8-2008

Scalable Medium Access Control for In-Network Data Aggregation

Jamie Macbeth
University of California, Los Angeles, jmacbeth@smith.edu

Majid Sarrafzadeh
University of California, Los Angeles

Scalable Medium Access Control for In-Network Data Aggregation

Jamie Macbeth
Department of
Computer Science
University of California
Los Angeles, CA USA
macbeth@cs.ucla.edu

Majid Sarrafzadeh
Department of
Computer Science
University of California
Los Angeles, CA USA
majid@cs.ucla.edu

ABSTRACT

We present conflict-free and contention-based medium access control (MAC) protocols designed for resource-aware data collection in sensor networks. We are interested in the performance of these schemes when used in in-network data aggregation systems. We introduce a Listen-and-Suppress (LAS) MAC protocol paradigm which can conserve network and node resources and cut delays through the interaction between the constituent nodes. In LAS-TDMA and LAS-CSMA, nodes listen to the channel and suppress their transmissions and sleep if their data is not needed.

Under these conditions, we compare conflict-free scheduling and random scheduling in a general setting along several performance metrics. We find that, for conflict-free scheduling, collecting the aggregate minimum or maximum of a data value in records residing on n nodes in the network requires, on average, $O(\lg n)$ record transmissions and $O(n \lg n)$ listens collectively. Without our scheme, n transmissions and n^2 collective listens are required. We simulate in the random scheduling domain, and examine how delay can be reduced by increasing the offered load on the channel at the cost of greater power dissipation due to collisions. For networks of 20 nodes, LAS-CSMA reduces the average delay by 58% in comparison to CSMA, and for networks of 100 nodes, it reduces the average delay by 80%.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Networks*

General Terms

Design, Performance, Theory

DialM-POMC'08, August 22, 2008, Toronto, Ontario, Canada.

Keywords

Medium Access Control, In-Network Data Aggregation, Wireless Sensor Networks

1. INTRODUCTION

As computing system platforms shrink in size and cost, become more prevalent, gather data, but retain the ability to communicate with their peers, the study of applications, protocols and algorithms for these systems increases in importance. The introduction of battery-powered devices equipped with sensors, actuators, one or more processing elements, and network interfaces has spurred a revolution in systems research. Sensor networks are studied and deployed for a broad range of applications including medical and environmental monitoring, habitat monitoring, and participatory urban sensing [16] [8] [18] [2].

Embedded networking systems include a broad range of devices that collect data and take actions through various types of sensors and actuators, cameras, human input devices or by other means. They exhibit a broad range of sizes, communication link types and power consumption profiles. Typically, data collected by these embedded system nodes is forwarded to a base station node for processing, querying, and other purposes.

But sensor nodes are resource limited in many respects. They are usually battery powered, and in many cases a wired link or wireless radio element dominates the energy usage of the node. More generally, any type of data network may be bandwidth limited. It is desired that the data collection and forwarding process conserves network and node resources and cuts delays. Reducing energy and network bandwidth usage can save costs and improve the lifetime and reliability of the system.

Many applications that use data generated by a sensor network will only require a cross section of the data collected. The process of efficiently collecting summaries and cross sections of data in a sensor or data network is known as data aggregation. Data aggregation is an important component in the operation and use of many sensor and data networks [7] [6] [25].

Data aggregation techniques attempt to conserve sensor network resources by minimizing data processing and network bandwidth usage through careful planning. Specifically, in-network aggregation systems program sensor networks to perform some query processing on the nodes themselves, thereby avoiding costly transmission and receipt of unneeded results in many cases [13] [20]. Through these savings we hope to extend the lives of sensor networks and lower their cost.

1.1 Medium Access Control

Routing methods and protocols are important in getting packets efficiently to their destinations in networks, and important in performance and resource conservation. However, controlling access to the communication medium is essential at the lowest level of network design, where issues such as transceiver power consumption and network fairness, delay, and throughput are of import.

