless sensors, e.g., the Crossbow motes [9]. The numerical properties of the proposed policies are investigated with a tunable traffic model. The simulation on a practical distributed data aggregation scenario demonstrates the effectiveness of the policies we developed, which can achieve a good energy-delay balance, compared to previous fixed degree of aggregation (FIX) scheme and on-demand (OD) aggregation scheme [10].

To the best of our knowledge, the problem of "to send or wait" described earlier, has not been formally addressed as a stochastic decision problem. Related work is also limited. Most of the research related to timing control in aggregation, i.e., how long should a node wait for samples from its children or neighbors before sending out an aggregated sample, focuses on tree based aggregation, such as directed diffusion [11], TAG [12], SPIN [13] and Cascading timeout [14]. In these schemes, each node has preset a specific and bounded period of time that it should wait. The transmission schedule at a node is fixed once the aggregation tree is constructed and there is no dynamic adjustment in response to the degree of aggregation (DOA), i.e., the number of samples collected in one aggregation operation, or the quality of aggregated information. One exception is [15], in which the authors have proposed a simple centralized feedback timing control for tree-based aggregation. In their scheme, the maximum duration for one data aggregation operation is preset by the sink and propagated to each node within the aggregation tree; then each node can calculate its waiting time for aggregation and execute the aggregation operation; when the data is collected by the sink, the sink will evaluate the quality of aggregation and adjust the maximum duration for aggregation for the next cycle. Distributed control for DOA is introduced in [10]. The target of the control loop proposed in their scheme is to maximize the utilization of the communication channel, or equivalently, minimize the MAC layer delay, as they mainly focus on real-time applications in sensor networks. Energy saving is only an ancillary benefit in their scheme. Our concern is more general than that in [10] as the objective here is to achieve a desired energy-delay balance. Minimizing MAC delay is only one extreme performance point that can be reduced from the general formulation proposed in this paper.

As one of the most important models for stochastic sequential decision problems, the Markov decision process (MDP) and its generalization SMDP have been applied to solve various engineering problems in practice (see numerous examples in [16]). In network research literature, MDP and SMDP models are well-known for solving problems such as admission control, buffer management, flow and congestion control, routing, scheduling/polling of queues as well as the Internet web search (e.g. see [17] and the references therein). The optimal stopping problems are an important subset of stochastic sequential decision problems, with important applications in areas of statistics, economics and mathematical finance [18], [19]. Among various existing optimal stopping problems, our work has some of the flavor of the fishing problem [20], [21], the proofreading and debugging problem [18] as well as the aggregation problem in web search [22]. The existing results for these problems, however, can not be directly applied to the aggregation problem considered in this paper due to some commonly used but unrealistic assumptions in these prior works. For example, the total number of random events, such as the number of fish in a lake, the number of bugs in a manuscript or the number of information sources in web search, is usually assumed to be either deterministic and known or random but its distribution is known. And the random events are usually following an (known) independent and identical distribution (i. i. d.). We relax these assumptions in analyzing the data aggregation problem. Moreover, with the introduction of the SMDP model and the learning approaches, the solution provided in this paper is more practically useful in the sense that it can be applied to the data aggregation problem in continuous-time domain, with an unknown probability model.

II. Problem Formulation

A. A Semi-Markov Decision Process Model

During a data aggregation operation, from a node's localized point of view, the arrivals of samples, either from neighboring nodes or local sensing, are random and the arrival instants can be viewed as a random sequence of points along time, i.e., a point process. We define the associated counting process as the *natural process*. As an aggregation operation begins at the instant of the first sample arrival, the *state of the node* at a particular instant, i.e., the number of collected samples by that instant, lies in a state space $S' = \{1, 2, ...\}$. On the other hand, for a given node, the availability of the multi-access channel for transmission can also be regarded as random. This can be justified by the popularity of random access MAC protocols in wireless sensor networks (e.g. [23]). Only when the channel is sensed to be free, the sample with aggregated information could be sent. Thus, at each available transmission epoch, the node decides to either (a) "send", i.e., stop current aggregation operation and send the aggregated sample or (b) "wait" and thus give up the opportunity of transmission and continue to wait for a larger degree of aggregation (DOA). These available transmission epochs can also be called *decision epochs/stages*. The distribution of the inter-arrival time of the decision epochs could be arbitrary, depending, for example, on the specific MAC protocol. The sequential decision problem imposed on a node is thus to choose a suitable *action* (to continue to wait for more aggregation, or stop immediately) at each decision epoch, based on the history of observations up to the current decision epoch. A *decision horizon* starts at the beginning of an aggregation operation. When the decision for stopping is made, the sample with aggregated information is sent out and the node enters an (artificial) absorbing state and stays in this absorbing state until the beginning of the next decision horizon. See Fig. 1 for a schematic diagram illustrating these operations.

To model the decision process on an individual node, we assume that, at an available transmission epoch with s_n collected samples on the node, the time interval to the next available transmission epoch (i.e., the instant that the channel is idle again) and the number of samples that will arrive on the node in this interval only depend on the number of samples

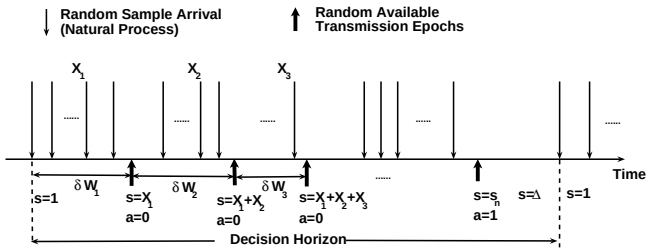

Fig. 1. A schematic illustration of the decision process model for data aggregation at a node. The decisions are made at available transmission epochs; with the observation of the current node's state s, i.e., the number of samples collected, and the elapsed time $\sum_i \delta W_i$, action a is selected (0: continuing for more aggregation; 1: stopping current aggregation). After the action for stopping, the node enters the absorbing state Δ till the beginning of the next decision horizon.

already collected, s_n, irrelevant to when and how these s_n samples were collected. We state this condition formally in the following assumption. The effectiveness of this condition will be justified by the performance of decision policies based on it in Section V-B.

Assumption 2.1: Given the state $s_n \in S'$ at the nth decision epoch, if the decision is to continue to wait, the random time interval δW_{n+1} to the next decision epoch and the random increment X_{n+1} of the node's state are independent of the history of state transitions and the nth transition instant t_n.

With Assumption 2.1 and the observation that the distribution of the inter-arrival time of the decision epochs might be arbitrary, the decision problem can be formulated with a semi-Markov decision process (SMDP) model. The proposed SMDP model is determined by a 4-tuple $\{S, A, \{Q_{ij}^a(\tau)\}, R\}$, which are the state space S, action set A, a set of action-dependent state transition distributions $\{Q_{ij}^a(\tau)\}$ and a set of state- and action-dependent instant rewards R. Specifically,

- $S = S' \cup \{\Delta\}$, where Δ is the absorbing state;
- $A = \{0, 1\}$, with $A_s = \{0, 1\}, \forall s \in S'$ and $A_s = \{0\}$ for $s = \Delta$, where $a = 0$ represents the action of continuing for aggregation and $a = 1$ represents stopping the current aggregation operation;
- $Q_{ij}^a(\tau) \triangleq \mathbb{P}_r\{\delta W_{n+1} \leq \tau, s_{n+1} = j | s_n = i, a\}$, $i, j \in S, a \in A_i$ is the transition distribution from state i to j given the action at state i is a; $Q_{i\Delta}^1(\tau) = u(\tau)$ for $i \in S'$ and $Q_{\Delta\Delta}^0(\tau) = u(\tau)$, where $u(\tau)$ is the step function;
- $R = \{r(s, a)\}$, where

$$r(s,a) = \begin{cases} g(s), & a = 1,\ s \in S' \\ 0, & otherwise \end{cases}$$

with $g(s)$ as the aggregation gain achieved by aggregating s samples when stopping, which is nonnegative and nondecreasing with respect to (w.r.t.) s.

The specific form of $g(s)$ depends on the application. Specifically, the energy saving in two classes of aggregation problems can be appropriately characterized by the aggregation gain $g(s)$ in this formulation.

1) The aggregation problems using application-independent data aggregation (AIDA) scheme [10]. In AIDA, the collected samples in an aggregation operation are concatenated to form a new aggregated sample. The energy saving of this operation comes from the reduction in MAC control overhead and is a simple function of DOA, i.e., the state s. Thus, the aggregation gain $g(s)$ defined above may be used to represent the actual energy saving in AIDA.
2) The aggregation problems using application-dependent data aggregation (ADDA) with the interested quantity summary in which the actual energy saving has a simple relation to the number of samples aggregated. The examples of such quantity summary include the maximum/minmum, average, count and range of the interested quantity. In these examples, the actual energy saving is approximately proportional to the number of aggregated samples and thus can also be modeled by the function $g(s)$ defined above.

One should also note that the actual energy gain using ADDA might be complicated in some cases, not purely determined by the number of collected samples. One of such examples is the aggregation operation performed with lossless compression algorithms. In this case, the energy saving depends on the correlation structure of the collected samples and in general, this correlation structure can not be simply determined by the number of samples, but closely related to other properties of the samples, such as the locations and/or the instants of generation of these samples. To apply the proposed framework to these aggregation problems, we can redefine the *state* of a node in the decision process model by *including the factors that affect actual energy saving in the aggregation*, though this redefinition of the state space in the decision process model would change the size of the state space and thus might raise some computational issues in implementation. For example, assume the energy saving is determined by the physical locations of collected samples and all possible sample locations are in a finite set Y. By redefining the state space S' in our decision process model as the set of all subsets of Y except the null set, there exists certain function $g(s), s \in S'$ to represent the actual energy gain.

With this SMDP model, the objective of the decision problem becomes: find a *policy* π^* composed of *decision rules* $\{\mathbf{d}_n\}, n = 1, 2, ...$, to maximize the expected reward of aggregation, where the decision rule $\mathbf{d}_n, n = 1, 2, ...$, specifies the actions on all possible states at the nth decision epoch. As our target is to achieve a desired energy-delay balance, the reward of aggregation should relate to the state of the node when stopping (which in turn determines the aggregation gain $g(s)$) and the experienced aggregation delay. To incorporate the impact of aggregation delay in decisions, we adopt the expected total discounted reward optimality criterion with a discount factor $\alpha > 0$ [16]. That is, for a given policy $\pi = \{\mathbf{d}_1, \mathbf{d}_2, ...\}$ and an initial state s, the expected reward is defined as

$$v^\pi(s) = \mathbb{E}_s^\pi \left[\sum_{n=0}^{\infty} e^{-\alpha t_n} r(s_n, \mathbf{d}_{n+1}(s_n)) \right] \quad (1)$$

where $s_0 = s,\ t_0 = 0$ and $t_0,\ t_1, ...$ represent the instants of successive decision epochs. The motivations for the choice of an exponential discount of the reward w.r.t. delay in (1) are as follows. First, exponential discount is monotone and thus satisfies the intuition on the monotonic decrease

of the reward w.r.t. the increase of delay. Second, from an application perspective, an exponential discount function on delay can be a good indicator on the information accuracy. For example, in a commonly used Gauss-Markov field model for spatial-temporal correlated dynamic phenomena, the accuracy of information decreases exponentially with the delay [5]. Third, the proposed multiplicative reward structure and the exponential discount function can also handle the additive discount (w.r.t. delay) cases. For example, the commonly used reward $r(s) - \alpha t$ in optimal stopping/MDP literature can be easily translated into the proposed reward structure with $g(s) \triangleq e^{r(s)}$. The monotonicity of the exponential function guarantees that two reward structures have the same maximizer. Finally, the exponential discount w.r.t. delay in the reward structure is helpful in developing practically useful control-limit policies (in Section III) and learning algorithms (in Section IV) as it perfectly fits the SMDP model and thus simplifies mathematical manipulations. On the other hand, we admit that there are other choices for selecting the delay discount function in the decision process model [22]. The basic idea of the proposed decision framework can still be applied, though the analytical results might be slightly different under these different reward settings.

