On the Interaction between Data Aggregation and Topology Control in Wireless Sensor Networks

Vijay Erramilli Ibrahim Matta Azer Bestavros
Computer Science Department
Boston University
{evijay, matta, best}@cs.bu.edu
June 18, 2004
Technical Report BUCS-2004-024

***Abstract*— Wireless sensor networks are characterized by limited energy resources. To conserve energy, application-specific aggregation (fusion) of data reports from multiple sensors can be beneficial in reducing the amount of data flowing over the network. Furthermore, controlling the topology by scheduling the activity of nodes between active and sleep modes has often been used to uniformly distribute the energy consumption among all nodes by de-synchronizing their activities. We present an integrated analytical model to study the joint performance of in-network aggregation and topology control. We define performance metrics that capture the tradeoffs among delay, energy, and fidelity of the aggregation. Our results indicate that to achieve high fidelity levels under medium to high event reporting load, shorter and fatter aggregation/routing trees (toward the sink) offer the best delay-energy tradeoff as long as topology control is well coordinated with routing.**

I. Introduction

Motivation: A sensor network consists of one or more "sinks" which subscribe to specific events by expressing interest in the form of queries. The sensors in the network act as "sources" which detect events and push relevant data to the appropriate subscriber sinks. For example, there may be a sink that is interested in a particular spatio-temporal phenomenon, *e.g.* is there any activity in any of the two conference rooms during lunch hour, noon–1pm? During times of interest, if sensors in the corresponding spatial portion of the network detect the event in question, they act as sources and push data corresponding to that event toward the subscribing sink. Wireless sensor networks are expected to operate in highly dynamic environments under severe energy constraints. However since many sensor nodes (sources) in a certain area/neighborhood often detect common phenomenon, there is likely to be some redundancy in the data which various sources communicate to a particular sink. This redundancy, often referred to as *over-sampling*, is specially prevalent in large-scale (dense) sensor networks. Data aggregation or fusion [1], [2] has been proposed as an in-network filtering and processing technique to help eliminate redundancy and conserve the scarce energy resources. The idea is to combine, in an application-specific manner, the data signals coming from different sources en-route, thus minimizing the number of transmissions.

Another widely employed technique for saving energy in wireless sensor networks is to routinely place nodes in a low energy "sleep" mode during idle periods [3]. So during idle stages, instead of expending valuable energy listening, a node switches itself off. This in effect controls the actual topology of the network by the connectivities among those nodes currently awake.

Our Contribution: It is not hard to discern that a tradeoff exists between energy and performance of the network. To the best of our knowledge, no analytical model has been developed to investigate this tradeoff in the presence of *both* data aggregation and topology control (through the sleep/active dynamics of sensor nodes). In this paper we present such integrated analytical model and illustrate its generality in capturing a whole range of data aggregation behavior and how it is affected by sleep/active dynamics and the resulting levels of channel contention. We define performance metrics to evaluate the conflicting goals of minimizing energy consumption and decreasing end-to-end response times. One performance metric we define is the "fidelity" of aggregation, which captures the quality of the aggregated signal based on the number of sensor nodes which had contributed to it. Our results support the following main conclusions:

- Under medium to high event reporting load, to achieve full fidelity in aggregation, routing/aggregation trees with higher node degree (*i.e.* trees that are shorter and fatter) offer a better delay-energy tradeoff as the savings in energy offset the

This work was supported in part by NSF grants ANI-0095988, ANI-9986397, EIA-0202067, and ITR ANI-0205294.

increase in delay that may be caused by increased contention among sibling sensor nodes.

- Topology control (through active/sleep schedules) is often detrimental to the sensor network in terms of increased delays, if in-network aggregation is employed and high aggregation fidelity is desired. Hence, in the presence of in-network aggregation, careful coordination between routing and topology control should be exercised.

A complete list of our observations/findings can be found in Section V-B.

Paper Organization: The remainder of the paper is organized as follows. Section II reviews previous related work. In Section III we describe the network system and assumptions we make for the construction of our analytical model. In Section IV we describe our Markov model in detail and develop a complete network model that accounts for in-network data aggregation, channel contention, and topology control through sleep/active dynamics. In Section V we present our results, and Section VI concludes the paper.

II. Related Work

The work of Krishnamachari $et\ al.$ [4] was the first to deal with the performance issues of sensor data aggregation. They show that the problem of constructing "optimal" data-aggregation trees rooted toward the sink is NP-hard. They also point out the important delay-energy tradeoff, in the presence of non-trivial (time-consuming) aggregation. While in this paper we assume trivial aggregation, our model can be easily extended to relax this assumption. Nevertheless, the delay-energy tradeoff manifests itself in our results as aggregation from a larger number of sensor nodes further reduces energy consumption but at the expense of increased delays.

Boulis $et\ al.$ [5] study the energy-accuracy tradeoff under two different types of aggregation: "snapshot" aggregation that is performed once, and "periodic" aggregation that is regularly performed. Intanagonwiwat $et\ al.$ [6] study the effect of network density on constructing energy-efficient aggregation trees. Scheduling nodes for "sleeping" [3] during their idle periods has also been proposed to alleviate energy consumption. While aggregation has been studied extensively as a network-level problem and scheduling nodes for "sleeping" has been studied as a MAC-layer problem, there has not been a study of the joint problem of data aggregation and topology control through sleep-active dynamics of nodes. In this paper we do just that by following a methodology similar to that of [7] on a model we develop that integrates aspects of *both* aggregation and topology control.

III. System Description and Assumptions

A. System Description

We define our system to consist of three components: *Nodes*, *Energy Model*, and *Channel Access Model*.

1) Nodes: We consider wireless sensor networks with stationary nodes of two types: *normal* nodes and *aggregating* nodes.