The constraints which are important to bandwidth limited data networks and battery-powered sensor networks present challenges to medium access control protocol design [1]. The traffic created by these networks exhibits patterns that are aren't very well addressed by classical MAC designs, which are rather general purpose. The exploitation of predictable network traffic patterns to extend network lifetime and increase reliability and efficiency is the aim of research of MAC protocols for sensor networks.

In this paper, we present conflict-free and contention-based medium access control protocols designed for resource-aware data collection in sensor networks. We analyze and compare the performance of multiple access protocols when nodes are part of a data aggregation application. In these applications, each node collects and stores data, and, collectively, they are treated like a database of sensor data. Database-style queries are initiated at a base station node, and the data nodes transmit and collect data to respond to the query.

In the Tiny Aggregation Service (TAG) [13], a communication tree is designed to minimize data aggregation processing among nodes as data is sent to the base station node. Also, they minimize power consumption from transmissions between the data nodes and to the base station node. Because not all data nodes can transmit to the base station node, data is routed optimally from the data nodes to the base station node. We consider the case where all nodes are able to transmit directly to the base station node and no routing is necessary. The channel is assumed to be a broadcast medium, and nodes can listen in on each other's transmissions. In our study, the data aggregation network problem is reduced to a problem of medium access control.

We present conflict-free and contention-based medium access control (MAC) protocols designed for resource-aware data collection in sensor networks. Conflict-free multiple access schemes are compared to contention-type random scheduling protocols in query processing applications along several performance metrics. We introduce LAS-TDMA and LAS-CSMA protocol concepts in which nodes listen to the channel and avoid transmitting and sleep if their data is not needed. We compare the overall behavior of conflict-free scheduling and random scheduling in a general setting.

The average delay from query initiation to receipt of the final message in response to the query is studied. We also examine the power dissipation of executing queries by considering the average total number of transmissions made or attempted by all nodes in the network, and we consider the collective average time that nodes spend listening to the network during the query process. The latter is important because for many types of networks listening time is strongly related to node power consumption. These processes are examined mainly through simulation, but we provide theoretical results wherever possible for both contention-based and contention-free multiple access schemes.

2. RELATED WORK

The Tiny AGgregation (known as TAG or TinyDB) service [13] has an database-like interface for performing queries over aggregate data among nodes. It executes the query intelligently to attempt to minimize delay and power consumption, but only in the routing domain. TAG uses a tree structure to route queries to data nodes, and to route query response messages back to the base station. This scheme may be sensitive to transmission failures, so other schemes employ multi-path routing, and make the aggregation scheme immune to problems with duplicate data and data ordering problems.

Both TAG [13] and Cougar [20] allow the user to specify queries with a declarative SQL-style language. This provides a simple data aggregation interface which is independent of the application and sensor network configuration. These data aggregation techniques have been applied to network monitoring. In [23], a system is described which computes aggregates of network properties (packet counts, energy efficiency, packet loss rates, etc.) for applications of sensor networks in harsh environments. Other work addresses systems to aggregate data from sparse sets of nodes which are deployed to detect rare events [5]. A survey of data aggregation techniques for wireless sensor networks can be found here [4].

In the literature, there are also many studies of medium access control protocol design aimed at sensor networks [21] [14] [3] [9]. Ye, Heidemann and Estrin were the first to introduce an energy-efficient MAC protocol designed for sensor networks, S-MAC [22]. S-MAC implements sleep schedules to help nodes conserve energy, and a message passing system to reduce contention.

Kulkarni [10] discusses common communication patterns of sensor networks, presents TDMA algorithms customized for these patterns and compares them to CSMA. Most data aggregation applications fit the "convergecast" pattern in which a large number nodes in the network transmit to a single receiver node. The algorithms fit various grid topologies of node placement; delay and collision characteristics are discussed for both TDMA and CSMA.

DMAC [12] is designed to solve data forwarding interruption problems where not all nodes on a multi-hop path to the base station node are aware of a data collection operation. Zadorozhny, Chrysanthis and Krishnamurthy [24] present a Data Transmission Algebra that can capture the structure of data transmissions, constraints and requirements. It enables cross-network-layer coordination and optimization of transmission scheduling for queries. The inventors of Q-MAC [19] propose a new sleep schedule for query based sensor networks that provides minimum end-to-end latency with energy efficient data transmission.

