This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2007 proceedings.

Optimal Policies for Distributed Data Aggregation in Wireless Sensor Networks

Zhenzhen Ye, Alhussein A. Abouzeid and Jing Ai
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180-3590, USA
Email: {yez2, aij}@rpi.edu, abouzeid@ecse.rpi.edu

Abstract— **We consider the scenario of *distributed data aggregation* in wireless sensor networks, where each sensor can obtain and estimate the information of the whole sensing field through local data exchange and aggregation. The intrinsic trade-off between energy and delay in aggregation operations imposes a crucial question on nodes to decide optimal instants for forwarding their samples. The samples could be composed of the information from their own sensor readings or an aggregation of information with other 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 due to the asynchronous nature of information exchange among neighboring nodes, we propose a *decision process* model to analyze this problem and determine the optimal decision policies at nodes with local information. We show 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, we provide two learning algorithms to solve a finite-state approximation model of the decision problem. Simulations on a practical distributed data aggregation scenario demonstrate the effectiveness of the developed policies, which can also achieve a desired energy-delay tradeoff.**

I. Introduction

Data aggregation is recognized as one of the basic distributed data processing procedures in wireless sensor networks for saving energy and reducing medium access layer contention. We consider the scenario of *distributed data aggregation* where each sensor can obtain and estimate the information of the whole sensing field through data exchange and aggregation with its 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. Authors in [1] 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 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 considered as it is directly related to the accuracy of the information (e.g. see [2]). This is especially true for some time-sensitive applications in large-scale wireless sensor networks, such as environment monitoring, disaster relief and target tracking. Therefore, *the fundamental trade-off between energy and delay imposes a decision-making problem in aggregation operations.* A node should decide when is the optimal time instant for sending out the aggregated information with its local knowledge of the randomness of sample arrival as well as channel contention.

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 the impact of delay into account. We show that, once the statistics of sample arrival and the availability of the multi-access channel satisfy certain conditions (described in Section III), there exists simple *control-limit* type policies (see [3] [4] for an overview of optimal stopping theory and control-limit policies) which are easy to implement in practice. In the case that the required conditions are not satisfied, we propose a finite-state approximation for the original decision problem, and verify its convergence as a function of the degree of the truncated state space. Then we provide two on-line algorithms, adaptive real-time dynamic programming (ARTDP) and real-time Q-learning (RTQ), to solve the finite-state approximation. The numerical properties of the proposed policies are investigated in Section V-A 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 desired energy-delay balance, compared to previous fixed degree of aggregation (FIX) scheme and on-demand (OD) aggregation scheme [5].

Up to 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 [6], TAG [7], SPIN [8] and Cascading timeout [9]. 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

0743-166X/07/$25.00 ©2007 IEEE

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 [10], in which the authors propose a simple centralized feedback timing control for tree-based aggregation. In their scheme, the maximum duration for one data aggregation operation is set by the sink with the knowledge of the information quality in the previous aggregation operation. Distributed control for DOA is introduced in [5]. 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 [5] 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 here.

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. [11]). 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 Figure 1 for a schematic diagram illustrating these operations.

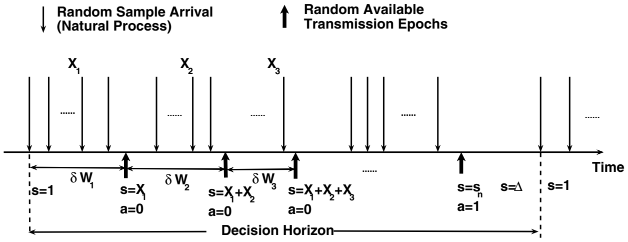

Fig. 1. A schematic illustration of the decision process model for data aggregation. The decisions are made at available transmission epochs; with the observation of the current node's state s, i.e., the number of samples collected, and the elapsed time $\sum_i \delta W_i$, action a is selected (0: continuing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

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 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 1: *Given the state $s_n \in S'$ at the nth decision epoch, if the decision is to continue to wait, then 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 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 $\{S, A, \{Q^a_{ij}(\tau)\}, R\}$, which are the state space S, action set A, a set of action dependent state transition distributions $\{Q^a_{ij}(\tau)\}$ 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;
- $Q^a_{ij}(\tau) \triangleq P_r\{\delta W_{n+1} \leq \tau, s_{n+1} = j | s_n = i, a\}$, $i, j \in S, a \in A_i$ is the transition distribution from state i to j given the action at state i is a; $Q^1_{i\Delta}(\tau) = u(\tau)$ for $i \in S'$ and $Q^0_{\Delta\Delta}(\tau) = u(\tau)$, 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 $g(1) = 0$ and $g(s) \geq 0,\ s > 1$ as the aggregation gain achieved by aggregating s samples when stopping.