By defining

$$v^*(s) = \sup_{\pi} v^{\pi}(s) \tag{2}$$

as the optimal expected reward with initial state $s \in S$, we are trying to find a policy π^* for which $v^{\pi^*}(s) = v^*(s)$ for all $s \in S$. It is clear that $v^*(s) \geq 0$ for all $s \in S$ as $r(s,a) \geq 0$ for all $s \in S$ and $a \in A_s$. We are especially interested in $v^*(1)$ since an aggregation operation always begins at the instant of the first sample arrival[2]. Furthermore, in an aggregation operation, by stopping at the nth decision epoch with state $s_n \in S'$ and total elapsed time t_n, the reward obtained at the stopping instant is given by

$$Y_n(s_n, t_n) = g(s_n)e^{-\alpha t_n} \tag{3}$$

where the achieved aggregation gain $g(s_n)$ is discounted by the delay experienced in aggregation. To ensure there exists an optimal policy for the problem, we impose the following assumption on the reward at the stopping instant [18].

Assumption 2.2: (1) $\mathbb{E}[\sup_n Y_n(s_n, t_n)] < \infty$; and (2) $\lim_{n\to\infty} Y_n(s_n, t_n) = Y_\infty = 0$ with probability 1.

This assumption is reasonable under almost all practical scenarios. Condition (1) implies that, for any possible initial state $s \in S'$, the expected reward under any policy is finite [18]. This is realistic as the number of samples expected to be collected within any finite time duration is finite. For any practically meaningful setting of the aggregation gain, its expected (delay) discounted value should be finite. In condition (2), $Y_\infty = 0$ represents the reward of an endless aggregation operation. In practice, with the elapse of time (as $n \to \infty$, $t_n \to \infty$), the reward should go to zero since aggregation with indefinite delay is useless.

[2]Note that the first actual available transmission epoch within a decision horizon is not necessary to be the instant that $s = 1$ (as shown in Fig. 1).

B. The Optimality Equations and Solutions

Under Assumption 2.2, obtaining the optimal reward $\mathbf{v}^* = [v^*(\Delta)\ v^*(1)\ \ ...\ ]^T$ and corresponding optimal policy can be achieved by solving the following optimality equations

$$\begin{aligned} v(s) &= \max\{g(s) + v(\Delta), \mathbb{E}[v(j)e^{-\alpha\tau}|s]\} \\ &= \max\{g(s) + v(\Delta), \sum_{j\geq s} q^0_{sj}(\alpha)v(j)\} \end{aligned} \tag{4}$$

$\forall s \in S'$ and[3] $v(\Delta) = v(\Delta)$ for $s = \Delta$, where the first term in the maximization, i.e., $g(s) + v(\Delta)$, is the reward obtained by stopping at state s, and the second term, $\mathbb{E}[v(j)e^{-\alpha\tau}|s]$, represents the expected reward if continuing to wait at state s. In (4), $q^a_{sj}(\alpha) \triangleq \int_0^\infty e^{-\alpha\tau} dQ^a_{sj}(\tau)$, $a \in A_s$, is the Laplace-Stieltjes transform of $Q^a_{sj}(\tau)$ with the parameter $\alpha(> 0)$. And it is straightforward to see that $\sum_{j\geq s} q^0_{sj}(\alpha) < 1$.

Note that the solution of the above optimality equations is not unique. Following similar procedures to the proofs of Theorem 7.1.3, 7.2.2 and 7.2.3 in [16] by substituting the transition probability matrix $\mathbf{P}_d$ in the theorems with Laplace-Stieltjes transform matrix $\mathbf{M}_d \triangleq [q^a_{ij}(\alpha)], d(i) = a, i, j \in S$, in our problem, we have

Result 1: optimal reward $\mathbf{v}^* \geq 0$ is the minimal solution of the optimality equations (4) and consequently, $v^*(\Delta) = 0$.

Furthermore, by applying Theorem 3 (Chapter 3) in [18] on the SMDP model, we obtain

Result 2: there exists an optimal stationary policy $\mathbf{d}^\infty = \{\mathbf{d}, \mathbf{d}, ...\}$ where the optimal decision rule $\mathbf{d}$ is

$$d(s) = \arg\max_{a\in A_s} \{ag(s) + (1-a)\sum_{j\geq s} q^0_{sj}(\alpha)v^*(j)\} \tag{5}$$

$\forall s \in S'$ and $d(\Delta) = 0$.

Although (5) gives a general optimal decision rule and the corresponding stationary policy, it relies on the evaluation of the optimal reward $\mathbf{v}^*$. In the given countable state space S', we have not yet provided a way to solve or approximate the value of $\mathbf{v}^*$. To obtain an optimal (or near-optimal) policy, we will investigate two questions:

1) Is there any structured optimal policy which can be obtained without solving $\mathbf{v}^*$ and is attractive in implementation? What are the conditions for the optimality of such a policy? And how much we lose in the value of reward by using such policies when the optimality conditions are not satisfied?
2) Without structured policies, can we approximate the value of $\mathbf{v}^*$ with a truncated (finite) state space, and is there any efficient algorithm to obtain the solution for such finite-state approximation?

The answers to the questions will be presented in the following two sections, respectively.

III. Control-limit Policies

In this section, we will discuss the structured solution of the optimal policy in (5). Such a solution is attractive for implementation in energy and/or computation capability limited sensor networks as it significantly reduces the search effort for the optimal policy in the state-action space once we know there

[3]The equation states that the value of the absorbing state Δ is a free variable in the optimality equations and thus mathematically, there are infinite number of solutions $\mathbf{v}(\geq 0)$ to satisfy the optimality equations.

exists an optimal policy with certain special structure. We are especially interested in a *control-limit* type policy as its action is monotone in state $s \in S'$, i.e., $\pi = \mathbf{d}^\infty = \{\mathbf{d}, \mathbf{d}, ...\}$ with the decision rule $\mathbf{d}$

$$d(s) = \begin{cases} 0, & s < s^* \\ 1, & s \geq s^* \end{cases}, \tag{6}$$

where $s^* \in S'$ is a *control limit*. Thus, the search for the optimal policy is reduced to simply finding s^*, i.e., a *threshold* on the number of samples that a node should aggregate before initiating a transmission.

A. Sufficient Conditions for Optimal Control-Limit Policies

By observing that the state evolution of the node is non-decreasing with time, i.e., the number of samples collected during one aggregation operation is nondecreasing, we provide in Theorem 3.1 a sufficient condition for the existence of an optimal control-limit policy under Assumption 2.2, which is primarily based on showing the optimality of one-stage-lookahead (1-sla) decision rule (or stopping rule [18]).

Theorem 3.1: Under Assumption 2.2, if the following inequality (7) holds for all $i \geq s,\ i, s \in S'$ once it holds for certain s,

$$g(s) \geq \sum_{j \geq s} q^0_{sj}(\alpha) g(j), \tag{7}$$

then a control-limit policy with the control limit

$$s^* = \min \{s \geq 1 : g(s) \geq \sum_{j \geq s} q^0_{sj}(\alpha) g(j)\} \tag{8}$$

is optimal and the expected reward is

$$\tilde{v}(s) = \begin{cases} \sum_{j \geq 1} H_{sj}(\alpha) g(j + s^* - 1), & s < s^* \\ g(s), & s \geq s^* \end{cases}, \tag{9}$$

where $\mathbf{H}(\alpha) \triangleq [H_{sj}(\alpha)] = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ with $\mathbf{A} \triangleq [A_{ij}] \in \mathbb{R}^{(s^*-1)\times(s^*-1)}$

$$A_{ij} = \begin{cases} q^0_{ij}(\alpha), & 1 \leq i \leq j < s^* \\ 0, & otherwise \end{cases} \tag{10}$$

and $\mathbf{B} \triangleq [B_{ij}] \in \mathbb{R}^{(s^*-1)\times\infty}$

$$B_{ij} = q^0_{ij}(\alpha), \quad 1 \leq i < s^*, j \geq s^*. \tag{11}$$

Proof: See Appendix A. ∎

In Theorem 3.1, the optimality of 1-sla decision rule tells us that, at a transmission epoch, if the node thinks that the currently obtained aggregation gain, discounted by the delay, is larger than the expected discounted aggregation gain at the next transmission epoch, it should stop the aggregation operation and send the aggregated sample at current transmission epoch. However, this sufficient condition for the optimality of the control-limit policy in Theorem 3.1 requires to check (7) for all states, which is rather difficult computationally. We would thus like to know if there exists any other condition which is more convenient for us to check for the optimality of 1-sla decision rule in practice, even if it is sufficient most but not all of the time. For this purpose, we show that if

1) the aggregation gain is concavely or linearly increasing with the number of collected samples; and,
2) with a smaller number of collected samples at the node (e.g., state i), it is more likely to receive any specific number of samples or more (e.g., $\geq m$ samples), than that with a larger number of samples already collected (e.g., state $i+1$), by the next decision epoch;

then the condition for the existence of an optimal control-limit policy in Theorem 3.1 *almost always* holds. We formally state the above conditions in the following Corollary.

Corollary 3.2: Under Assumption 2.2, suppose $g(i+1) - g(i) \geq 0$ is non-increasing with state i for all $i \in S'$ and if the following inequality (12) holds for all states $i \geq s,\ i, s \in S'$ once (7) is satisfied at certain s,

$$\sum_{j \geq k} Q^0_{ij}(\tau) \geq \sum_{j \geq k} Q^0_{i+1,j+1}(\tau),\ \forall k \geq i,\ \forall \tau \geq 0. \tag{12}$$

Then, there exists an optimal control-limit policy.

As a special case of Corollary 3.2, if the dependency of $Q^0_{ij}(\tau)$ on the current state i can be further relaxed, i.e., the (random) length of the interval between consecutive transmission epochs and the (random) number of samples arrived within this interval are independent of the number of samples already collected by the node, (12) is satisfied as $Q^0_{ij}(\tau) = Q^0_{i+1,j+1}(\tau) \triangleq Q^0_{j-i}(\tau), \forall j \geq i, i \in S', \forall \tau \geq 0$. Thus, there exists an optimal control-limit policy. Furthermore, for a linear aggregation gain $g(s) = s - 1$,

$$\begin{aligned} \textstyle\sum_{j \geq s} q^0_{sj}(\alpha) g(j) &= \textstyle\sum_{j \geq s} \int_0^\infty (j-1) e^{-\alpha\tau} dQ^0_{sj}(\tau) \\ &= \mathbb{E}[X e^{-\alpha \delta W}] + (s-1)\mathbb{E}[e^{-\alpha \delta W}], \end{aligned}$$

where δW is the random interval of consecutive available transmission epochs and X is the increment of the natural process (i.e., the number of arrived samples) in the interval. From (8), a closed-form expression for the optimal control limit s^* can be obtained as

$$s^* = \left\lceil \frac{\mathbb{E}[X e^{-\alpha \delta W}]}{1 - \mathbb{E}[e^{-\alpha \delta W}]} + 1 \right\rceil. \tag{13}$$

Eqn. (13) is practically attractive since the optimal threshold number of the samples that a node should aggregate can be obtained by directly measuring the expected "incremental reward" $\mathbb{E}[X e^{-\alpha \delta W}]$ and the expected "delayed-induced discount factor" $\mathbb{E}[e^{-\alpha \delta W}]$ during the aggregation operation.