- *Normal* nodes sense the desired event and forward the data toward the sink. These nodes can only transmit. Borrowing from the model in [7], such normal node has two major operational states, *active* (A) and *sleep* (S). The number of time slots spent by the node in state A is a geometrically distributed random variable with parameter p. The number of slots spent in state S is also geometrically distributed with parameter q.
  The active state A is further divided into a main phase R and (possibly) a phase N. During the R phase the node can sense and transmit data. The sensed data is stored in a local buffer, waiting for transmission. If the duration of time during which the node stays awake runs out, and if the buffer is not empty, then the node enters a "closing" phase N, and this duration is extended till all the data in the buffer are transmitted, and then the node enters the sleep state S. Once the node enters state S and the sleep time expires, the node returns to the active state A.
- *Aggregating* nodes (henceforth called *aggregators* for short) perform the function of aggregating and forwarding data to the sink. A level-1 *aggregator* receives data from one or more *normal* nodes, performs an aggregation function (*e.g.* sum, average), and then forwards the aggregate packet. At higher levels, aggregators repeatedly aggregate data in this manner all the way along a routing tree toward the sink. Aggregators can transmit, receive and perform an aggregation function, however they cannot sense. Aggregators are also characterized by two operational states, *active* and *sleep*. The *active* state is further divided into *receive* mode and *transmit* mode. In the former mode, the aggregator waits for child nodes to send their data, and in the latter mode, the aggregator aggregates the data received from its child nodes and forwards the aggregated packet. An aggregator can be in either of these modes, but not both.

The time spent in either the *active* or *sleep* mode is geometrically distributed with parameters p and q, respectively, similar to the behavior of *normal* nodes.

For both types of nodes, while in the active state, nodes can perform their designated functions like sense, transmit, receive, etc. On the other hand, while in the sleep state, a node cannot take part in any network activity. Thus the (effective) topology of the network keeps on constantly changing as nodes enter/exit the sleep state.

2) Energy Model: The energy consumption for a node is calculated using the quantities defined in Table I.

$E^{(elec)}$	Energy expended by the transreceiver electronics (receiving and transmitting packets)
$E^{(proc)}$	Energy expended in normal processing
$E^{(amp)}$	Energy expended by the amplifier
E_t	Energy expended in switching from sleep to active phase
E_s	Energy expended in the sleep mode

TABLE I
PARAMETERS OF ENERGY MODEL

3) Channel Access: We use the access model proposed in [8] and later adapted by [7]. Consider a one-hop transmission between nodes l and m. The transmission is successful if:

- the distance between l and m is not greater than r; and

$$d_{l,m} \leq r \quad (1)$$

- for every other node, n, simultaneously transmitting

$$d_{n,m} > r \quad (2)$$

where r is the reception range of a node. In other words, the model accounts for channel *contention*, and does not model *collisions*.

4) Performance Parameters and Metrics: Our main objective in this paper is to study aggregation under various conditions. To that end, we abstract and model the following behavior: an aggregation node stays awake for a predetermined amount of time during which it receives one data unit from as many children as it can before it aggregates and forwards the aggregate packet to its parent along the aggregation tree toward the sink. We define the following metrics to help us characterize and study aggregation.

- **Round:** A round defines the time during which the aggregator stays awake. If such a node receives more than one data unit from a child during a round, it merely assimilates it in the present round; it does not store it for the next round but rather consumes it as more recent information from that child.
- **Unique Packets:** We define unique packets received by an aggregator during a round as packets received from each individual child of that aggregator. An ideal situation from the aggregator's perspective is that it receives at least one packet from each child during a round.
- **Aggregation Fidelity:** We define the fidelity of the aggregation as the ratio of the number of children which successfully transmit unique packets in a round over the total number of children of an aggregator. Ideally this ratio should be one.

B. Assumptions

Here we summarize our assumptions on the topology, routing and MAC protocols. We assume stationary sensor nodes which have a common maximum radio range r and are equipped with omni-directional antennas. The buffers at the sensors are assumed to be of infinite capacity (hence no losses in the network) and are modeled as FIFO queues. The information sensed by the sources is organized into data units of fixed size, and sent in fixed-time slots. A sensor cannot simultaneously transmit and receive. For aggregators, we assume that such a node knows the number of its children. We assume that the aggregator node aggregates all the data it receives into one packet which it then forwards to its parent along the aggregation tree toward the sink. We assume the aggregation process itself to be trivial, and hence does not add to the processing time.

Routing is performed by following an aggregation tree whose leaves are normal nodes. See Figure 1. Aggregators constitute the internal nodes of the tree and the sink is its root. While constructing optimal aggregation trees is in itself an open problem [4], we assume that every node knows apriori which node it has to route to, *i.e.* each node knows (and is within communication range of) its parent along the aggregation tree. We assume that sibling nodes are not within communication range from each other.