The specific form of $g(s)$ depends on the application. For example, for certain types of queries in sensor networks such as maximum/minimum, average, count, etc, a linear aggregation gain $g(s)$ may be used, as the energy saving in transmission is roughly proportional to the number of samples aggregated. One should also notice that the actual energy gain by aggregation might be complicated in some cases, not purely determined by the number of collected samples. In these cases, we could redefine the *state* of a node in our model by incorporating the factors that affect energy gain. For example, if the aggregation is achieved by jointly source coding correlated samples, the state of a node in aggregation should be redefined to include the information of the number of collected *original* samples[1] as well as the time instants and locations of generation of these samples.

With this SMDP model, the objective of the decision problem becomes to 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 $\alpha > 0$ [3]. That is, for a given policy $\pi = \{\mathbf{d}_1, \mathbf{d}_2, ...\}$ and initial state s, the expected reward is defined as

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

where $s_0 = s,\ t_0 = 0$ and $t_0,\ t_1, ...$ represent the times of successive decision epochs. By defining

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

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

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

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 [4].

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

This assumption is reasonable under almost all practical scenarios. Condition (1) implies that the expected reward under any policy is bounded, i.e., $v^*(s) < \infty$ for all $s \in S'$ [4]. This is reasonable 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 bounded. 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.

B. The Optimality Equations and Solutions

Under Assumption 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

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

$\forall s \in S'$ and $v(\Delta) = v(\Delta)$ 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, $E[v(j)e^{-\alpha\tau}|s]\}$, 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 parameter α.

It can be shown that

- Result 1: *optimal reward* $\mathbf{v}^* \geq 0$ *is the minimal solution of the optimality equations (4) and consequently,* $v^*(\Delta) = 0$*;*
- Result 2: *there exists an optimal stationary policy* $\boldsymbol{d}^\infty = \{\boldsymbol{d}, \boldsymbol{d}, ...\}$ *where the optimal decision rule* $\boldsymbol{d}$ *is given by*

$$d(s) = \arg\max_{a\in A_s}\{g(s),\ \sum_{j\geq s} q^0_{sj}(\alpha)v^*(j)\} \qquad (5)$$

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

Result 1 directly follows similar procedures to the proofs of Theorem 7.1.3, 7.2.2 and 7.2.3 in [3] 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; Result 2 is the application of Theorem 3 (Chapter 3) in [4] on the SMDP model.

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, and what are the conditions for the existence of such a policy?
2) Without structured policies, can we approximate the value of $\mathbf{v}^*$ with a truncated (finite) state space, and

[1] The *original* sample is defined as the sample with the information of a single sensor reading, i.e. the raw sample.

[2] Although the first actual available transmission epoch within a decision horizon is not necessary to be the instant that $s = 1$ (as shown in Figure 1), we can still treat the instant of $s = 1$, i.e., the beginning of an aggregation operation, as the initial decision epoch with action $a = 0$. Thus $s = 1$ is the initial state. This setting would not change the optimal reward or policy as long as aggregation has a benefit, i.e., $\exists s \in S', g(s) > 0$.

is there any efficient on-line algorithm to obtain the solution for such finite-state approximation?

The answers to the questions will be presented in the following two sections, respectively. Due to space constraints, We skip the proofs for all the results listed below and refer the interested reader to [12].

III. Existence of An Optimal Control-limit Policy

In this section, we discuss the structured solution of the optimal policy in (5). Such kind of solution is attractive for implementation in energy and/or computation limited sensor networks as it significantly reduces the search effort for the optimal policy in the state-action space once we know there 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 $s \in S'$ [3], i.e.,

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

where $s^* \in S'$ is a *control limit*. Thus, the search for the optimal policy is reduced to simply find s^*.

A. Sufficient Conditions for Optimal Control-Limit Policies

By observing that the state evolution of the node is nondecreasing with time, i.e., the number of samples collected during one aggregation operation is nondecreasing, we provide in Theorem 1 a sufficient condition for the existence of an optimal control-limit policy under Assumption 2, which is primarily based on showing the optimality of one-stage-lookahead (1-sla) stopping rule [4].