B. An Upper-Bound on Reward Loss with the Control-limit Policy in Theorem 3.1

When the conditions in Theorem 3.1 or Corollary 3.2 are not satisfied, the control-limit policy with the control-limit in (8) is not necessary to be optimal. In this case, a natural question is how much we lose in the value of reward by using the control-limit policy with the control-limit in (8). To characterize such loss, we impose the following assumption.

Assumption 3.3: (1) $\exists \beta \in (0,1)$, such that $\sum_{j \geq s} q^0_{sj}(\alpha) \leq \beta, \forall s \in S'$; (2) $\exists L > 0$, such that $\sum_{j \geq s} q^0_{sj}(\alpha) g(j) \leq g(s) + L, \forall s \in S'$.

In this assumption, condition (1) implies that the (random) time for the state transition between two consecutive decision epochs is not identically zero, which ensures that only a finite number of state transitions in a finite period of time. This is realistic since a node always needs nonzero time to receive/process samples. For example, if there is a fixed "response" time period $\delta W_f > 0$ for the node to make

a decision or process the received sample(s), we can set $\beta = e^{-\alpha\delta W_f} < 1$. Condition (2) implies that the expected increase of aggregation gain at any state $s \in S'$ by waiting one more decision stage is bounded. Such constraint on $g(s)$ is not restrictive in practice, which can be illustrated from the following examples.

Example 3.4: When a lossless compression scheme based on the temporal-spatial correlation between samples is used, the aggregation gain is generally bounded, i.e., $\exists M > 0, g(s) \leq M < \infty, \forall s \in S'$. Thus $\sum_{j\geq s} q_{sj}^0(\alpha)g(j) \leq \beta M$. Let $L = \beta M$, condition (2) in Assumption 3.3 is satisfied.

Example 3.5: When the type of aggregation is maximum/minimum, average or count, the model of a (unbounded) linear gain $g(s) = s - 1$ might be used. When the optimality condition for (13) is satisfied and if the expected number of samples arrived between two consecutive decision epochs (i.e., X) is finite, we may set $L = \mathbb{E}[Xe^{-\alpha\delta W}]$ and for any $s \in S'$

$$\begin{aligned}\sum_{j\geq s} q_{sj}^0(\alpha)g(j) &= \sum_{j\geq s} q_{sj}^0(\alpha)(j-s) + (s-1)\sum_{j\geq s} q_{sj}^0(\alpha) \\ &\leq \mathbb{E}[Xe^{-\alpha\delta W}] + \beta(s-1) \leq L + g(s).\end{aligned}$$

We first state the following lemma which bounds the difference between the optimal reward $v^*(s)$ and the aggregation gain $g(s)$ for any $s \in S'$.

Lemma 3.6: Under Assumptions 2.2 and 3.3,

$$v^*(s) - g(s) \leq \frac{L}{1-\beta}, \forall s \in S'. \tag{14}$$

With Lemma 3.6, we can bound the reward loss of using the control-limit policy in Theorem 3.1 as follows.

Theorem 3.7: Under Assumptions 2.2 and 3.3, for the control-limit policy with the control limit

$$s^* = \min\{s \geq 1 : g(s) \geq \sum_{j\geq s} q_{sj}^0(\alpha)g(j)\},$$

the loss between the achievable reward $\tilde{v}(s)$ and the optimal reward $v^*(s)$ for any $s \in S'$ is bounded by

$$v^*(s) - \tilde{v}(s) \leq \begin{cases} \sum_{j\geq 1} H_{sj}(\alpha)\frac{L}{1-\beta}, & s < s^* \\ \frac{L}{1-\beta}, & s \geq s^* \end{cases}. \tag{15}$$

Eqn. (15) shows that the performance gap between the optimal policy and the control-limit policy proposed in Theorem 3.1 would *not* be arbitrarily large, even for an *unbounded* aggregation gain setting, as long as Assumption 3.3 is satisfied. As we have shown that Assumption 3.3 is not a restrictive assumption, Theorem 3.7 implies that the control-limit policy would be useful as a low-complexity alternative to the optimal policy in practice. In Section V-B we will show that the control-limit policy can achieve a near-optimal performance in a practical distributed data aggregation scenario.

Since the control-limit policy developed in Theorem 3.1 is based on the 1-sla decision rule [18], we can also find an upper-bound for the reward loss of using the policy with the 1-sla decision rule, which is stated in the following corollary.

Corollary 3.8: Under Assumptions 2.2 and 3.3, for the policy with the 1-sla decision rule $\mathbf{d}$ given by

$$d(s) = \arg\max_{a\in\{0,1\}} \{ag(s) + (1-a)\sum_{j\geq s} q_{sj}^0(\alpha)g(j)\},$$

for $s \in S'$ and $d(\Delta) = 0$. The loss between the achievable reward $\hat{v}(s)$ and the optimal reward $v^*(s)$ for any $s \in S'$ is bounded by

$$v^*(s) - \hat{v}(s) \leq \frac{\beta L}{1-\beta}. \tag{16}$$

C. Comparison to Aggregation Policies in the Literature

From (8) and (13), we can see some similarities and differences between the control-limit policies and the previously proposed fixed degree of aggregation (FIX) and on-demand (OD) schemes [10].

In the FIX scheme, its target is to aggregate a fixed number of samples and, once the number is achieved, the aggregated sample will be sent to the transmission queue at the MAC layer. To avoid waiting an indefinite amount of time before being sent, a time-out value is also set to ensure that aggregation is performed, regardless of the number of samples, within some time threshold. The target of (13) is also to collect at least s^* samples, but this threshold value is based on the estimation of statistical characteristics of the sample arrival and the channel availability, rather than a preset fixed value; also, different nodes might follow different values of s^*.

In OD (or opportunistic) aggregation scheme, an aggregation operation continues as long as the MAC layer is busy. Once the transmission queue in the MAC layer is empty, the aggregation operation is terminated and the aggregated sample is sent to the queue. The objective of the OD scheme is to minimize the delay in the MAC layer. Now let the delay discount factor $\alpha \to \infty$ in (13) to emphasize the impact of delay, or let the aggregation gain $g(s)$ be a positive constant in (8) to remove the energy concern, then the optimal control-limit in either (8) or (13) is reduced to a special (extreme) case such that $s^* = 1$. This implies that as long as one or more samples have been collected, they should be aggregated and sent out at the current decision epoch (i.e. the instant that the channel is free and transmission queue is empty). In this extreme case, the control-limit policy with $s^* = 1$ is similar to the OD scheme. Therefore, the OD scheme can be viewed as a special case of the more general control-limit policies derived in this section.

IV. Finite-State Approximations for the SMDP Model

In case that the optimal policies with special structures, e.g., monotone structure, do not exist, we can look for approximate solutions of (4)-(5). Although we do not impose any restriction on the number of states and decision epochs in the original SMDP model, the number of collected samples during one aggregation operation is always finite under a finite delay tolerance in practice. Therefore, it is reasonable as well as practically useful to consider the reward and policy based on a finite-state approximation of the problem. In this section, we first introduce a finite-state approximation for the SMDP model. We verify its convergence to the original countable state-space model, and then bound its performance in terms of the reward loss between the actual achievable reward with this approximation model and the optimal reward for any given initial state $s \in S'$. For practical implementation, we

finally provide two on-line algorithms to solve the finite-state approximation and obtain the near-optimal policies.

A. A Finite-State Approximation Model

Considering the truncated state space $S_N = S'_N \cup \{\Delta\}, S'_N = \{1, 2, ..., N\}$ and setting $v_N(s) = 0, \forall s > N$, the optimality equations become

$$v_N(s) = \max\{g(s) + v_N(\Delta), \sum_{j \geq s} q^0_{sj}(\alpha) v_N(j)\} \tag{17}$$

for $s \in S'_N$ and $v_N(\Delta) = v_N(\Delta)$. Let $\mathbf{v}^*_N \geq 0$ denote the minimal solution of the optimality equation (17). Consequently $v^*_N(\Delta) = 0$.

The policy based on this finite-state approximation is given by $\pi = \mathbf{d}^\infty = \{\mathbf{d}, \mathbf{d}, ...\}$, where the decision rule $\mathbf{d}$ is

$$d(s) = \arg \max_{a \in \{0,1\}} \{a g(s) + (1-a) \sum_{j \geq s} q^0_{sj}(\alpha) v^*_N(j)\} \tag{18}$$

for $s \leq N$, $d(s) = 1$ for $s > N$ and $d(\Delta) = 0$. If we treat $v^*_N(s)$ as the approximation of the optimal reward $v^*(s)$, the above decision rule can be seen as a *greedy* rule by selecting for each state the action that maximizes the state's reward [24]. The achievable reward at any state $s \in S'$ by using this decision rule is denoted as $\tilde{v}_N(s)$ which satisfies

$$\tilde{v}_N(s) = \begin{cases} \sum_{j \geq s} q^0_{sj}(\alpha) \tilde{v}_N(j), & d(s) = 0 \\ g(s), & d(s) = 1 \end{cases}. \tag{19}$$

We note that $v^*_N(s)$ and $\tilde{v}_N(s)$ might be different in values since the former one is calculated from the finite-state approximation model and the later one is the actual achievable reward by using the greedy decision policy based on $\mathbf{v}^*_N$. Therefore, from now on, we differentiate these two values by calling the former one as the *calculated value* and the later one as the *actual value* in the finite-state approximation model.

Before going to the algorithm design for this finite-state approximation mode, we first verify the (point-wise) convergence of $v^*_N(s)$ and $\tilde{v}_N(s)$ to the optimal value $v^*(s)$ for any $s \in S'$, as the degree of finite-state approximation $N \to \infty$.

Lemma 4.1: $v^*_N(s)$ monotonically increases with N, $\forall s \in S'$.

Lemma 4.2: $v^*_N(s) \leq v^*(s)$, $\forall s \in S'$ and $\forall N > 0$.

Lemma 4.3: $\tilde{v}_N(s) \geq v^*_N(s)$, $\forall s \in S'$ and $\forall N > 0$.

Theorem 4.4: $\lim_{N\to\infty} v^*_N(s) = \lim_{N\to\infty} \tilde{v}_N(s) = v^*(s)$, $\forall s \in S'$.

Proof: See Appendix B. ∎

Recall that an aggregation operation always starts from $s = 1$, i.e., at least one sample is available at the node. Thus, if a sufficiently large value of N is chosen in the finite-state approximation model, the actual expected reward $\tilde{v}_N(1)$ of one aggregation operation will be very close to $v^*(1)$, according to Theorem 4.4.