The rapidly growing collection of MAC protocols designed for sensor nets has been dubbed an "alphabet soup" of protocols [1] [11]. Those that address data gathering operations don't do so in an in-network aggregation context. Rather than commit strongly to a particularly named MAC, we focus on general MAC paradigms for in-network aggregation.

3. NETWORK AND QUERY MODEL

To investigate the in-network data aggregation medium access problem, we assume that we have a data collection network or sensor network with n data nodes, and, for simplicity, a single base station node. It is assumed that the base station node and each data node are able to transmit to and receive from every other node on the network using the same channel. Equivalently, the network is a single-hop broadcast network in which no routing need take place; most generally, the network may be wired or wireless. In the wireless case, we assume that there are no hidden nodes.

The base station node will initiate user queries for data from the n data nodes using the network. In the first phase, typically called the *query distribution* phase, the base station uses the channel to broadcast the query to all data nodes. In the second phase, the *collection* phase, data nodes access the channel to send appropriate records back to the base station node in response to the query.

We focus our attention on channel access for the collection phase. The medium access control protocols we will study allow data nodes to cancel their scheduled transmissions in the collection phase based on transmissions by other data nodes on the network. For simplicity we will idealize the data nodes speed of reception and processing power by allowing them to instantaneously suppress a transmission when data is received. They can also instantaneously turn their radios off in response to a reception. We will disregard cases in which messages may be lost due to noise and other environmental interference, and disregard propagation delay.

3.1 Aggregate Query Taxonomy

Our study considers tradeoffs between multiple access protocol schemes for applications in which a base station node collects results from data nodes using a query and response process. A query system may provide an interface through which the base station node can issue queries in a declarative language similar to that described in [13], but it is not essential to our work.

To discuss aggregate requests through queries, we use a SQL-like query language. In the query context, a table called MYTABLE will name the table of interest, and the column of interest will be called MYCOLUMN. Different types of aggregate queries will engender different patterns of response from the data nodes to the query. For example, a query like "SELECT MYCOLUMN FROM MYTABLE" will require that each data node sends all records from table MYTABLE regardless of the records it may hear being sent from other data nodes for the same query.

However, for a query like "SELECT UNIQUE (MYCOLUMN) FROM MYTABLE" or "SELECT MIN (MYCOLUMN) FROM MYTABLE," not all records are required to be sent by all data nodes. In the former case, if, in response to the query, a data node hears a record value from another data node that is equivalent to one of its own, it need not send along a duplicate of that value. In the latter case, a sensor need not send along a value if, in response to the same query, it hears another sensor send a record value less than one of its own. When the MAX and MIN operators are used, each data node is required to send at most a single value, since it can perform a local aggregation of its records.

We find it useful to refer to the TAG query taxonomy [13]. TAG classifies aggregates according to four properties. In *duplicate-sensitive* aggregate queries, a data node is not required to respond with a record that duplicates a record stored at another data node. However for *duplicate-insensitive* queries the response is unaffected by duplicates.

For *exemplary* aggregates the query result is a representative value from the set of all values measured. In contrast, for *summary* aggregates, the query result is some property computed over the collection of all values among all data nodes. *Monotonic* aggregates represent a total order over all values, and the query result is the greatest or least of the values stored among data nodes.

In this paper we restrict our attention to the *duplicate-insensitive exemplary monotonic* aggregates MAX and MIN. For nearly all other query types, data nodes are unconditionally required to send all stored data records. Because duplicate-insensitive exemplary monotonic aggregates allow data nodes to choose not to transmit based on listening to the transmissions of other data nodes, it appears that our medium access control design will have the most dramatic performance benefits for these queries. Additionally, since MAX and MIN are duals that have identical query response patterns, we can discuss MAX without mentioning MIN without loss of generality.