Theorem 1: *Under Assumption 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 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 total discounted reward (for initial state $s = 1$) is

$$v^*(1) = \sum_{j \geq s^*} m_{1j}(\alpha) g(j), \tag{9}$$

where $\boldsymbol{m}(\alpha) \triangleq [m_{ij}(\alpha)] = (\boldsymbol{I} - \boldsymbol{M}_d^{S'})^{-1}$ with $\boldsymbol{M}_d^{S'} \triangleq [M_{ij}]$ and

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

In Theorem 1, the optimality of 1-sla stopping rule tells us that once the reward by stopping at current stage exceeds the expected discounted reward by continuing one more stage, it is optimal to stop at the current stage. However, the sufficient condition for the existence of an optimal control-limit policy listed in Theorem 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 stopping 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 1 *almost always* holds. We formally state the above conditions in the following Corollary.

Corollary 1: *Under Assumption 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 (11) holds for all states $i \geq s,\ i, s \in S'$ once (7) is satisfied at certain s,*

$$\sum_{j \geq k} Q^0_{ij}(\tau) \geq \sum_{j \geq k} Q^0_{i+1,j+1}(\tau),\ \forall k \geq i,\ \forall \tau \geq 0. \tag{11}$$

Then, there exists an optimal control-limit policy.

As a special case of Corollary 1, if the dependency of $Q^0_{ij}(\tau)$ on the current state i can be further relaxed, i.e., the inter-arrival time of consecutive decision epochs and the increment of the natural process are independent of the current state, (11) 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, and for a linear aggregation gain $g(s) = s - 1$, the closed-form expression for the control limit s^* in (8) can be readily obtained as

$$s^* = \left\lceil \frac{E[Xe^{-\alpha \delta W}]}{1 - E[e^{-\alpha \delta W}]} + 1 \right\rceil \tag{12}$$

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.

B. Comparison to Aggregation Policies in the Literature

From (12), we can see some similarities and differences between the control-limit policy and the previously proposed fixed degree of aggregation (FIX) and on-demand (OD) schemes [5]. 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 (12) 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 (12) to emphasize the impact of delay, then (12) is reduced to a special (extreme) case such that $s^* = 1$. It 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 policy 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 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 will first introduce a finite-state approximation model for the original problem and verify its convergence to the original countable state space model. Then, two on-line algorithms are provided to solve the finite-state approximation model.

A. A Finite-State Approximation Model and its Convergence

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

$$v_N(s) = \max\{g(s) + v_N(\Delta), \sum_{j \geq s} q^0_{sj}(\alpha) v_N(j)\} \qquad (13)$$

for $s \in S'_N$ and $v_N(\Delta) = v_N(\Delta)$. Let $\mathbf{v}^*_N \geq 0$ be the minimal solution of the optimality equation (13). Consequently $v^*_N(\Delta) = 0$. To verify the (point-wise) convergence of the finite-state approximation model (13) to (4), we state the following Lemmas and Theorem 2 without proofs.

Lemma 1: *$v^*_N(s)$ monotonically increases with N, $\forall s \in S'$.*

Lemma 2: *$v^*_N(s) \leq v^*(s)$, $\forall s \in S'$ and $\forall N > 0$.*

Theorem 2: *$\lim_{N \to \infty} v^*_N(s) = v^*(s), \forall s \in S'$.*

Theorem 2 states that for each $s \in S'$, $v^*_N(s)$ is expected to be close to the optimal $v^*(s)$ under a sufficiently large value of N (which depends on s). 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 expected optimal reward $v^*_N(1)$ of one aggregation operation will be very close to $v^*(1)$.

In finite-state approximations, 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 [3] can be used to solve (13). However, in practice, $q^0_{sj}(\alpha), \forall s, j \in S'_N$ are 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. In the following, we provide two kinds of learning algorithms for solving the finite-state approximation model.