In the finite-state approximation model, if $q^0_{sj}(\alpha), \forall s, j \in S'_N$, or equivalently, the distributions of sojourn time for all state transitions under action $a = 0$ are known *a priori*, backward induction or linear programming (LP) can be used to solve (17). The LP formulation is given by

$$\begin{array}{ll} \min & \sum_{s \in S'_N} c(s) v_N(s) \\ s.t. & v_N(s) \geq g(s), \quad s \in S'_N, \\ & v_N(s) \geq \sum_{N \geq j \geq s} q^0_{sj}(\alpha) v_N(j), \quad s \in S'_N \end{array}$$

where $c(s), s = 1, ..., N$ are arbitrary positive scalars which satisfy $\sum_{s=1}^N c(s) = 1$. On the other hand, the knowledge on the model is helpful to characterize the loss of reward between the actual value $\tilde{v}_N(s)$ and the optimal $v^*(s)$. The following theorem gives an upper-bound on such reward loss.

Theorem 4.5: Under Assumptions 2.2 and 3.3, for an N-state approximation with $N \geq 1$,

$$v^*(s) - \tilde{v}_N(s) \leq \begin{cases} \sum_{i=1}^N [(\mathbf{I} - \mathbf{Q}^{(N)})^{-1}]_{si} \psi(i), & s \leq N \\ \frac{L}{1-\beta}, & s > N \end{cases} \tag{20}$$

for $s \in S'$, where $\mathbf{Q}^{(N)} \triangleq [Q^{(N)}_{ij}]$ with $Q^{(N)}_{ij} = q^0_{ij}(\alpha)$ for $1 \leq i \leq j \leq N$ and zero otherwise, and for $1 \leq i \leq N$,

$$\psi(i) \triangleq \begin{cases} \sum_{j>N} q^0_{ij}(\alpha) \left[\frac{L}{1-\beta} + g(j)\right] \\ + \sum_{j=i}^N q^0_{ij}(\alpha)[\tilde{v}_N(j) - v^*_N(j)], & d(i) = 1 \\ \sum_{j>N} q^0_{ij}(\alpha) \frac{L}{1-\beta}, & d(i) = 0 \end{cases} \tag{21}$$

In (21), $\sum_{j>N} q^0_{ij}(\alpha)$ and $\sum_{j>N} q^0_{ij}(\alpha) g(j)$ might be estimated from actual node state transitions in aggregation operations (e.g., see the algorithm in Table I), or be bounded *a priori* by $\beta - \sum_{i \leq j \leq N} q^0_{ij}(\alpha)$ and $g(i) + L - \sum_{i \leq j \leq N} q^0_{ij}(\alpha) g(j)$, respectively, according to Assumption 3.3, where the values of β and L depend on the specific aggregation scenario (see the examples in Section III-B); the reward difference $\tilde{v}_N(j) - v^*_N(j), j \leq N$ can also be estimated from actual aggregation operations or be analytically determined (see the proof of Lemma 4.3 in Appendix H in [8] for details).

Similarly, the following corollary characterizes the loss of reward on the calculated value $v^*_N(s)$, compared to the optimal one $v^*(s)$.

Corollary 4.6: Under Assumptions 2.2 and 3.3, for an N-state approximation with $N \geq 1$,

$$v^*(s) - v^*_N(s) \leq \sum_{i=1}^N [(\mathbf{I} - \mathbf{Q}^{(N)})^{-1}]_{si} \phi(i), \quad s \leq N \tag{22}$$

for $s \in S'$, where $\phi(i) \triangleq \sum_{j>N} q^0_{ij}(\alpha) \left[\frac{L}{1-\beta} + g(j)\right]$, $1 \leq i \leq N$.

In practice, the Laplace-Stieltjes transform of the state transition distributions, $q^0_{sj}(\alpha), \forall s, j \in S'_N$ are generally unknown. Hence we should either obtain the estimated values of $q^0_{sj}(\alpha)$ from actual aggregation operations or use an alternate "model-free" method, i.e., learning a good policy without the knowledge on the state transition probabilistic model. In the following, we provide two kinds of learning algorithms for solving the finite-state approximation model.

B. Algorithm I: Adaptive Real-time Dynamic Programming

Adaptive real-time dynamic programming (ARTDP) (see [25], [26]) is essentially a kind of asynchronous value iteration scheme. Unlike the ordinary value iteration operation which needs the exact model of the system (e.g. $q^0_{ij}(\alpha)$ in our problem), ARTDP merges the model building procedure into value iteration and thus is very suitable for on-line implementation. The ARTDP algorithm for the finite-state approximation model is summarized in Table I. In line 6 of the algorithm, a value update proceeds based on current estimated system model; then a randomized action selection (i.e., *exploration*) is carried out (lines 7-9); the selected action

is then performed and the estimation of the system model (i.e., $q_{ij}^0(\alpha)$) might be updated (lines 12-16).

A key step in ARTDP is to estimate the value of $q_{ij}^0(\alpha)$ for all $i, j \in S'_N$. The integration in Laplace-Stieltjes transform can be approximated by the summation of its discrete format with time step δt. By defining $\eta(i,j,l)$ as the number of transitions from state i to j with sojourn time $\delta W_l \in [l\delta t, (l+1)\delta t), l = 0, 1, ...,$ and $\eta(i)$ as the total number of transitions from state i, we have

$$\hat{q}_{ij}^0(\alpha) \approx \sum_{l=0}^{\infty} \frac{\eta(i,j,l)}{\eta(i)} e^{-\alpha \delta W_l}. \tag{23}$$

Let $\omega(i,j) \triangleq \sum_{l=0}^{\infty} \eta(i,j,l) e^{-\alpha \delta W_l}$, the estimation of $\hat{q}_{ij}^0(\alpha)$ can be improved by updating $\omega(i,j)$ and $\eta(i)$ at each state transition as shown in lines 12-16 of Table I. Similarly, we can estimate $\sum_{j>N} q_{ij}^0(\alpha)$ and $\sum_{j>N} q_{ij}^0(\alpha) g(j)$ on-line for calculating the performance bound in Theorem 4.5, which is shown in lines 17-18 and 23-24[4].

In ARTDP, the rating of actions and exploration procedure (lines 7-9) follow the description in [26]. The calculation of the probability $P_r(a)$ for choosing action $a \in \{0,1\}$ uses the well-known Boltzmann distribution (line 9), where T is typically called the *computational temperature* which is initialized to a relative high value and decreases properly over time. The purpose of introducing randomness in action selection, instead of choosing the optimal one based on current estimation, is to avoid the overestimation of values at some states in an inaccurate model during initial iterations. When the calculated value converges to $\mathbf{v}_N^*$, the corresponding decision rule is

$$d_N^*(s) = \arg \max_{a \in \{0,1\}} \{ag(s) + (1-a) \sum_{N \geq j \geq s} \hat{q}_{sj}^0(\alpha) v_N^*(j)\} \tag{24}$$

for $s \in S'_N$ and for those $s > N$, we set $d_N^*(s) = 1$.

C. Algorithm II: Real-time Q-learning

Real-time Q-learning (RTQ) [25] provides another way for on-line calculation of the optimal reward value and policy under N-state approximation. Unlike ARTDP, RTQ does not require the estimation of $q_{ij}^0(\alpha)$ and even does not take any advantage of the semi-Markov model. It is a model-free learning scheme and relies on stochastic approximation for asymptotic convergence to the desired Q-function. It has a lower computation cost in each iteration than ARTDP but convergence is typically rather slow. In our case, the optimal Q-function is defined as $Q_*^N(s,1) = g(s)$, $Q_*^N(s,0) = \sum_{j \geq s} q_{sj}^0(\alpha) v_N^*(j)$, $\forall s \in S'_N$, $Q_*^N(s,a) = 0$, $\forall s > N, a \in \{0,1\}$ and $Q_*^N(\Delta, 0) = 0$. it is straightforward to see that $v_N^*(s) = \max_{a \in \{0,1\}} [Q_*^N(s,a)], s \in S'$. Therefore, optimizing Q-learning rule is given in line 10 and 12 for $s \in S'_N$ and $Q_{k+1}(s,0) = Q_k(s,0) = 0$ for $s > N$, where j is the next state in actual state transition. Table II gives the detailed RTQ algorithm. In RTQ, the exploration procedure (lines 7-8) is the same as the one used in ARTDP. α_k is defined as the *learning rate* at iteration k, which is generally state and action dependent. To ensure the convergence of RTQ, Tsitsiklis has shown in [27] that α_k should satisfy (1) $\sum_{k=1}^{\infty} \alpha_k = \infty$ and (2) $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$ for all states $s \in S'_N$ and actions $a \in \{0,1\}$. An example of the choice of α_k can be found in [26]. As $\alpha_k \to 0$ with $k \to \infty$, we can see that $Q_k(s_k, 1) \to g(s_k), s_k \in S'_N$. When $Q_k(s_k, a)$ converges to the optimal value $Q_*^N(s,a)$ for all states and actions, the corresponding decision rule is given by

$$d_N^*(s) = \arg \max_{a \in \{0,1\}} \{Q_*^N(s,a)\} \tag{25}$$

for $s \in S'_N$ and for those $s > N$, we set $d_N^*(s) = 1$.

[4]The lines are in comments as the estimation is optional, not mandatory in the algorithm.

TABLE I
ADAPTIVE REAL-TIME DYNAMIC PROGRAMMING (ARTDP) ALGORITHM.

1	**Set** $k = 0$
2	**Initialize** counts $\omega(i,j)$, $\eta(i)$ and $\hat{q}_{ij}^0(\alpha)$ for all $i, j \in S'_N$
3	**Repeat** {
4	Randomly choose $s_k \in S'_N$;
5	**While** ($s_k \neq \Delta$) {
6	Update $v_{k+1}(s_k) = \max\{g(s_k), \sum_{N \geq j \geq s_k} \hat{q}_{s_k j}^0(\alpha) v_k(j)\}$;
7	Rate $r_{s_k}(0) = \sum_{N \geq j \geq s_k} \hat{q}_{s_k j}^0(\alpha) v_k(j)$ and $r_{s_k}(1) = g(s_k)$;
8	Randomly choose action $a \in \{0,1\}$ according to probability
9	$P_r(a) = \frac{e^{r_{s_k}(a)/T}}{e^{r_{s_k}(0)/T} + e^{r_{s_k}(1)/T}}$;
10	**if** $a = 1$, $s_{k+1} = \Delta$;
11	**else** observe actual state transition $(s_{k+1}, \delta W_{k+1})$
12	$\eta(s_k) = \eta(s_k) + 1$;
13	**if** $s_{k+1} \leq N$,
14	Update $\omega(s_k, s_{k+1}) = \omega(s_k, s_{k+1}) + e^{-\alpha \delta W_{k+1}}$;
15	Re-normalize $\hat{q}_{s_k j}^0(\alpha) = \frac{\omega(s_k, j)}{\eta(s_k)}, \forall N \geq j \geq s_k$;
16	**else**
17	% Update $x(s_k) = x(s_k) + e^{-\alpha \delta W_{k+1}}$,
18	% Update $z(s_k) = z(s_k) + g(s_{k+1}) e^{-\alpha \delta W_{k+1}}$,
19	$a = 1$,
20	$s_{k+1} = \Delta$;
21	$k = k + 1$.}
22	}
23	% $\sum_{j>N} \hat{q}_{sj}^0(\alpha) = \frac{x(s)}{\eta(s)}, \forall s \leq N$
24	% $\sum_{j>N} \hat{q}_{sj}^0(\alpha) g(j) = \frac{z(s)}{\eta(s)}, \forall s \leq N$