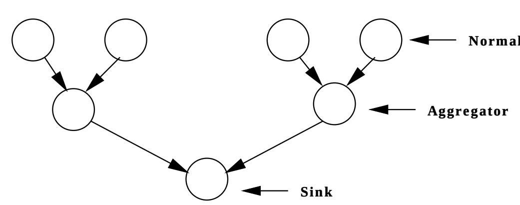

Fig. 1. **Aggregation tree rooted at the sink**

The MAC layer is assumed to be based on a

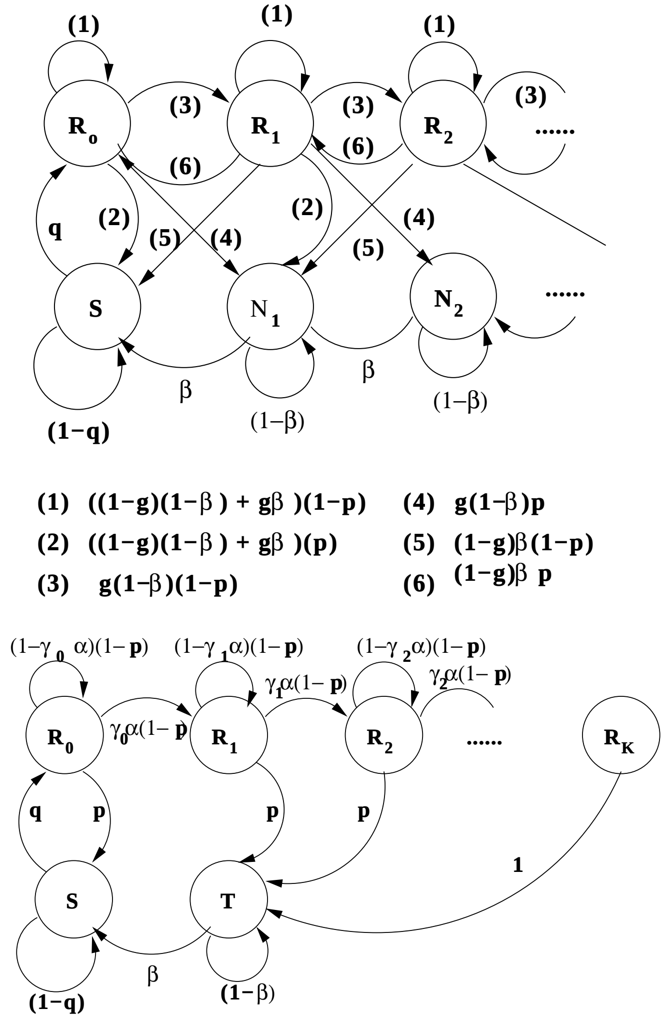

Fig. 2. **(top) DTMC of Normal node, (bottom) DTMC of Aggregator node**

contention-avoidance scheme (*e.g.* CSMA/CA). However one can easily extend our model to include TDMA-type protocols as well. The wireless channel is assumed to be error-free.

IV. Sensor Model

A. Node Model

We start by studying the behavior of a single node (of each type) by developing a discrete-time Markov chain (DTMC) model, in which the time is slotted according to a data-unit transmission time, that is the time needed to transmit a data unit including the overhead required by the MAC layer. The salient features of the models for different types of nodes are as follows:

- *Normal* Nodes: We adopt the model of [7] which we briefly describe here for completeness. The states of the Markov chain (shown in Figure 2(top) along with the transition probabilities) are defined by the phase the sensor could be in during the current time slot (namely S, R_i or N_i) and the number of data units i in the buffer, which could range from 0 to ∞. Let **P** be the transition matrix, whose element P(s_o, s_d) denotes the probability that the chain moves in one time slot from the origin state s_o to the destination state s_d. In deriving such transition probabilities, the following dynamics are taken into account:
  - The active periods are controlled by the input parameter p. Smaller values of p mean that the node remains active for a longer time.
  - The sleep periods are controlled by the input parameter q. Smaller values of q mean that the node remains in the sleep state for a longer time.
  - During phase R only, new data is generated (sensed) at a rate g according to a Poisson distribution.
  - During phases R and N only, a data unit is successfully transmitted in a time slot with probability β. As we compute it later, β accounts for contention as well as the fact that the next-hop (parent along the aggregation tree) might be asleep and thus can not receive.
- *Aggregator* Nodes: The states of the Markov chain (shown in Figure 2(bottom) along with the transition probabilities) are defined by the phase the sensor could be in during the current time slot (namely, sleep S, receive R_i or transmit T) and the number of *unique* data units i in the buffer, which accounts for packets successfully transmitted by each child of the aggregator.

  The behavior of the aggregator node can be defined by (i) p, which determines the length of the active period during which an aggregator receives one or more unique data units from its children; (ii) q, which determines the length of the sleep period after which an aggregator goes back to wait for new data from its children; (iii) β, which denotes the probability of the aggregator successfully transmitting the aggregated packet; (iv) α, which denotes the probability of the aggregator successfully receiving a packet sent by one of its children; and (v) γ_i, a state-dependent probability of receiving a new *unique* packet, *i.e.* a fresh packet from one of the children. Thus, γ_i is defined by $\frac{K-i}{K}$ where i is the number of unique packets received so far from the aggregator's children and K is the number of child nodes. Hence γ_i takes a value between 1 and 0.

  One can clearly note the tradeoff between the aggregation *fidelity* achieved and energy consumed—The more an aggregator remains in the active state, the greater is the chance to achieve a fidelity value of one. However the more time the node spends in the active state, the more the node expends energy, not to mention increased delay.

We note that our model is fairly general and can be used to model and study different types of behavior. For example, we can study the effect of varying the parameters p and q on the steady-state probability of being in the state with aggregation fidelity value of one.[1] While p, q and K are input parameters, both α and β need to be estimated through a network model that considers the interactions between neighboring nodes, as we later show in this section.

Solution of DTMC: Once we have a node model, we solve the corresponding DTMC using the Matrix Geometric technique [9] to obtain the stationary distributions $\pi = \{\pi_s\}$ where s generically denotes the state of the model. Once we obtain π, we derive the following metrics:

- The overall probabilities of nodes spending their time in various phases
- The average number of data units (sensed and) generated in a slot by a *normal* node:

$$\lambda_n = \sum_{i=0}^{\infty} \pi_{R_i}\ g \tag{3}$$

- The throughput T_n (T_a), defined as the average number of data units forwarded in a time slot by a normal (aggregator) node:

$$T_n = \sum_{i=1}^{\infty} (\pi_{R_i} + \pi_{N_i})\beta \tag{4}$$

$$T_a = \pi_T\ \beta \tag{5}$$

- The average buffer occupancy $\bar{B}_n$ ($\bar{B}_a$) of a normal (aggregator) node:

$$\bar{B}_n = \sum_{i=1}^{\infty} (\pi_{R_i} + \pi_{N_i}) i \tag{6}$$

$$\bar{B}_a = \sum_{i=1}^{K} i\pi_{R_i} + \pi_T \tag{7}$$

[1]Note that while the steady-state probability of being in a particular state is not the same as the probability of reaching the state of interest, it gives us useful insights.