TABLE I

Adaptive Real-time Dynamic Programming (ARTDP) Algorithm.

```
1   Set $k = 0$
2   Initialize counts $\omega(i, j)$, $\eta(i)$ and $\hat{q}^0_{ij}(\alpha)$ 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 $v_{k+1}(s_k) = \max\{g(s_k), \sum_{N \geq j \geq s_k} \hat{q}^0_{s_k j}(\alpha) v_k(j)\}$;
7     Rate $r_{s_k}(0) = \sum_{N \geq j \geq s_k} \hat{q}^0_{s_k j}(\alpha) v_k(j)$ and $r_{s_k}(1) = g(s_k)$;
8     Randomly choose action $a \in \{0, 1\}$ according to
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      $\eta(s_k) + +$;
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 $\hat{q}^0_{s_k j}(\alpha) = \frac{\omega(s_k, j)}{\eta(s_k)}, \forall N \geq j \geq s_k$;
16      else $a = 1$, $s_{k+1} = \Delta$;
17    $k + +$. }
18  }
```

B. Algorithm I: Adaptive Real-time Dynamic Programming (ARTDP)

Adaptive real-time dynamic programming (ARTDP) (see [13], [14]) 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., $q^0_{ij}(\alpha)$) might be updated (lines 12-16).

A key step in ARTDP is to estimate the value of $q^0_{ij}(\alpha)$ 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

$$\hat{q}^0_{ij}(\alpha) \approx \sum_{l=0}^{\infty} \frac{\eta(i, j, l)}{\eta(i)} e^{-\alpha \delta W_l}. \qquad (14)$$

Let $\omega(i, j) \triangleq \sum_{l=0}^{\infty} \eta(i, j, l) e^{-\alpha \delta W_l}$, the estimation of $\hat{q}^0_{ij}(\alpha)$ can be improved by updating $\omega(i, j)$ and $\eta(i)$ at each state transition as shown in lines 12-16 of Table I.

In ARTDP, the rating of actions and exploration procedure (lines 7-9) follow the description in [14]. 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

TABLE II

REAL-TIME Q-LEARNING (RTQ) ALGORITHM.

```
1    **Set** $k = 0$
2    Initialize Q-value $Q_k(s,a)$ for $s \in S'_N, a \in \{0,1\}$
              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
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 $Q_{k+1}(s_k, 0) = (1 - \alpha_k)Q_k(s_k, 0) +$
13                  $\alpha_k[e^{-\alpha \delta W_{k+1}} \max_{b \in \{0,1\}} Q_k(s_{k+1}, b)]$
14       **if** $s_{k+1} > N$, $a = 1$, $s_{k+1} = \Delta$;
15     $k++$. }
16   }
```

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 given by

$$d_N^*(s) = \arg\max_{a \in \{0,1\}} \{g(s), \sum_{N \geq j \geq s} \hat{q}_{sj}^0(\alpha) v_N^*(j)\} \tag{15}$$

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

C. Algorithm II: Real-time Q-learning (RTQ)

Real-time Q-learning (RTQ) [13] 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 $v_N^*(s) = \max_{a \in \{0,1\}} [Q_*^N(s,a)], s \in S'$. The optimizing Q-learning rule is given in Table II (line 10 and lines 12-13).

In RTQ, the exploration procedure (lines 7-8) is the same as the one in ARTDP. In kth Q-value update (lines 9-13), α_k is defined as the *learning rate*, which is generally state and action dependent. To ensure the convergence of RTQ, Tsitsiklis has shown in [15] 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 [14]. 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

$$d_N^*(s) = \arg\max_{a \in \{0,1\}} \{Q_*^N(s,a)\} \tag{16}$$

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

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 1, which we call the CNTRL scheme, and its special case in (12) 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 1 and Corollary 1, 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 $\overline{\delta W}_s = \delta W_0 e^{-A(s-1)} + \delta W_{min}$, 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 [3]) and $A > 0$ 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[3] $1/\overline{\delta W}_s$. 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 have a Poisson distribution with a rate $\lambda_s = \lambda_0 e^{-B(s-1)}$, where λ_0 is a constant which represents the rate of sample arrival at state $s = 1$ and $B > 0$ is a constant to control the degree of state-dependency of the natural process. By adjusting parameters A and B, 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 linear programming (LP) solution as a reference, which uses the estimated $\hat{q}_{ij}^0(\alpha)$ (as described in ARTDP algorithm). With a proper randomized action selection and a large number of iterations in ARTDP, $\hat{q}_{ij}^0(\alpha)$ provides a good approximation of $q_{ij}^0(\alpha)$. Thus the solution of LP is expected to be close to $\mathbf{v}_N^*$. We also distinguish the terms *calculated value* and *actual value* of the reward at state $s \in S'_N$, where the *calculated value* is the value of the reward obtained from the LP solution or iterative calculation in learning algorithms and the *actual value* of the reward is obtained from the measured statistics of actual aggregation operations. When the truncation effect of the state

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

space at state s is non-negligible, i.e., N is not large enough for state s, the calculated value is different from the actual value, as expected. 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.