TABLE II
REAL-TIME Q-LEARNING (RTQ) ALGORITHM

1	**Set** $k = 0$
2	**Initialize** Q-value $Q_k(s,a)$ for each $s \in S'_N, a \in \{0,1\}$ and set $Q_k(s,a) = 0, \forall s > N, a \in \{0,1\}$
3	**Repeat** {
4	Randomly choose $s_k \in S'_N$;
5	**While** ($s_k \neq \Delta$) {
6	Rate $r_{s_k}(0) = Q_k(s_k, 0)$ and $r_{s_k}(1) = Q_k(s_k, 1)$;
7	Randomly choose action $a \in \{0,1\}$ according to probability
8	$P_r(a) = \frac{e^{r_{s_k}(a)/T}}{e^{r_{s_k}(0)/T} + e^{r_{s_k}(1)/T}}$;
9	**if** $a = 1$, $s_{k+1} = \Delta$,
10	Update $Q_{k+1}(s_k, 1) = (1-\alpha_k) Q_k(s_k, 1) + \alpha_k g(s_k)$;
11	**else** observe actual state transition $(s_{k+1}, \delta W_{k+1})$,
12	Update $Q_{k+1}(s_k, 0) = (1-\alpha_k) Q_k(s_k, 0) + \alpha_k [e^{-\alpha \delta W_{k+1}} \max_{b \in \{0,1\}} Q_k(s_{k+1}, b)]$
13	**if** $s_{k+1} > N$, $a = 1$, $s_{k+1} = \Delta$;
14	$k = k + 1$. }
15	}

D. How Practical are the Learning Algorithms

The implementation of the above learning algorithms in practical sensor nodes is also an important concern. A similar concern on the implementation of learning algorithms for micro-robots has been investigated in artificial intelligence

society. For example, in [28], an optimized Q-learning algorithm has been implemented in a micro-robot with stringent processing and memory constraints, where the microprocessor works at a 4MHz clock frequency with a 14K-byte flash program memory and a 368-byte data memory. The authors show that an integer-based implementation of the Q-learning algorithm occupies about 3.5K bytes program memory and 48 bytes data memory where the state-action space (i.e., the table size of Q-values) in their example is $15 \times 3 = 45$. Considering that the current sensor nodes are becoming more and more powerful in processing and storage, e.g., a Crossbow mote has a 128K-byte flash program memory, a $4 \sim 8$K-byte RAM and 512K-byte flash data logger memory and its microprocessor works at a 16MHz clock frequency [9], a learning algorithm is practically implementable on current sensor nodes.

In our case, if the degree of the finite-state approximation is N and a similar integer-based implementation as that in [28] is used, ARTDP needs about $3N$ bytes and RTQ needs $2N$ bytes data memory in one decision stage. The total storage space required for ARTDP (for storing $\omega(i,j)$, $\eta(i)$, $\hat{q}^0_{ij}(\alpha)$ and $v(i)$, $i \leq j \leq N$) is $N^2 + 2N$ bytes and $2N$ bytes for RTQ (for storing $Q(i,\cdot)$). In practice, as the number of samples in one aggregation operation is usually small, a small value of N is sufficient for the finite-state approximation. For example, if N is set to be 20, ARTDP takes about 1.5% data memory (if a 4K-byte RAM is used) and 0.08% total storage space, and RTQ takes about 1.0% data memory and 0.0078% total storage space.

V. Performance Evaluation

A. Comparison of Schemes under a Tunable Traffic Model

We have considered three schemes of policy design for the decision problem in distributed data aggregation: (1) control-limit policy, including Theorem 3.1, which we call the CNTRL scheme, and its special case in (13) for a linear aggregation gain, which we call the EXPL scheme; (2) Adaptive Real-time Dynamic Programming (ARTDP); and (3) Real-time Q-learning (RTQ). Recall that CNTRL and EXPL are based on the assumption that there exists certain structure of the statistics of state transitions as specified in Theorem 3.1 and Corollary 3.2, respectively; while ARTDP and RTQ are for general cases of the problem. Except for the EXPL scheme, the computation of all the other schemes require a finite-state approximation of the original problem. We now perform a comparison of all the schemes using a tunable traffic model. The purpose of such comparison is not to exactly rank the schemes, but to qualitatively understand the effects of different traffic patterns and degrees of finite-state approximation on the performance of these schemes.

1) Traffic Model: We use a conditional exponential model for random inter-arrival time of decision epochs. That is, given the state $s \in S'$ at current decision epoch, the mean value of inter-arrival time to the next decision epoch is modelled as $\overline{\delta W}_s = \delta W_0 e^{-\theta(s-1)} + \delta W_{min}$, where $\delta W_0 + \delta W_{min}$ represents the mean value of inter-arrival time for $s = 1$, $\delta W_{min} > 0$ is a constant to avoid the possibility of an infinite number of decision epochs within finite time (e.g. see [16]) and $\theta \geq 0$ is a constant to control the degree of state-dependency. It follows that the random time interval to the next decision epoch obeys an exponential distribution with a rate[5] $\mu = 1/\overline{\delta W}_s$. For the natural process, given the state $s \in S'$ at the current decision epoch and the time interval to the next decision epoch, the number of arrived samples is assumed to be a Poisson process with a rate $\lambda_s = \lambda_0 e^{-\rho(s-1)}$, where λ_0 is a constant which represents the rate of sample arrival at state $s = 1$ and $\rho \geq 0$ is a constant to control the degree of state-dependency of the natural process. By adjusting parameters θ and ρ, we can control the degree of state-dependency of this SMDP model.

2) Comparison of Schemes: For the performance of finite-state approximations, we include an off-line LP solution as a reference, which uses the estimated $\hat{q}^0_{ij}(\alpha)$ (as described in ARTDP algorithm). With a proper randomized action selection and a large number of iterations in ARTDP, $\hat{q}^0_{ij}(\alpha)$ provides a good approximation of $q^0_{ij}(\alpha)$. Thus the solution of LP is expected to be close to $\mathbf{v}^*_N$ obtained from (20). As each decision horizon begins at state $s = 1$, we will focus on evaluating the value of the reward with this initial state. In the following, we set $\delta W_0 = 0.13$ sec, $\delta W_{min} = 0.013$ sec, $\lambda_0 = 38.5$ sample/sec, delay discount factor $\alpha = 3$ and a linear aggregation gain function $g(s) = s - 1$ for all schemes. We note that, if there is no state-dependency (i.e., $\theta = \rho = 0$) or a very low state-dependency (i.e., θ, ρ are small) in the given traffic model, the control limit in (13) can be seen as optimal. Under current model parameter setting, we have

$$s^* = \left\lceil \frac{\lambda_0 \mathbb{E}[\delta W e^{-\alpha \delta W}]}{1 - \mathbb{E}[e^{-\alpha \delta W}]} + 1 \right\rceil = \left\lceil \frac{\lambda_0 \mu}{\alpha(\alpha + \mu)} + 1 \right\rceil = 10,$$

where $\mu = 1/\overline{\delta W}_s = 1/(\delta W_0 + \delta W_{min})$.

Figure 2 shows the effect of state-dependency of the traffic on the performance of the schemes. The degree of finite-state approximation N is set to be 40. In the upper plot, $\theta = 0.001, \rho = 0.001$, represents the scenario of a low degree of state-dependency in the SMDP model. In this case, the value of the reward in the EXPL scheme is approximated to be $v^*(1)$. The values for $s = 1$ in LP and all schemes with N-state approximation are very close to that in EXPL, which demonstrates (1) the negligible truncation effect on state space for state $s = 1$ with $N = 40$; (2) the correct convergence of learning algorithms. The policies obtained from all schemes are control-limit type with the same control limit $s^* = 10$, i.e., the optimal one. In the bottom plot, $\theta = 1$ and $\rho = 1$ represents the scenario of a high degree of state-dependency in the SMDP model. As the assumption for the optimality of EXPL does not hold in this case, it converges to a lower value of reward than the other schemes. The policies obtained from ARTDP, RTQ, CNTRL and LP are control-limit type with $s^* = 3$ while EXPL gives a control limit at 4.

From Lemma 4.3 and Theorem 4.4, we already know that, when the truncation effect of the state space at state s is non-negligible, i.e., N is not large enough for state s, the calculated value $v^*_N(s)$ is different from the actual value $\tilde{v}_N(s)$, and when N is sufficiently large with respect to s, both $v^*_N(s)$ and $\tilde{v}_N(s)$ converge to the optimal value $v^*(s)$. Here we experimentally show such impact of finite-state approximation

[5]The distribution is set to be unchanged even if there are state transitions during the interval.

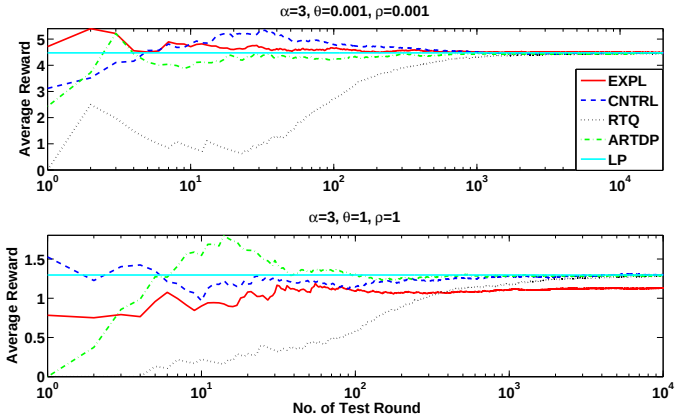

Fig. 2. Convergence of the values of the reward for initial state $s = 1$ in EXPL, CNTRL, ARTDP and RTQ under different traffic patterns: $\theta = 0.001, \rho = 0.001$, i.e., a low degree of state-dependency (upper) and $\theta = 1, \rho = 1$, i.e., a high degree of state-dependency (bottom); delay discount factor $\alpha = 3$; finite-state approximation $N = 40$. The different degrees of the state-dependency affect the optimality of the EXPL scheme.