B. Network Model

We use an open network of queues to incorporate our node models within a network setting. We regard each queue as corresponding to the buffer of a sensor. The external arrival rate corresponds to the data unit generation (sensing) rate at the normal sensors, which constitute the leaves of the aggregation tree rooted at the sink (cf. Fig. 1). Given the aggregation tree topology, the traffic from normal (leaf) nodes gets routed all the way to the sink. Since at steady-state, the input flow rate equals the output flow rate, the throughput into a node, denoted by α in the previous node model, is easily computed from the throughput out of its children nodes, which are given by equations (4) or (5).

C. Interference Model

Following the model of [7], the purpose of this model is to compute for each node the parameter β.

For each sensor node, we define a set I as the set of all nodes whose transmission range covers the next-hop of that node (i.e. its parent in the aggregation tree). We first use equations (1) and (2) to determine which nodes interfere with the transmission of a particular node to its parent. Then the average probability t_I that a node in set I is ready to transmit a packet is given by:

$$t_I = \frac{1}{\mid I \mid} \sum_{n \in I} (\sum_{i=1}^{\infty} \pi^n_{N_i} + \sum_{i=1}^{\infty} \pi^n_{R_i} - \pi^n_{R_0}) + \sum_{a \in I} (\pi^a_T) \tag{8}$$

where n and a represent a normal node and an aggregator, respectively.

We then consider that a node will be able to transmit only if it gets the control of the channel before any other node in set I. Assuming that all nodes in this set are equally likely to seize the channel, we can consider their probability to be ready to transmit as being independent and derive the following equation for β [7]:

$$\beta = \sum_{k=0}^{|I|} \frac{1}{k+1} \binom{\mid I \mid}{k} t_I^k (1 - t_I)^{|I|-k} (1 - (\pi_S + \pi_T)) \tag{9}$$

where π_S refers to the probability that the next-hop (parent) is sleeping, and π_T refers to the probability that the next-hop is transmitting (hence will not be able to receive).

D. Complete Model: Fixed Point Approximation

Our overall solution involves all three components we just described, namely (i) sensor/node model; (ii) network model; and (iii) interference model. We use a Fixed

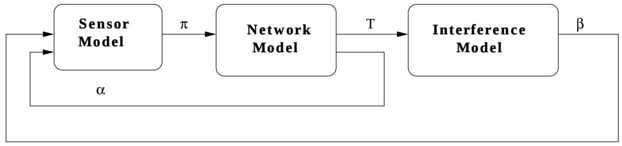

Fig. 3. **Fixed Point Approximation model**

Point approximation (FPA) method, in which all the above three sub-models interact by exchanging various parameters along a closed-loop till a final equilibrium of parameters is reached. The FPA process is illustrated in Figure 3.

The process starts with the solution of the DTMCs of individual sensor nodes in the network, from which we obtain stationary probabilities π's. We run the network model next to obtain the throughput out of sensor nodes T_n's and T_a's, from which we obtain the throughput into sensor nodes α's. We then use the interference model to estimate the corresponding β values, which are fed back, together with the α's, into the sensor model, thereby closing the loop.

We use as stopping criterion the relative error of throughput at the sink for two successive estimates. For the results in this paper, we use error ratio of less than 10^{-4}, which resulted in an average of 15 iterations to converge.

V. RESULTS

We consider a network with sources modeled as *normal* nodes and arranged as leaves in an aggregation tree, and all the intermediate nodes in that tree act as *aggregators*. The general requirements of such a network would be to sustain a high aggregation *fidelity* value, while at the same time deliver data with low delays and maintain a fairly high network lifetime. Given that a tradeoff exists between energy, fidelity and delay, the following questions can be asked.

- What are the tradeoffs involved in trying to achieve a high fidelity value (to be more specific, achieving a fidelity value of one)?
- What role do sleep-active dynamics used for topology control purposes play?
- How does network density (manifested by the degree of aggregator nodes) affect metrics of our interest?
- If we relax the high fidelity requirement, how would the various performance metrics change?

To answer these questions, we design two sets of experiments: the first set attempts to answer the first three questions; while the second set attempts to answer the last question.

For both sets of experiments, we use a common topology setup. We have a base tree topology of 61 nodes (including the sink). We construct various trees (and the corresponding network of queues) with increasing average degrees of intermediate (aggregator) nodes. So an aggregation tree with average degree of two is deemed thin and long, whereas a tree with average degree of six is fat and short. This enables us to study the effect of network density on the performance metrics of interest.

A. Performance Measures

The main metrics which we study are the average network delay (in slots), the average energy expenditure (in joules) per slot, and a fidelity-energy index (ratio) which captures the gain in aggregation fidelity per consumed energy.

(1) Average Network Delay: We calculate delay by applying Little's law to the whole network as follows:

$$\bar{D} = \frac{\sum_{k=1}^{M} \bar{B}_k}{C} \tag{10}$$

where M is the total number of nodes in the nework, C refers to the network capacity which is the total arrival rate of data units at the sink, and $\bar{B}_k$ is the average buffer size at node k which is calculated using equations (6) or (7). Thus $\bar{D}$ represents the average number of time slots to deliver one data unit to the sink.