4. LAS-TDMA

First we will describe Listen-and-Suppress TDMA (LAS-TDMA), a time division multiple access protocol concept which will allow us to avoid unneeded transmissions for MAX and MIN aggregates. Assume that the packet size needed for any of the data nodes to transmit a result in response to a MAX query is T. We will assume that all data nodes only need to respond with a single record since they can perform a local aggregation, and that the response packet size is identical for all data nodes. In the following, we will label the n data nodes $1 \leq i \leq n$ and x_i will denote the locally aggregated value at data node i.

Without loss of generality, let us assume that the query uses the MAX operator; the query result is greatest of all n record values that have been locally aggregated. Assume that all of the record values x_i are independent and identically distributed over a range of possible values $a < x_i < b$.

The query, sent by the base station node, will initiate the data aggregation process, and following the end of query distribution is the TDMA collection frame. The frame will be of length nT, and will be divided into n time slots of length T, with one slot assigned to each data node. In typical TDMA fashion, each data node waits for its slot to transmit, but can also listen to the transmissions of other data nodes in order to decide whether or not it should transmit.

$\mathrm{Trans}(n, i)$ will denote the total number of data nodes in a network of n data nodes transmitting in slot i. The total number of transmissions among all n data nodes will be $\sum_i \mathrm{Trans}(n, i)$, with each transmission consuming a slot. The number of data nodes in a network of n data nodes listening–with the radio in receive mode–at slot i will be $\mathrm{Listens}(n, i)$, and the total number of listens is $\sum_i \mathrm{Listens}(n, i)$. The delay between the start of the response frame and the end of the last time slot where the base station listens to the data nodes is denoted $\mathrm{Delay}(n)$, not accounting for propagation delay.

The LAS-TDMA concept modifies TDMA and allows each data node to listen to all of the of the transmissions before it and conserve resources by transmitting its record value only if it is maximal among all of the records it has heard. As a result, for many slots, there will be no transmission if the maximal record value is transmitted early on in the sequence. To reduce the time that data nodes spend listening on the network, in LAS-TDMA, a data node can turn off the radio completely if it hears a record value greater than its record value.

Without loss of generality, we now label the data nodes in the same order as in the TDMA slot sequence so that data node i is scheduled to transmit at slot i. In this case, the only characteristic of the n record values that affects the total number of transmissions in the scheme is the total order of their record values x_i. Because the record values are independent and identically distributed, each total order is equally probable. As a result, we only need to consider the ordered list of all record values, permutations of this total order, and the ordering in the transmission sequence.

4.1 Expected Number of Transmissions

If data node i holds a record value x_i, it will only transmit if it has a record value greater than all record values transmitted before it: data node i transmits if $x_i < x_j$ for all $1 \leq j \leq i-1$. The probability of this occurring is a ratio of permutations of all record values x_j where $j < i$. There are $(i-1)!$ permutations where $x_i > x_j$ where $\forall j < i$ and data node i has the greatest record value. There are $i!$ total permutations of the x_j such that $\forall j \leq i$.

$$P(\text{Trans}(n,i) = 1) = \frac{(i-1)!}{i!} = \frac{1}{i}$$

For n data nodes, then, the expected number of transmissions is

$$E[\sum_i \text{Trans}(n,i)] = \sum_{i=1}^{n} \frac{1}{i} = H_n$$

Where H_n denotes the harmonic series. It grows asymptotically as $O(\lg n)$ and can be approximated as

$$H_n \simeq \gamma + \ln n$$

Where γ is the Euler-Mascheroni constant $\gamma = 0.577215....$

4.2 Expected Listening Time

If we assume that not every data node needs to know the maximal recorded value, and that the base station node records the result, then data node i need not listen to all $i-1$ transmissions. More specifically data node i need not listen after it has transmitted, or after hearing a record value $x_j > x_i$ (for a MAX query) that will indicate that it won't need to transmit. The 1st data node in the schedule, data node 1, is never required to listen, and always transmits.