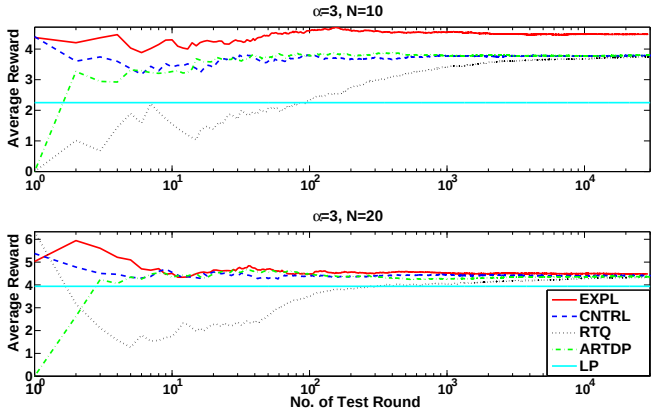

Fig. 3. Convergence of the values of the reward for initial state $s = 1$ in EXPL, CNTRL, ARTDP and RTQ under different degrees of finite-state approximation: $N = 10$ (upper) and $N = 20$ (bottom); delay discount factor $\alpha = 3$; traffic pattern $\theta = 0.001, \rho = 0.001$, i.e., a low degree of state-dependency. A larger reward value loss occurs on the schemes when a larger degree of state-space truncation (i.e., a smaller N) is used.

on the performance of the schemes in Fig. 3 and Table III. We consider $\theta = 0.001, \rho = 0.001$ in which the EXPL scheme provides a value of 4.48 (initial state $s = 1$) and a (optimal) control-limit policy at $s^* = 10$. In the upper plot, $N = 10$, the actual values of the reward with initial state $s = 1$ in ARTDP, RTQ and CNTRL converge to values ($\approx 3.78 \sim 3.80$) lower than that in EXPL but significantly higher than the calculated values in LP and learning algorithms (LP: 2.26, ARTDP: 2.26 and RTQ: 2.25). This is because the calculated values are based on (17) in which $v_N^*(s) = 0, s > N$. When the probability of transition from $s = 1$ to a state beyond N is non-negligible in actual aggregation operations, the calculated values underestimate the actual reward. On the other hand, the policies obtained from ARTDP, RTQ and CNTRL are exactly the same as the one in LP, i.e., $s^* = 4$, which is far from the optimal control-limit $s^* = 10$. When $N = 20$, we see that the actual performance gap between finite-state approximations and EXPL becomes smaller even though the calculated values (LP: 3.94, ARTDP: 3.94 and RTQ: 3.93) still give a conservative estimation of the reward at $s = 1$. The policies given by finite-state approximations are improved to

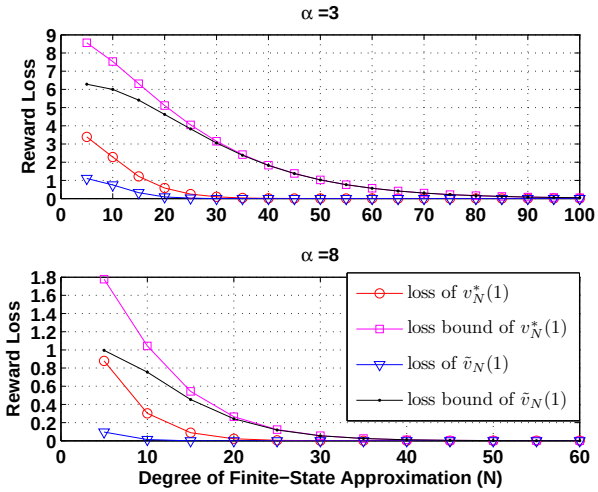

Fig. 4. The comparison of the reward loss bounds for $s = 1$ (in Theorem 4.5 and Corollary 4.6) and the simulated reward losses $v^*(1) - \tilde{v}_N(1)$ and $v^*(1) - v_N^*(1)$, under different N and discount factor α (= 3 and 8, respectively); traffic pattern $\theta = \rho = 0$. The bounds provide a way in setting the least degree of finite-state approximation (i.e., N) to satisfy a certain performance guarantee.

have a control limit $s^* = 8$. Further improvement at $N = 40$ for finite-state approximation has been shown in Table III and Fig. 2, in which the control-limits in all schemes have converged to the optimal one. On the other hand, comparing the two learning algorithms, we find that for all cases, both schemes converge to similar values in reward and identical policies, but ARTDP shows a faster convergence speed than RTQ. This demonstrates the benefit of using the SMDP model in ARTDP. The slow convergence partially counteracts the computational benefit of RTQ.

3) Evaluation of the Reward Loss Bounds in Theorem 4.5 and Corollary 4.6: By setting $\theta = \rho = 0$ in the given traffic model, we can numerically evaluate the reward loss bounds in Theorem 4.5 and Corollary 4.6 for the finite-state approximation model. With some manipulation, it is not hard to show that, for any $s \leq N$

$$q_{sj}^0(\alpha) = \left(\frac{\mu}{\alpha+\mu+\lambda_0}\right)\left(\frac{\lambda_0}{\alpha+\mu+\lambda_0}\right)^{j-s}, j \geq s,$$
$$\sum_{j>N} q_{sj}^0(\alpha) = \left(\frac{\mu}{\alpha+\mu}\right)\left(\frac{\lambda_0}{\alpha+\mu+\lambda_0}\right)^{N+1-s},$$
$$\sum_{j>N} q_{sj}^0(\alpha) g(j) = \left(\frac{\mu}{\alpha+\mu}\right)\left(\frac{\lambda_0}{\alpha+\mu+\lambda_0}\right)^{N+1-s}\left(N + \frac{\lambda_0}{\alpha+\mu}\right).$$

With (4), (17) and (19), we can numerically solve $v^*(s), v_N^*(s)$ and $\tilde{v}_N(s)$ for any $s \leq N$. Furthermore, by setting $\beta = \mathbb{E}[e^{-\alpha\delta W}] = \mu/(\alpha+\mu)$ and $L = \mathbb{E}[Xe^{-\alpha\delta W}] = \lambda_0\mu/[\alpha(\alpha+\mu)]$, we numerically evaluate the reward loss bounds in Theorem 4.5 and Corollary 4.6. Figure 4 illustrates the simulated reward losses $v^*(1) - \tilde{v}_N(1)$, $v^*(1) - v_N^*(1)$ and the bounds, under different degrees of finite-state approximation and different delay discount factors. From Fig. 4, we find that, by calculating the bounds, we can guarantee the performance of the finite-state approximation by adjusting N, without knowing the optimal value of the original infinite-state model in (4). For example, by setting $N > 25$ in the case that $\alpha = 8$, we can ensure that the reward loss of the finite-state approximation is no worse than 0.1. On the other hand,

TABLE III
VALUES AND POLICIES IN THE SCHEMES WITH FINITE-STATE APPROXIMATION ($\alpha = 3$, $\theta = 0.001$, $\rho = 0.001$)

	Calculated Value ($s = 1$)			Actual Value ($s = 1$)			Control limit s^*	
N	LP	ARTDP	RTQ	ARTDP	RTQ	CNTRL	LP	ARTDP/RTQ/CNTRL
10	2.26	2.26	2.25	3.80	3.77	3.77	4	4
20	3.94	3.94	3.93	4.36	4.34	4.41	8	8
40	4.47	4.47	4.46	4.47	4.46	4.48	10	10

although the reward loss bounds are rather conservative under this traffic setting, we note that the traffic model under the evaluation is unfavorable to the bounds since the optimal policy is the control-limit type. In such case, $v^*(s) - g(s) = 0$ for any state s which is larger than the optimal control-limit while the bounds have no knowledge on this optimal policy structure and still use the general result in Lemma 3.6 for any s larger than N and the optimal control-limit. We emphasize that the bounds in Theorem 4.5 and Corollary 4.6 provide a general characterization on the performance of the finite-state approximation, though it would be possible to develop tighter bounds on the reward loss with some *a priori* knowledge or conjecture on the optimal value and/or the structure of the optimal policy.

B. Evaluation in Distributed Data Aggregation

We provide further simulation to evaluate of the proposed schemes as well as the existing schemes in the literature (i.e. the OD and FIX schemes) in a distributed data aggregation scenario in which each sensor in the network is expected to track the time-varying maximum value of an underlying dynamic phenomenon in the sensing field that the network resides in. The phenomenon model, the nodal communications procedures and all aggregation schemes are implemented in MATLAB. In our simulator, each sensor node is an entity where all nodal communications procedures and aggregation algorithms run.

The dynamic phenomenon in the sensing field is modelled as a spatial-temporal correlated discrete Gauss-Markov process $\mathbf{Y}(t) = \mathbf{C} + \mathbf{X}(t)$, where $\mathbf{C}$ is a constant vector and $\mathbf{X}(t)$ is a first-order Markov process with the spatial distribution to be Gaussian. As we are only concerned with the values of the phenomenon sampled at the sensor nodes, $\mathbf{Y}(t), \mathbf{C}$ and $\mathbf{X}(t)$ are in $\mathbb{R}^l$, where l is the number of sensor nodes in the network. In the simulation, $\mathbf{C} = \mathbf{1}$ and $\mathbf{X}(t)$ is zero-mean with variance 0.1 and intensity of correlation[6] 0.001.

There are 25 sensor nodes randomly deployed in a two-dimensional sensing field of size $30m \times 40m$. Each node is equipped with an omnidirectional antenna and the transmission range is $10m$. The data rate for inter-node communication is set as 38.4 kbps and the energy model of individual nodes is: 686 nJ/bit (27 mW) for radio transmission, 480 nJ/bit (18.9 mW) for reception, 549 nJ/bit (21.6 mW) for processing and 343 nJ/bit (13.5 mW) for sensing, which are estimated from the specifications of the Crossbow mote MICA2 [9]. Nodes sample the field to obtain the local values of the phenomenon, according to a given sampling rate. The size of a (original) sample is 16 bits, including the information of the sample value and the instant of sampling.

In this data aggregation simulation, when a node receives samples from its neighbors, only the samples from the same sampling instant are aggregated (i.e. selecting the one with the maximum value as the aggregated sample). Also, the repeated samples[7] are dropped. Transmissions are broadcast, under the control of a random access MAC model which is assumed ideal in avoiding collisions. A transmitted packet concatenates the samples which are from different sampling instants and needed for transmission at the transmission epoch. The packets transmitted at different transmission epochs thus might be in variable-size. Since all packets are broadcasted, no packet header is considered in the simulation. The delay discount factor is set as $\alpha = 8$ and the degree of finite-state approximation is set as $N = 10$. The linear function $g(s) = s - 1$ is used as the nominal aggregation gain since the energy saving in this data aggregation procedure is approximately proportional to the number of samples aggregated. For the FIX scheme, we consider three different DOAs, i.e., DOA$= 3, 5, 7$, which is based on the observation that the DOA on different nodes varies from 2 to 7 in simulating the proposed schemes.

Figure 5 shows the average reward (initial state $s = 1$) obtained by each scheme during aggregation operations, where the average is over all aggregation operations at all nodes after the scheme reaches its steady state. RTQ and ARTDP achieve the best performance among all schemes as they do not rely on any special structure of state transition distributions. CNTRL also shows a higher reward than EXPL as it relies on a weaker assumption (in Theorem 3.1). All the proposed schemes in this paper have shown a significant gain in reward over OD and FIX schemes with DOA$= 3, 7$. One exception is the FIX with DOA$= 5$, which achieves a higher reward than EXPL when the sampling rate is higher than 7 Hz and a comparable reward value to CNTRL when the sampling rate is above 13 Hz. However, the performance of FIX is very sensitive to the setting of DOA, which can be seen from the significant performance difference of FIX with DOA$= 3, 5, 7$. Furthermore, the proper setting of DOA in FIX relies on the *a priori* knowledge of the range of DOA in actual aggregation operations (e.g., the DOA setting for FIX here is based on the simulation results on the proposed schemes), which is generally unknown during the setup phase of aggregation.

Figure 6 evaluates the average delay for collecting the time-varying maximum values of the field in each scheme, where the delay at a specific node is defined as the time duration from the sampling instant of a maximum value to the instant that the node receives it. The average is over all maximum

[6]The spatial correlation of two samples separated by distance d_{ij} is $\exp[-\kappa d_{ij}]$, where κ is defined as the intensity of correlation [29].