(2) Energy Consumption per Slot:

To calculate the energy consumption per slot for a node, we calculate the consumption at the different operational states of the node. For a normal node, the energy expended on packet processing is given by:

$$\pi_S \; E_s + (\sum_{i=1}^{\infty} \pi_{N_i} + \sum_{i=1}^{\infty} \pi_{R_i}) E^{(proc)} \tag{11}$$

For an aggregator node, it is given by:

$$\pi_S \; E_s + (\pi_T + \sum_{i=1}^{K} \pi_{R_i}) E^{(proc)} \tag{12}$$

where $E^{(proc)}$ and E_s are defined in Table I.

The energy expended by a node in transmitting and receiving data as well as switching from sleep to active is given by:

$$T(d^2 \; E^{amp} + (\pi_S) q E_t) \tag{13}$$

where T is the throughput out of the node, $d^2 \; E^{amp}$ is the energy expended to transmit data over distance d to the next-hop node (parent in the aggregation tree), and E^{amp} and E_t are defined in Table I.

By summing up all the above energy costs, we obtain the total energy consumption per slot per node. We denote by $\bar{E}$, the energy consumption per slot averaged over all nodes.

(3) Fidelity-Energy Index:

This index is defined as:

$$\frac{\bar{\pi}_{R_K}}{\bar{E}} \tag{14}$$

where $\bar{\pi}_{R_K}$ is the steady-state probability of reaching full aggregation fidelity (averaged over the whole network). Higher values indicate that high fidelity in the aggregation is achieved at low energy consumption per slot.

B. General Observations

Under full fidelity in aggregation, we make the following main observations:

- Without topology control through active/sleep schedules, routing trees with higher node degree save energy at the expense of increased delays under medium to high event reporting loads.
- Under low load, routing trees with lower node degree may offer a better delay-energy tradeoff since full aggregation over less sensors can be achieved sooner.
- Under higher load, routing trees with higher node degree may offer a better delay-energy tradeoff as the savings in energy offset the increase in delay that may be caused by increased contention among sibling nodes.
- More aggressive topology control resulting in much fewer active (awake) nodes is more detrimental to aggregation trees of higher node degree since both delay and energy cost may increase as full aggregation over more sensors becomes harder.
- Under less aggressive topology control, as event reporting load increases, the overall delay increases due to increased contention among sibling nodes, but then the overall delay decreases as the decrease in aggregation delays offsets the increase in contention delays.

Under partial (lower) fidelity in aggregation, we make the following main observation:

- The increase in delay due to topology control (through active/sleep schedules) may offset the savings in energy from aggregation. Hence, in the presence of in-network aggregation, careful coordination between routing and topology control should be exercised.

C. Experiment 1

In order to appreciate the tradeoffs involved in achieving high fidelity values for data aggregation, we define a base-case where we consider aggregation without scheduling nodes to sleep. In other words, we model the following behavior for the aggregator node: The aggregator waits for *each* one of its children to send one data packet, and then it aggregates and transmits the aggregated packet upstream to its parent. This type of high-fidelity, no sleep behavior is modeled by instantiating the DTMC of Figure 2(bottom) with $p = 0$ and $q = 1$. Note that by setting $p = 0$ but $q < 1$, we model a high-fidelity behavior where an aggregator goes to sleep immediately after it transmits a packet aggregated from packets received from each of its children. These instantiations demonstrate the generality of our model in capturing various aggregation behavior.

Figure 4 shows (on a log-log scale) delay under different load conditions as q increases. We observe that delay decreases with increasing q values for all aggregation/routing trees of various node degrees. At lower q values, nodes sleep for a longer time. This naturally leads to higher delays as nodes trying to transmit will more likely have to wait for their respective parents to be in the active (awake) state.

For increasingly dense networks (*i.e.* higher node degrees and thus fatter shallower aggregation trees), the delay generally increases. Given that an aggregator node has to wait for all of its children to send data, the more children, the higher that waiting time. In addition, contention increases with increasing number of sibling nodes, which causes delay to further increase.

As external (sensing) load g increases, we observe that initially the delay decreases for low values of q. However for higher q, the delay increases under medium load and then decreases under high load. This phenomenon is clearly due to contention. At low q values, due to a low number of active (awake) nodes, the delay decreases under higher load since it becomes more likely that packets (carrying sensed data) are generated and hence progress in aggregation is likely to be faster. In addition at low q values, there is less channel contention. By increasing g and q, increased channel contention causes increased delays. Further increase in the load increases the capacity of the network, offsetting the increase in contention and leading to decrease in delay at high q values.

Figure 5 shows the energy consumption under different load conditions as q increases. We make the following observations. First, the energy consumption increases with increasing q under all loads and over