Take an LAS-TDMA slot labeled i. No matter how many data nodes transmit in the slots labeled j, $1 < j < i-1$, a data node l only listens to slot i if its value x_l is greater than all values x_j, $\forall j < i$. Since $i-1$ slots have passed, the probability that a data node whose slot has not passed is listening at slot i is the same as the probability that it has a greater value than the $i-1$ previous data nodes:

$$P(x_i > x_j, \forall j < i) = \frac{1}{i-1}$$

Only $n-i$ data nodes have slots that have not passed, and their labels are $i+1$ or greater. So, the expected number of data nodes listening to slot i is

$$E[\text{Listens}(n,i)] = \begin{cases} n-1 & \text{if } i = 1 \\ \frac{n-i}{i-1} & \text{if } i > 1 \end{cases}.$$

The total expected number of slot listens in the frame among all data nodes is

$$E[\sum_i \text{Listens}(n,i)] = n - 1 + \sum_{i=2}^{n} \frac{n-i}{i-1} = (n-1)H_{n-1},$$

which grows asymptotically as $O(n \lg n)$.

4.3 Expected Delay

As the base station is required to listen to an entire TDMA response frame, the expected delay is $E[\text{Delay}(n)] = nT$. For fixed-schedule protocols, we may be able to minimize the number of transmissions, but because of the fixed scheduling, we are unable to affect the total time from the initial to the final transmission. In some cases we would like to increase the efficiency of the channel per unit time, or throughput, at the multiple access protocol level.

We may like to make tradeoffs between the power used for transmissions, and the total time required between the initial transmission and the end of the calculation of the collective record value result. Specifically, if scheduling of transmissions is a bit more loose, we run the risk of having transmissions from different data nodes collide if they try to transmit simultaneously. Data nodes will have to transmit multiple times to get their record value through on the channel, but the more important record values can be heard sooner, cutting down the total delay.

5. LAS-CSMA

We analyze the performance of LAS-CSMA, a random-access protocol, on in-network data aggregation tasks. The random-access protocol domain includes many very widely used MAC protocols such as ALOHA and the many variations of CSMA. To provide a fair comparison with a TDMA protocol scheme, we chose that LAS-CSMA be a slotted non-persistent CSMA to represent the performance of the random-access protocol paradigm.

In reservation-free medium access schemes, no users have prescribed slots for transmission, and each user simply attempts to access the channel at the moment a packet is ready to transmit. In CSMA, users listen to the channel to determine if the channel is in use before transmitting and will attempt to transmit when the channel is idle. If multiple users simultaneously detect the channel to be idle and transmit, there will be a collision; the use of the channel by more than one transmitter ruins all concurrent transmissions. All collided packets must be retransmitted at a later time.

Many features and variants of CSMA attempt to reduce or minimize the negative effects of collisions. In CSMA/CD, or CSMA with collision detection, users listen while they transmit and end their transmissions if they detect a collision. CSMA/CA (collision avoidance) avoids collisions on packets by sending shorter Request To Send (RTS) and Clear To

Send (CTS) messages to reserve the channel. We won't discuss the impact of any of these features on aggregate data queries here.

A typical analysis of random access protocols considers a population of nodes generating packets of equal length T according to a Poisson process with rate λ packets/sec. To include packets that have been scheduled for retransmission at some random time in the future we define the packet scheduling process as a Poisson process with rate $g > \lambda$. g is known as the offered load of nodes in the network and represents the rate at which the schedules of packets occur or arrive in time. Another essential parameter is τ, the maximum propagation delay among all nodes. We assume that nodes can detect when they take part in a collision, but that each node that takes part in a collision transmits for an entire time slot during the collision.

The throughput of CSMA is typically calculated as the average fraction of time that a successful transmission is being made on the channel. In a steady state operation of a channel with slotted non-persistent CSMA, the channel throughput S is given as a function of g, T and τ [17].

We characterize the activity on the channel as a sequence of busy periods and idle periods. A busy period is a succession of slots in which some transmission takes place, whether it is successful or not. In idle periods, no transmissions take place. We will denote the mean length of busy and idle periods B and I respectively. We define a cycle as a time period consisting of a busy period followed by an idle period. We denote by U the amount of time within a cycle during which the channel carries useful information. When a transmission period is successful the channel carries useful information for T seconds, while it carries no useful information in unsuccessful transmission periods.