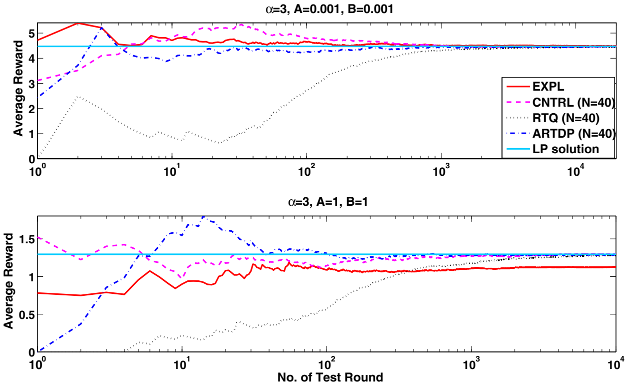

Fig. 2. Convergence of the values of the reward for initial state $s = 1$ in EXPL, CNTRL, ARTDP and RTQ under different traffic patterns: $A = 0.001, B = 0.001$, i.e., a low degree of state-dependency (upper) and $A = 1, B = 1$, i.e., a high degree of state-dependency (bottom); delay discount factor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

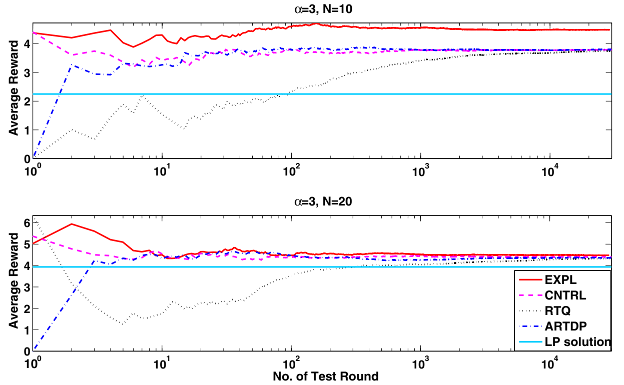

Fig. 3. Convergence of the values of the reward for initial state $s = 1$ in EXPL, CNTRL, ARTDP and RTQ under different degrees of finite-state approximation: $N = 10$ (upper) and $N = 20$ (bottom); delay discount factor $\alpha = 3$; traffic pattern $A = 0.001, B = 0.001$, i.e., a low degree of state-dependency.

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, $A = 0.001, B = 0.001$, 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 $s^* = 10$. In the bottom plot, $A = 1$ and $B = 1$ 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.

Figure 3 shows the effect of finite-state approximation on the performance of the schemes. We consider $A = 0.001, B = 0.001$ in which the EXPL scheme provides a value of 4.48 (initial state $s = 1$) and a 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 a value (≈ 3.78) 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 (13) 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 $s^* = 10$. 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 have a control limit $s^* = 8$. Further improvement at $N = 40$ for finite-state approximation has been shown in Figure 2. 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.

B. Evaluation in Distributed Data Aggregation Scenario

We provide a further evaluation of the proposed schemes as well as the existing schemes in the literature (i.e. the OD and FIX schemes) in a practical distributed data aggregation application in which each sensor is expected to track the maximum value of an underlying time-varying phenomenon in a sensing field. There are 25 sensor nodes randomly deployed in a two-dimensional sensing field. The phenomenon is modelled as a spatial-temporal correlated discrete Gauss-Markov process. Each node is equipped with an omnidirectional antenna and the expected communication range is $r_0 = 10\ m$. 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 Berkeley Motes

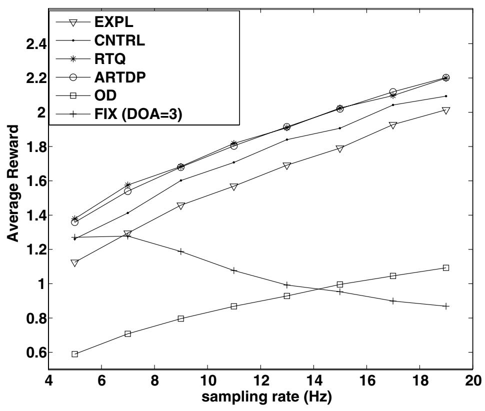

Fig. 4. Average rewards of EXPL, CNTRL, ARTDP, RTQ, OD and FIX in a distributed data aggregation; delay discount factor $\alpha = 8$, finite-state approximation $N = 10$.