[7]The repeated sample is a received sample with the value no greater than the ones that are from the same sampling instant and have been transmitted by the node in previous decision horizons.

values collected at all nodes, after a scheme reaches its steady state. Note that, as we did not consider any transmission loss and noise in reception, this delay (i.e., tracking lag) provides an appropriate metric for evaluating tracking performance [30]. OD, RTQ and ARTDP have a similar delay performance which is slightly higher than CNTRL and lower than EXPL. The delay performance of FIX is very sensitive to the sampling rate as it can not dynamically adjust its DOA in response to different network congestion scenarios.

Energy costs for tracking the maximum values in different schemes are compared in Fig. 7, where the energy cost of any scheme is averaged over all maximum values collected at all nodes, after the scheme reaches its steady state. OD shows an overall highest energy cost as aggregation for energy saving is only opportunistic. FIX with DOA$=7$ costs the least energy as it has the highest DOA among all schemes (see Fig. 8). However, this does not mean a higher DOA is better since aggregation delay should be taken into consideration. Again, RTQ and ARTDP have similar performance in energy cost. From Fig. 6 and 7, we can clearly see a delay-energy tradeoff in the schemes (except FIX with DOA$=3$). Among them, RTQ and ARTDP achieve the best balance between delay and energy.

Figure 8 gives the average DOA, i.e., the number of samples collected per aggregation operation, in each scheme under different sampling rates, where the average is over all aggregation operations at all nodes, after the scheme reaches its steady state. It is clear that the proposed schemes and OD can adaptively increase their DOAs as the sampling rate increases. On the other hand, Figure 9 shows the average DOAs at different nodes under a given sampling rate (11 Hz), where the average is over the aggregation operations at a specific node. In Fig. 9, node 1 has three neighbors, node 7 has five neighbors and node 9 has six neighbors. Different node degrees implies different channel contentions and sample arrival rates. At node 1, with the lowest node degree among the three nodes, the schemes (except FIX) have the lowest DOAs. DOAs increase with the node degree in the proposed schemes as well as OD. This demonstrates the difference between the proposed control-limit policies and the previously proposed FIX scheme, as described in Section III-C, i.e., the control limit s^* in the proposed schemes is adaptive to the environment and the sampling rate, not as rigid as in the FIX scheme.

VI. Conclusions

In this paper, we have provided a stochastic decision framework to study the fundamental energy-delay tradeoff in distributed data aggregation in wireless sensor networks. The problem of balancing the aggregation gain and the delay experienced in aggregation operations has been formulated as a sequential decision problem which, under certain assumption, becomes a semi-Markov decision process (SMDP). The practically attractive *control-limit* type policies for the decision problem have been developed. Furthermore, we have proposed a finite-state approximation for the general case of the problem and provided two learning algorithms for solution. ARTDP has shown a better convergence speed than RTQ with a cost of computation complexity in learning the system model.

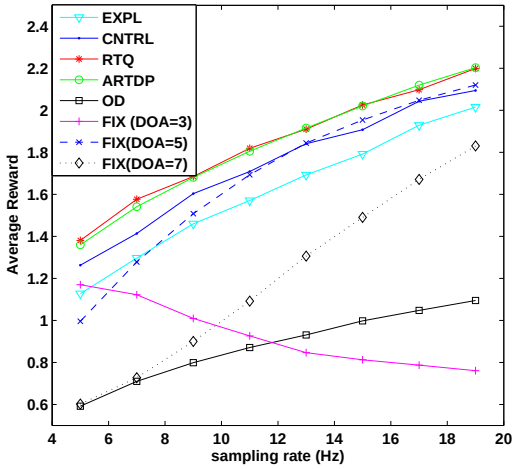

Fig. 5. Average rewards of EXPL, CNTRL, ARTDP, RTQ, OD and FIX in a distributed data aggregation; delay discount factor $\alpha = 8$, finite-state approximation $N = 10$. The control-limit type policies (i.e., CNTRL, EXPL) and the FIX scheme can achieve a close performance to the learning schemes (i.e., ARTDP, RTQ), while the FIX scheme is sensitive to the setting of DOA.

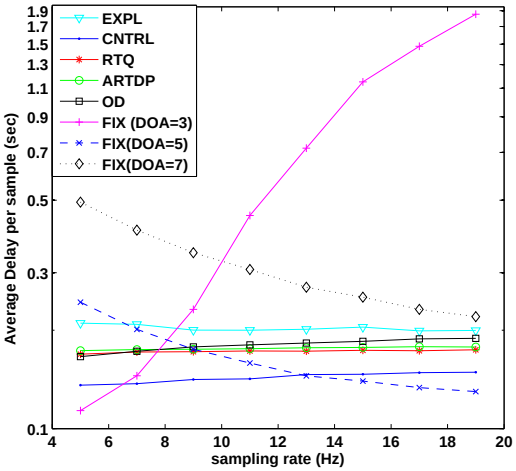

Fig. 6. Delay performance of EXPL, CNTRL, ARTDP, RTQ, OD and FIX in a distributed data aggregation; delay discount factor $\alpha = 8$, finite-state approximation $N = 10$. The y-axis is in logarithmic scale. The delay performance of the FIX scheme is sensitive to the setting of DOA.

The simulation on a practical distributed data aggregation scenario has shown that ARTDP and RTQ can achieve the best performance in balancing energy and delay costs, while the performance of control-limit type policies, especially the EXPL scheme in (13), is close to that of learning algorithms, but with a significantly lower implementation complexity. All the proposed schemes have outperformed the traditional schemes, i.e., the fixed degree of aggregation (FIX) scheme and the on-demand (OD) scheme.

Appendix

A. Proof of Theorem 3.1

As the state evolution of the node is non-decreasing, the satisfaction of (7) for all states $i \geq s$ once it holds for state $s \in S'$ implies that once the 1-sla decision rule calls for stopping

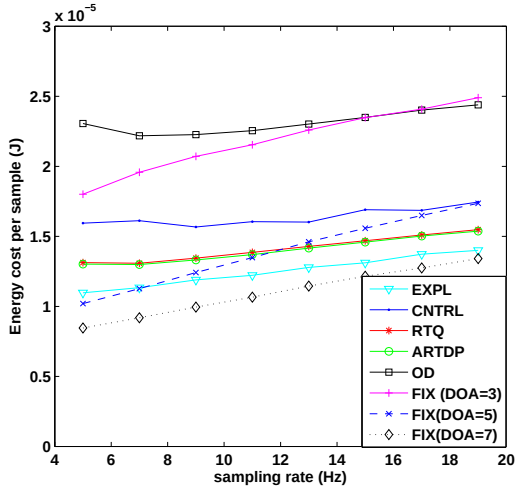

Fig. 7. Energy consumption (per sample) of EXPL, CNTRL, ARTDP, RTQ, OD and FIX in data aggregation; delay discount factor $\alpha = 8$, finite-state approximation $N = 10$.

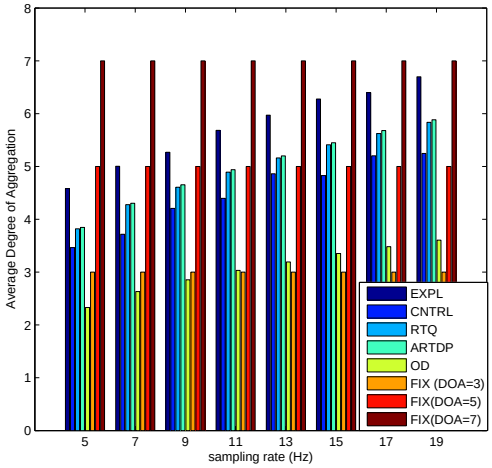

Fig. 8. Average degrees of aggregation (DOA) versus different sampling rates in EXPL, CNTRL, ARTDP, RTQ, OD and FIX in data aggregation; delay discount factor $\alpha = 8$, finite-state approximation $N = 10$. The proposed schemes and the OD scheme can adapt DOA with the sampling rates.

at the current decision epoch, it will always call for stopping in the following decision epochs. Thus the problem is *monotone* (see Chapter 5, [18]). Therefore, under Assumption 2.2, the 1-sla decision rule is optimal [18] and the optimal stopping instant is the first decision epoch with state $s \geq s^*$, where $s^* = \min\{s \geq 1 : g(s) \geq \sum_{j \geq s} q_{sj}^0(\alpha) g(j)\}$. As the 1-sla calls for stopping at any state $s \geq s^*$ and continuing for $s < s^*$, s^* is a control limit and the corresponding policy is optimal.

Next, we show that the optimal reward is given by (9). The decision rule $\mathbf{d} = [d(1), d(2), \dots]^T$ is given in (6) with control limit s^* in (8). The corresponding stationary policy is $\mathbf{d}^\infty = (\mathbf{d}, \mathbf{d}, \dots)$. Let the reward achieved by this policy be $\tilde{\mathbf{v}} \triangleq [\tilde{v}(1), \tilde{v}(2), \dots]^T$, $\mathbf{g} \triangleq [g(s^*), g(s^*+1), \dots]^T$ and $\mathbf{M}_d^{S'} \triangleq [\mathbf{A}\ \mathbf{B}; \mathbf{0}\ \mathbf{0}]$, where $\mathbf{A}$ and $\mathbf{B}$ are defined in (10) and (11), respectively. We have

$$\tilde{\mathbf{v}} = \mathbf{r}_d + \mathbf{M}_d^{S'} \tilde{\mathbf{v}} \tag{26}$$

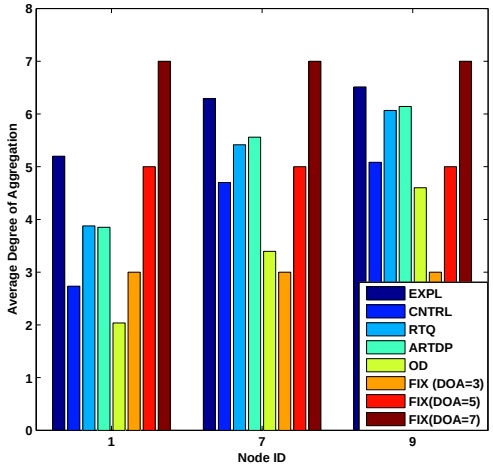

Fig. 9. Average degrees of aggregation (DOA) of EXPL, CNTRL, ARTDP, RTQ, OD and FIX at different nodes: Node 1 (node degree =3), Node 7 (node degree = 5) and Node 9 (node degree = 6); sampling rate is set as 11 Hz. The proposed schemes and the OD scheme can adapt DOA with the local traffic intensity.

where $\mathbf{r}_d \triangleq [\mathbf{0}_{1\times(s^*-1)}^T\ \mathbf{g}^T]^T$. It is straightforward to see that $\tilde{v}(s) = g(s), \forall s \geq s^*$. Let $\tilde{\mathbf{v}}^{s^*} \triangleq [\tilde{v}(1), \tilde{v}(2), ..., \tilde{v}(s^*-1)]^T$, with (26), we have $\tilde{\mathbf{v}}^{s^*} = \mathbf{A}\tilde{\mathbf{v}}^{s^*} + \mathbf{Bg}$. As $0 \leq \lambda(\mathbf{A}) < 1$, $(\mathbf{I} - \mathbf{A})$ is nonsingular. The result for the case $s < s^*$ in (9) follows by noting that $\tilde{\mathbf{v}}^{s^*} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{Bg} = \mathbf{H}(\alpha)\mathbf{g}$.