Given that arrival process is a Poisson process, B, I and U for slotted, non-persistent CSMA are written as [17]:

$$I = \frac{\tau}{1 - e^{-g\tau}}, \quad B = \frac{T + \tau}{e^{-g\tau}}, \quad U = T\frac{B}{T + \tau}P_{suc}.$$

U is given by the number of transmission periods of length $T+\tau$ in a busy period of length B, the useful information of length T in each of these periods, and the probability that a transmission is successful. The probability that a transmission is successful is the ratio of the probability that there is exactly one transmission to the probability that there is at least one transmission:

$$P_{suc} = \frac{g\tau e^{-g\tau}}{1 - e^{-g\tau}}.$$

Then, the channel throughput is the time in a cycle that the channel carries successful transmissions divided by the cycle length:

$$S_{\text{CSMA}} = \frac{U}{B + I} = \frac{Tg\tau e^{-g\tau}}{T + \tau - e^{-g\tau}}.$$

To calculate the total transmissions, including collided transmissions made by multiple data nodes, we calculate the total amount of time that data nodes are transmitting during the busy period, denoted by V. W is the total number of packet arrivals in a mini-slot τ assuming that there was at least one arrival. If there is only one arrival, the transmission in the next period T will be successful, and fits into the case above. If there are multiple arrivals, there will be many transmissions of length T that will comprise a collision.

$$V = T\frac{B}{T + \tau}W, \quad W = \frac{g\tau}{1 - e^{-g\tau}}$$

Let R denote the total transmission rate of successful or unsuccessful transmission. R can be written in terms of the total time that data nodes collectively are transmitting during the busy period, and the length of a cycle:

$$R_{\text{CSMA}} = \frac{V}{B + I} = \frac{Tg\tau e^{g\tau}e^{-g\tau}}{T + \tau - e^{-g\tau}} = S_{\text{CSMA}}e^{g\tau}$$

5.1 Expected Performance

We calculate the expected performance of LAS-CSMA to a first approximation. To calculate the average delay for the data nodes to transmit all needed records, we can reuse our result from the TDMA case by making a few observations. In the LAS-TDMA case, we considered permutations of the total order of record value with values labeled according to the transmission schedule. In the LAS-CSMA case, for any given time interval, the channel will have a number of successful transmission periods in which a data node performed a transmission and there was no contention.

In the LAS-CSMA case, the scheduling is random, but because our selection of LAS-TDMA schedule was arbitrary, we can view a random schedule using the same method. In LAS-TDMA, consider the data nodes remaining in the schedule after a transmission in slot i. The total number of transmissions after slot i depends on the arbitrary ordering of the slots of data nodes with record values $x_j > x_i$. If any data node labeled j with $j < i$ has a record value $x_j > x_i$, then data node j would have transmitted this larger value before i's slot, and i would not have transmitted.

Likewise, in LAS-CSMA, the total number of successful transmissions after the transmission of some data node i depends on the arbitrary ordering of the successful transmissions of data nodes with record values $x_j > x_i$. For a MAX query in LAS-CSMA, the successful transmission of a record by a data node will cause data nodes with lesser records to abandon their transmissions. Therefore, the only successful transmissions are made by data nodes whose record transmissions would have been required if they were ordered as in LAS-TDMA. We conclude that the average total number of transmissions in LAS-TDMA is identical to the average number of successful transmissions in LAS-CSMA in response to the query.

We approximate the delay to completing the query response by considering the expected number of successful transmissions and the time needed for this many successful transmissions under LAS-CSMA. We can calculate this using the throughput of CSMA, the average fraction of time that a successful transmission is being made on the channel. As stated above, the CSMA throughput is a function of the offered load of the data nodes: the frequency at which schedules for retransmission occur in time. It would be expected that, as successful transmissions get through, the scheduling rate would decrease. This is because the data nodes, after hearing a record value greater than theirs, would then cancel their schedules to transmit. This would change the throughput which, in turn, affects the frequency of successful transmissions. In our approximation, we assume that the throughput and the rate of successful transmissions in LAS-CSMA remain constant.