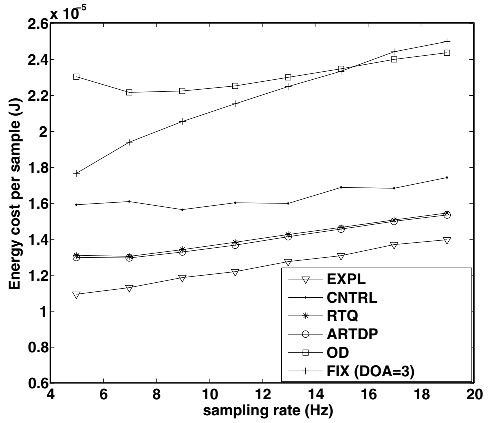

Fig. 6. Energy consumption (per sample) of EXPL, CNTRL, ARTDP, RTQ, OD and FIX in data aggregation; delay discount factor $\alpha = 8$, finite-state approximation $N = 10$.

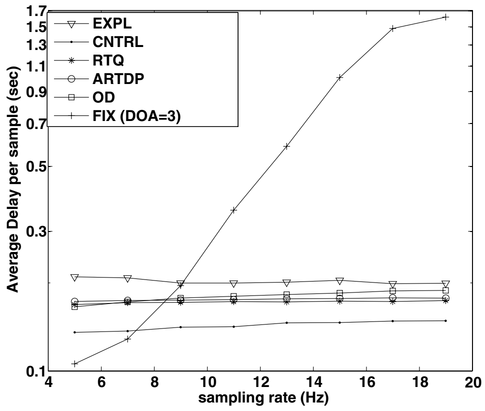

Fig. 5. Delay performance of EXPL, CNTRL, ARTDP, RTQ, OD and FIX in a distributed data aggregation; delay discount factor $\alpha = 8$, finite-state approximation $N = 10$. The y-axis is in logarithmic scale.

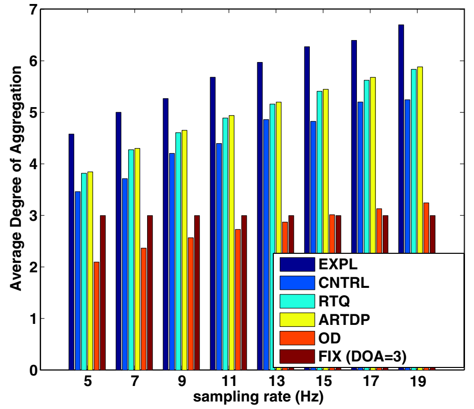

Fig. 7. Average degrees of aggregation (DOA) versus different sampling rates in EXPL, CNTRL, ARTDP, RTQ, OD and FIX in data aggregation; delay discount factor $\alpha = 8$, finite-state approximation $N = 10$.

MICA2 [16]. Each original sample is assumed to have $16\ bits$. We set the delay discount factor as $\alpha = 8$ and the degree of finite-state approximation as $N = 10$. The linear function $g(s) = s - 1$ is used as the nominal aggregation gain. Such linear function allows us to evaluate the EXPL scheme proposed in (12). In practice, other forms of the utility function $g(s)$ may be used to represent aggregation gains of interest to designers. The CNTRL and the learning algorithms would still work under these kinds of utility functions. In the FIX scheme, for illustration, the degree of aggregation (DOA) is set to 3. The simulation will show that (see Figure 7, 8) there exists no universal value of DOA which is optimal under all scenarios and the optimal DOA should be adaptive to the local traffic.

Figure 4 shows the average reward (initial state $s = 1$) obtained by each scheme during aggregation operations. 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 1). All the proposed schemes in this paper have a significant gain over the previously proposed OD and FIX schemes. One might also notice that FIX with DOA $= 3$ shows a decreasing trend in reward with the increase of the sampling rate while others have an increasing trend. This is because FIX can not dynamically adjust its DOA ($= 3$) when the sampling rate increases, unlike the other schemes.