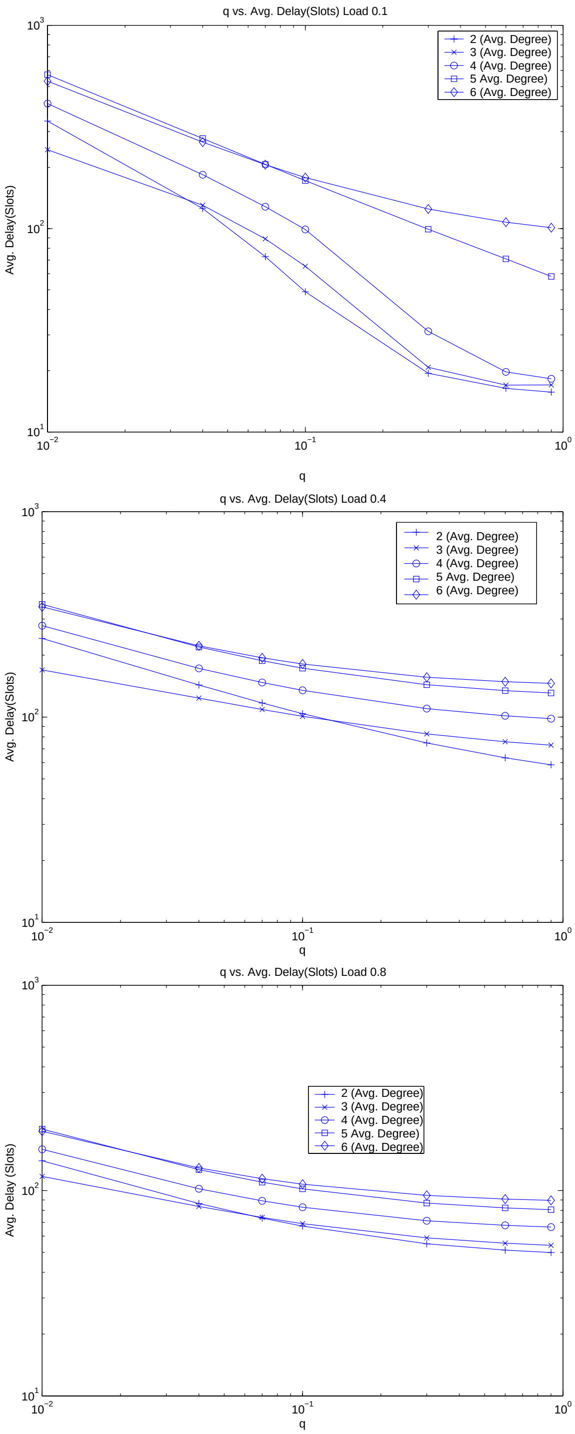

Fig. 4. **(top) Avg. delay under light load; (middle) Avg. delay under medium load; (bottom) Avg. delay under heavy load**

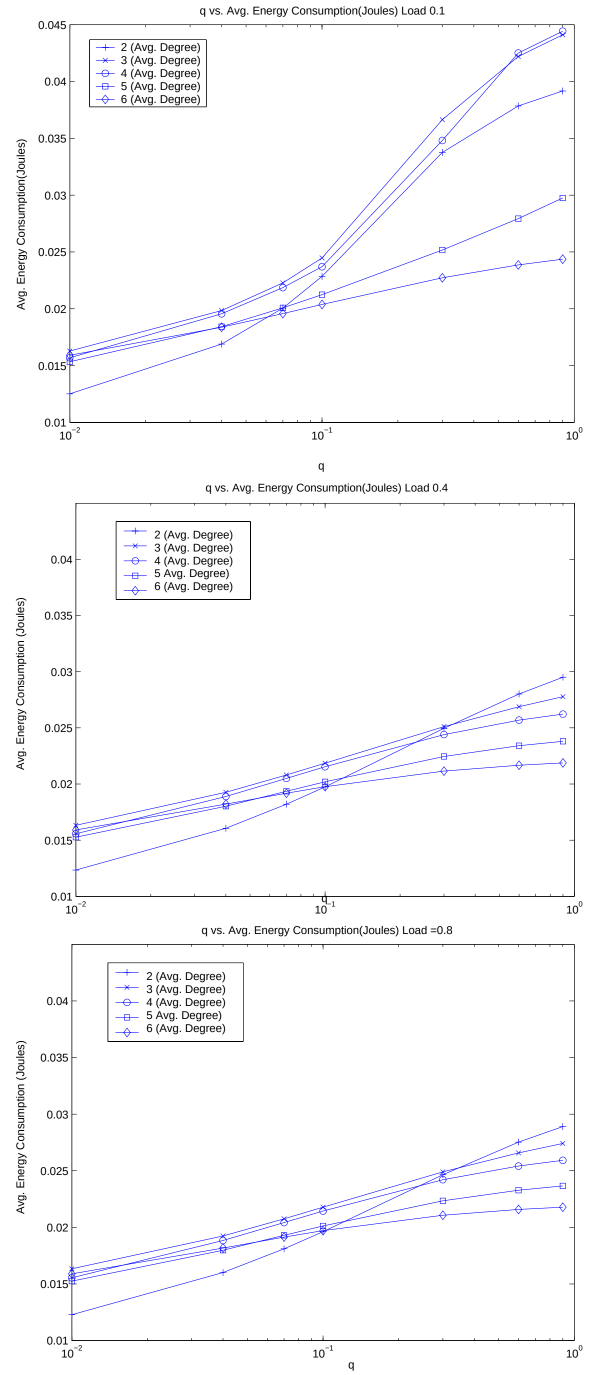

Fig. 5. **(top) Avg. energy consumption under light load; (middle) Avg. energy consumption under medium load; (bottom) Avg. energy consumption under heavy load**

all aggregation/routing tree topologies. This is intuitive since as q increases, the number of active (awake) nodes increases leading to more energy being consumed. Furthermore, this accounts for the *delay-energy tradeoff*, as we noticed that delay decreases (cf. Figure 4) with increasing q, at the expense of such increased energy consumption.

Interestingly, for increasingly dense networks (*i.e.* higher node degrees in the aggregation tree), we observe that for low values of q (*i.e.* fewer active nodes) the energy consumption increases with average node degree. This is because aggregators with higher degree expend more energy, so they complete their aggregation. On the other hand, as q increases, the energy consumption decreases with increasing average node degree. This is

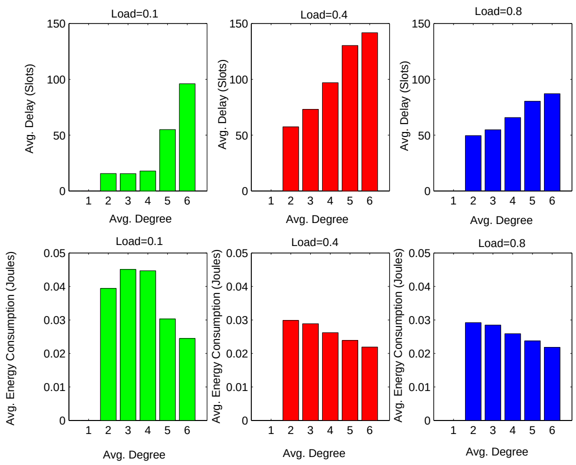

Fig. 6. **(top) Avg. delay; (bottom) Avg. energy consumption under different load values**

because the benefit of aggregation, in terms of energy savings, over trees with higher average node degree becomes more pronounced. This benefit offsets the the energy loss due to increased contention as the load g increases.