B. Proof of Theorem 4.4

From Lemmas 4.1 and 4.2, for any $s \in S'$, $v_N^*(s)$ converges as $N \to \infty$, denoted the limit by $v'(s)$. With (17),

$$v'(s) = \lim_{N\to\infty} \max\{g(s), \sum_{j\geq s} q_{sj}^0(\alpha) v_N^*(j)\}.$$

On the other hand, with Lemma 4.1 and 4.2, $\sum_{j\geq s} q_{sj}^0(\alpha) v_N^*(j)$ is monotonically increasing with N and bounded by $v^*(s)$ from the above, thus it converges. If $\lim_{N\to\infty} \sum_{j\geq s} q_{sj}^0(\alpha) v_N^*(j) \leq g(s)$, $v'(s) = g(s)$; otherwise, $\exists N^* \geq s$ such that $\sum_{j\geq s} q_{sj}^0(\alpha)\ v_N^*(j) > g(s), \forall N \geq N^*$, or equivalently, $\lim_{N\to\infty} \sum_{j\geq s} q_{sj}^0(\alpha) v_N^*(j) > g(s)$, then $v'(s) = \lim_{N\to\infty} \sum_{j\geq s} q_{sj}^0(\alpha) v_N^*(j)$. To show

$$\lim_{N\to\infty} \sum_{j\geq s} q_{sj}^0(\alpha) v_N^*(j) = \sum_{j\geq s} q_{sj}^0(\alpha) v'(j),$$

choose $\varepsilon > 0$. Then for any $n > 0$,

$$\begin{aligned}\sum_{j\geq s} q_{sj}^0(\alpha)[v'(j) - v_N^*(j)] &= \sum_{n\geq j\geq s} q_{sj}^0(\alpha)[v'(j) - v_N^*(j)] \\ &\quad + \sum_{j>n} q_{sj}^0(\alpha)[v'(j) - v_N^*(j)].\end{aligned}$$

As $0 \leq \sum_{j\geq s} q_{sj}^0(\alpha)[v'(j) - v_N^*(j)] \leq \sum_{j\geq s} q_{sj}^0(\alpha) v'(j) \leq v^*(s) < \infty$, then for each $s \in S'$, we can find an n' so that $\sum_{j>n} q_{sj}^0(\alpha) v'(j) < \varepsilon/2$ for all $n \geq n'$. Thus the second summation is less than $\varepsilon/2$. Choose $n \geq n'$, the first summation can be made less than $\varepsilon/2$ by choosing N sufficiently large. Thus $\lim_{N\to\infty} \sum_{j\geq s} q_{sj}^0(\alpha) v_N^*(j) = \sum_{j\geq s} q_{sj}^0(\alpha) v'(j)$ and

$$v'(s) = \max\{g(s), \sum_{j\geq s} q_{sj}^0(\alpha) v'(j)\}$$

for each $s \in S'$. As $v'(\Delta) = 0$, $\mathbf{v}' \geq 0$ is a solution of the original optimality equations in Section II-B. As $\mathbf{v}^* \geq 0$

is the minimal solution, $\mathbf{v}' \geq \mathbf{v}^*$. However, as $v'(s) = \lim_{N\to\infty} v_N^*(s) \leq v^*(s)$, $\forall s \in S'$ from Lemma 4.2, thus $v'(s) = v^*(s), \forall s \in S'$.

On the other hand, we note that $\tilde{v}_N(s) \leq v^*(s), \forall s \in S'$ and from Lemma 4.3, we also have $\tilde{v}_N(s) \to v^*(s), \forall s \in S'$ as N goes to infinity.

REFERENCES

[1] A. Boulis, S. Ganeriwal, and M. B. Srivastava, "Aggregation in sensor networks: an energy-accuracy trade-off," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 317–331, 2003.

[2] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proc. of ACM Int'l Conf. Info. Processing in Sensor Networks*, Nashville, TN, Apr. 2006, pp. 168–176.

[3] J.-Y. Chen, G. Pandurangan, and D. Xu, "Robust computation of aggregates in wireless sensor networks: distributed randomized algorithms and analysis," in *Proc. of IEEE Int'l Conf. Info. Processing in Sensor Networks*, Los Angeles, CA, Apr. 2005, pp. 348–355.

[4] V. Delouille, R. Neelamani, and R. Baraniuk, "Robust distributed estimation in sensor networks using embedded polygons algorithm," in *Proc. of IEEE Int'l Conf. Info. Processing in Sensor Networks*, Berkeley, CA, Apr. 2004, pp. 405–413.

[5] R. Cristescu and M. Vetterli, "On the optimal density for real-time data gathering of spatio-temporal processes in sensor networks," in *Proc. of IEEE Int'l Conf. Info. Processing in Sensor Networks*, Los Angeles, CA, Apr. 2005, pp. 159–164.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. of Hawaii Int'l Conf. Syst. Sciences*, Manoa, HI, Jan. 2000, pp. 1–10.

[7] I. F. Akyildiz, M. C. Vuran, O. B. Akan, and W. Su, "Wireless sensor networks: A survey revisited," *Computer Networks*, 2006.

[8] Z. Ye, A. A. Abouzeid, and J. Ai, *Optimal Policies for Distributed Data Aggregation in Wireless Sensor Networks*. Troy, NY: Tech. Report, ECSE Dept., RPI, 2008.

[9] Crossbow, *MPR/MIB User's Manual Rev. A, Doc. 7430-0021-08*. San Jose, CA: Crossbow Technology, Inc., 2007.

[10] T. He, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, "AIDA: Adaptive application-independent data aggregation in wireless sensor networks," *ACM Trans. Embedded Comput. Syst*, vol. 3, no. 2, pp. 426–457, May 2004.

[11] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proc. of ACM Int'l Conf. Mobile Computing and Networking*, Boston, MA, Aug. 2000, pp. 56–67.

[12] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: A tiny AGgregation service for ad-hoc sensor networks," in *Proc. of Symp. Operating Syst. Design and Implementation*, Boston, MA, Dec. 2002.

[13] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proc. of ACM Int'l Conf. Mobile Computing and Networking*, Seattle, WA, Aug. 1999, pp. 174–185.

[14] I. Solis and K. Obraczka, *In-network Aggregation Trade-offs for Data Collection in Wireless Sensor Networks*. Santa Cruz, CA: Tech. Report, Computer Science Dept., UCSC, 2003.

[15] F. Hu, X. Cao, and C. May, "Optimized scheduling for data aggregation in wireless sensor networks," in *Proc. of Int'l Conf. Info. Technology: Coding and Computing*, Las Vegas, NE, Apr. 2005, pp. 557–561.

[16] M. L. Puterman, *Markov Decision Processes—Discrete Stochastic Dynamic Programming*. New York, NY: John Wiley & Sons, Inc., 1994.

[17] E. Altman, "Applications of markov decision processes in communication networks," *Handbook of Markov Decision Processes: Methods and Applications*, pp. 489–536, 2002.

[18] T. S. Ferguson, *Optimal Stopping and Applications*, on-line: http://www.math.ucla.edu/∼tom/Stopping/Contents.html, 2004.

[19] Y. S. Chow, H. Robbins, and D. Siegmund, *Great Expectations: The Theory of Optimal Stopping*. Boston: Houghton Mifflin Co., 1971.

[20] T. Ferguson, "A poisson fishing model," *Festschrift for Lucien Le Cam - Research Papers in Probability and Statistics*, pp. 235–244, 1997.

[21] N. Starr and M. Woodroofe, *Gone fishin': Optimal Stopping based on Catch Times*. Ann Arbor, MI: Tech. Report, Statistics Dept., Univ. of Michigan, 1974.

[22] A. Z. Broder and M. Mitzenmacher, "Optimal plans for aggregation," in *Proc. of ACM Symp. Principles of Distributed Computing*, Monterey, CA, 2002.

[23] I. Demirkol, C. Ersoy, and F. Alagoz, "Mac protocols for wireless sensor networks: a survey," *IEEE Comm. Magazine*, pp. 115–121, 2006.

[24] S. P. Singh and R. C. Yee, "An upper bound on the loss from approximate optimal-value functions," *Machine Learning*, vol. 16, no. 3, pp. 227–233, 1994.

[25] A. G. Barto, S. J. Bradtke, and S. P. Singh, "Learning to act using real-time dynamic programming," *Artificial Intelligence*, vol. 72, no. 1-2, pp. 81–138, Jan. 1995.

[26] S. J. Bradtke, "Incremental dynamic programming for online adaptive optimal control," Ph.D. dissertation, Univ. of Massachusetts, Amherst, MA, 1994.

[27] J. N. Tsitsiklis, *Asynchronous Stochastic Approximation and Q-learning*. Cambridge, MA: Tech. Report LIDS-P-2172, MIT, 1993.

[28] M. Asadpour and R. Siegwart, "Compact q-learning optimized for micro-robots with processing and memory constraints," *Robotics and Autonomous Systems*, no. 48, pp. 49–61, 2004.

[29] N. Cressie, *Statistics for Spatial Data*. US: John Wiley and Sons, 1991.

[30] S. Haykin, *Adaptive Filter Theory*. London: Prentice-Hall, 2001.

Zhenzhen Ye (S'07) received the B.E. degree from Southeast University, Nanjing, China, in 2000, the M.S. degree in high performance computation from Singapore-MIT Alliance, National University of Singapore, Singapore, in 2003, and the M.S. degree in electrical engineering from University of California, Riverside, CA in 2005. He is currently working towards the Ph.D. degree in electrical engineering in Rensselaer Polytechnic Institute, Troy, NY.

His research interests lie in the areas of wireless communications and networking, including stochastic control and optimization for wireless networks, cooperative communications in mobile ad hoc networks and wireless sensor networks, and ultra-wideband communications.

Alhussein A. Abouzeid received the B.S. degree with honors from Cairo University, Cairo, Egypt in 1993, and the M.S. and Ph.D. degrees from University of Washington, Seattle, WA in 1999 and 2001, respectively, all in electrical engineering.

From 1993 to 1994 he was with the Information Technology Institute, Information and Decision Support Center, The Cabinet of Egypt, where he received a degree in information technology. From 1994 to 1997, he was a Project Manager in Alcatel telecom. He held visiting appointments with the aerospace division of AlliedSignal (currently Honeywell), Redmond, WA, and Hughes Research Laboratories, Malibu, CA, in 1999 and 2000, respectively.

He is currently Associate Professor of Electrical, Computer, and Systems Engineering, and Deputy Director of the Center for Pervasive Computing and Networking, Rensselaer Polytechnic Institute (RPI), Troy, NY.

His research interests span various aspects of computer networks. He is a recepient of the Faculty Early Career Development Award (CAREER) from the US National Science Foundation in 2006. He is a member of IEEE and ACM and has served on several technical program and executive committees of various conferences. He is also a member of the editorial board of Computer Networks (Elsevier).

Jing Ai (S'05) received his Ph.D. degree in Computer Systems Engineering from Rensselaer Polytechnic Institute in August 2008. He received his B.E. and M.E. in the Electrical Engineering from Huazhong University of Science and Technology (HUST) in 2000 and 2002, respectively. He is now a member of technical staff in Juniper Networks.

His research interests include coverage and connectivity in wireless sensor networks, dynamic resource allocation, stochastic scheduling and cross-layer design in various types of wireless networks, e.g., wireless ad hoc networks and cognitive radio networks.