Figure 5 evaluates the average delay for collecting the time-varying maximal values of the field in each scheme. Notice that, as we did not consider any transmission loss and noise in reception, delay (i.e., tracking lag) provides a suitable metric for evaluating tracking performance [17]. OD, RTQ and ARTDP have a similar delay performance which is slightly higher than CNTRL and lower than EXPL. FIX shows the worst delay performance when sampling rate is higher than $9\ Hz$ as its fixed DOA can not help much in reducing network congestion in a high sampling rate scenario.

Energy costs for different schemes are compared in Figure 6. OD shows an overall highest energy cost as aggregation

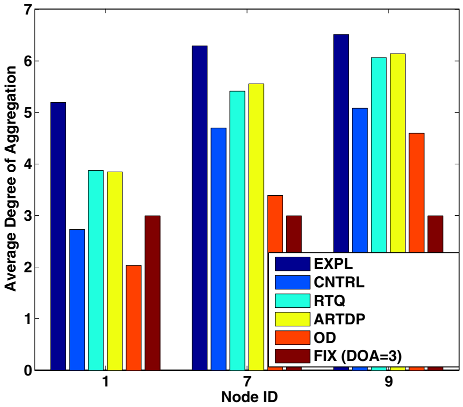

Fig. 8. Average degrees of aggregation (DOA) of EXPL, CNTRL, ARTDP, RTQ, OD and FIX at different nodes: Node 1 (node degree =3), Node 7 (node degree = 5) and Node 9 (node degree = 6); sampling rate is set as 11 Hz.

for energy saving is only opportunistic. EXPL shows the best energy saving performance among all schemes as it actually achieves a higher DOA than other schemes (see Figure 7), though this does not mean EXPL is optimal when aggregation delay is taken into consideration. Again, RTQ and ARTDP have similar performance in energy cost. From Figure 5 and 6, we can see a clear delay-energy trade-off in the proposed schemes as well as OD. Among them, RTQ and ARTDP achieve the best balance between delay and energy.

Figure 7 gives the average DOA, i.e., the number of samples collected per aggregation operation, in all schemes under different sampling rates. It is clear that the proposed schemes and OD can adaptively increase their DOAs as the sampling rate increases. On the other hand, Figure 8 shows the DOAs at different nodes under a given sampling rate (11 Hz), where 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-B, 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 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 was 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 were developed and the sufficient conditions for their optimality were found. Furthermore, we provided two on-line learning algorithms for the general case of the problem and investigated their performance under a tunable traffic model. ARTDP showed a better convergence speed than RTQ with a cost of computation complexity in learning the system model. The simulation on a practical distributed data aggregation scenario showed that ARTDP and RTQ achieved the best performance in balancing energy and delay costs, while the performance of control-limit type policies, especially the EXPL scheme in (12), is close to that of learning algorithms, but with a significantly lower implementation complexity. All the proposed schemes outperformed the traditional schemes, i.e., the fixed degree of aggregation (FIX) scheme and the on-demand (OD) scheme.

Acknowledgement

This work was funded in part by National Science Foundation grants No. CNS-0322956 and CNS-0546402.

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] R. Cristescu and M. Vetterli, "On the optimal density for real-time data gathering of spatio-temporal processes in sensor networks," in *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks, IPSN 2005*. Los Angeles, CA: IEEE, Apr. 2005, pp. 159–164.

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

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

[5] 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, May 2004.

[6] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *MOBICOM*, Boston, MA, Aug. 2000, pp. 56–67.

[7] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: A tiny AGgregation service for ad-hoc sensor networks," in *OSDI*, Boston, MA, Dec. 2002.

[8] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *MOBICOM*, Seattle, WA, Aug. 1999, pp. 174–185.

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

[10] F. Hu, X. Cao, and C. May, "Optimized scheduling for data aggregation in wireless sensor networks," in *ITCC (2)*. Las Vegas, NE: IEEE Computer Society, Apr. 2005, pp. 557–561.

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

[12] Z. Ye, A. A. Abouzeid, and J. Ai, *Optimal Policies for Distributed Data Aggregation in Wireless Sensor Networks*. Troy, NY: Tech. Report, Department of Electrical, System and Computer Engineering, RPI, 2006.

[13] 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.

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

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

[16] Crossbow, *MPR/MIB Users Manual Rev. A, Doc. 7430-0021-07*. San Jose, CA: Crossbow Technology, Inc., 2005.

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