Figure 6 singles out the results for the case of full aggregation without scheduling nodes to sleep, *i.e.* $p = 0$ and $q = 1$. As expected, compared to $q < 1$ cases, we observe lower delays and higher energy consumption for $q = 1$ as all nodes remain awake.

D. Experiment 2

In this second set of experiments, we take p = 0.1, that is, an aggregator node may not achieve an aggregation fidelity of one. We show the values of performance measures against the ratio q/p, which represents the number of active (awake) nodes in the network.

Figure 7 shows the delay results. Although the delay trends are similar to those observed in Figure 4, the delay values here are higher. This is because a lower p value means that an aggregator node may go to sleep. So even if a node is ready to transmit, it may not be able to do so successfully if its parent node is sleeping. This causes increase in delay and lower fidelity values. This performance degradation becomes more pronounced at higher levels of the aggregation tree. Clearly, topology control through active/sleep schedules may interfere with aggregation and may offset any benefits from aggregation.

Figure 8 shows the energy consumption averaged over all nodes. Again, although the trends are similar to those observed in Figure 5, the average energy consumption here is lower. This is expected because the lower p value causes nodes to sleep. However under increasing event reporting loads, we observe that the savings in energy

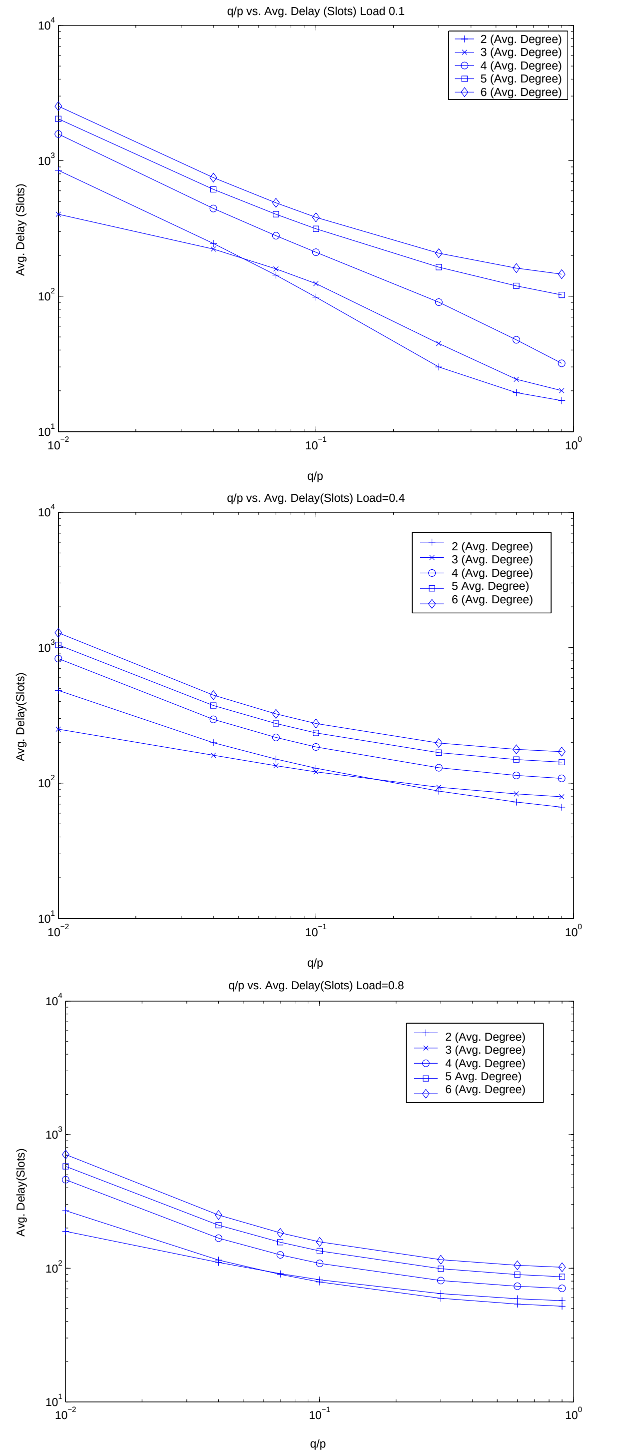

Fig. 7. **(top) Avg. delay under light load; (middle) Avg. delay under medium load; (bottom) Avg. delay under heavy load**

over those in Fig. 4 are not very significant, despite the fact that nodes in this setting do not reach a fidelity value of one. This exposes the tradeoff between fidelity of aggregation and energy.

Figure 9 shows the fidelity-energy index for routing/aggregation trees of varying node degree. We show

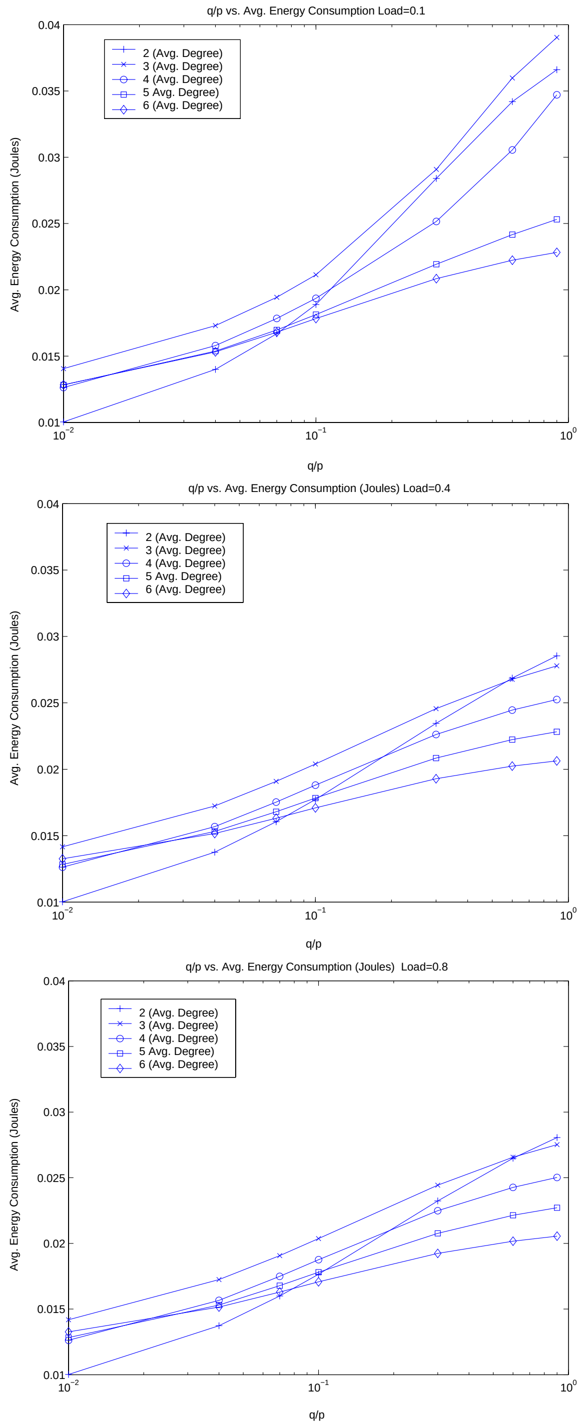

Fig. 8. **(top) Avg. energy consumption under light load; (middle) Avg. energy consumption under medium load; (bottom) Avg. energy consumption under heavy load**

results for different also vary both q and g values. Interestingly, the index increases initially and then decreases. This is because with increasing node degree, the average energy consumption decreases due to increased aggregation, however due to increased channel contention, the steady-state probability of reaching full fidelity decreases. In this setting, the optimal routing tree,

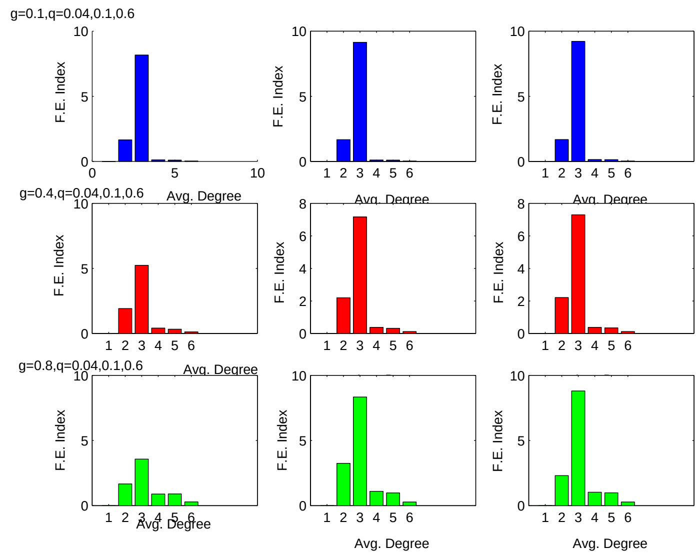

Fig. 9. **(top) Fidelity-energy index vs. degree under low load; (middle) under medium load; (bottom) under heavy load**

in terms of the fidelity-energy index, would be one with node degree of three.

In summary, scheduling nodes to sleep may be harmful in terms of delay due to its interference with the aggregation process.

VI. Conclusion

To our knowledge, we presented the first analytical model that *jointly* captures in-network aggregation and topology control. Our results indicate that, to achieve high fidelity levels in the aggregated data under medium to high event reporting load, shorter and fatter aggregation/routing trees (toward the sink) offer the best delay-energy tradeoff *as long as* topology control is well coordinated with routing. We are currently extending our model to capture the behavior of such coordinated control.

References

[1] R. Kumar, M. Wolenetz, B. Agarwalla, P. Hutto, and U. Ramachandran, “DFuse: A Framework for Distributed Data Fusion,” *Proc. of ACM SENSYS*, November 2003.

[2] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, “TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks,” *Proc. of OSDI*, December 2002.

[3] A. Sinha and A. Chandrakasan, “Dynamic Power Management in Wireless Sensor Networks,” *Proc. of ACM Mobicom*, September 2001.

[4] B. Krishnamachari, D. Estrin, and S. Wicker, “The Impact of Data-Aggregation in Wireless Sensor Networks,” *Proc. of DEBS*, July 2002.

[5] A. Boulis, S. Ganeriwal, and M. Srivastava, “Aggregation in Sensor Networks: An Energy-Accuracy Trade-off,” *Proc. of IEEE SANPA*, May 2003.

[6] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of Network Density on Data Aggregation in Wireless Sensor Networks," *Proc. of IEEE ICDCS*, July 2002.

[7] C.-F. Chiasserini and M. Garetto, "Modeling the Performance of Sensor Networks," *Proc. of IEEE Infocom*, March 2004.

[8] P. Gupta and P. R. Kumar, "The Capacity of Wireless Sensor Networks," *IEEE Transactions on Information Theory*, March 2000.

[9] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*. Phildelphia, PA: ASA-SIAM, 1999.