Therefore, the expected delay before the highest measured result is transmitted, thus rendering further transmissions unnecessary, is proportional to the total number of transmissions in the LAS-TDMA case. An approximation of the expected delay $E[\text{Delay(n)}]$ is

$$E[\text{Delay}(n)] \simeq \frac{TH_n}{S_{\text{CSMA}}};$$

which grows as $O(\lg n)$. We compare this to an expected delay of nT for the TDMA case, where we must wait for every slot in the time-divided frame. For LAS-CSMA we are likely to greatly reduce the delay if we can maintain a high throughput.

Similarly, we expect that the average number of transmissions, those that will be successful and those that will be unsuccessful due to collisions, to grow as $O(\lg n)$. If the throughput and offered load are kept constant as functions of n, then the scheduling and execution of all transmission attempts is a Poisson process with a constant rate over the length of the query response. The total number of transmissions depends only on the length of the query response and grows as $O(\lg n)$. We write the total expected number of transmissions as the length of the query response (the delay) multiplied by the total transmission rate:

$$E[\text{Trans}(n)] \simeq \frac{TH_n}{S_{\text{CSMA}}} S_{\text{CSMA}} e^{g\tau} = TH_n e^{g\tau}$$

To calculate the average number of slots spent listening among all data nodes, we realize that data nodes stop listening just before their own successful transmission, and just after hearing a transmission of a record value greater than theirs. On average, half of the remaining data nodes with pending transmissions will suppress their transmissions and stop listening after a successful transmission by another data node. This is because the transmitted record value is the median of the values of all remaining data nodes on average. We expect that all n data nodes will listen until the first successful transmission is made, then $n/2$ will listen until the second is made, and so on. If the throughput, and thus the average interval between successful transmissions, is kept constant, then the average number of listens will grow asymptotically as $O(n)$.

We can approximate $E[\text{Listens(n)}]$ by considering the mean time between successful transmissions,

$$E[\text{Listens}(n)] \simeq \frac{T}{S_{\text{CSMA}}}\ O(n).$$

The average delay, transmissions, and listens for LAS-CSMA are studied further using simulation results which are presented in the following section.

6. RESULTS

In this section we present simulation results for the relative performance of traditional CSMA and LAS-CSMA. Alongside simulation results, analytical results are plotted for TDMA and LAS-TDMA. All results presented are for a data or sensor network responding to a duplicate-insensitive exemplary monotonic aggregate query such as MAX or MIN. It is assumed that each data node has a record with which it will respond to the query. In simulations, each data node is given an integer value chosen randomly to represent the total order of records corresponding to the query. However, the query response pattern only depends on the total order of record values.

In simulation, the time domain is quantized into equal length slots of length T, and it is assumed that each query response will consume a single slot. Delay times, transmission times, and listening times are measured in slots, and they are measured to start just after the end of the query message from the base station. For simulations of CSMA and LAS-CSMA, idle slots and slots in which some transmission occurs are of equal length for simplicity. For each simulation, results are averaged over 1000 trials.

As a first step, we calibrate the CSMA offered load values to use in simulations where we measure performance as a function of n. We did this by simulating the delay of the complete query response of LAS-CSMA for a network of 100 data nodes for a range of values of the offered load g. Results are presented using the normalized offered load $G = gT$ in packets/slot. Figure 3 shows the query delay for a 100 data node LAS-CSMA system as a function of the offered load to the network, and the query delay for a 100 data node LAS-TDMA system. For MAX queries, because each data node will only make, at most, a single successful transmission in response to the query, the offered load is entirely determined by retransmission scheduling.

In our simulations of LAS-CSMA, all data nodes generate a packet to transmit just after the the query transmission completes. This means that at the first time slot of the simulation, all n data nodes, with packets ready, attempt to transmit, resulting in a collision of n transmissions. We have chosen to remove this artifact from our transmission results.

Figure 1 compares the delay characteristics of LAS-CSMA to CSMA where data nodes don't curb their transmissions by listening. These results were obtained by running the LAS-CSMA simulation program with and without the component which removes pending transmissions for data nodes after a successful transmission of a greater record value. The percentage improvement of LAS-CSMA over sending all results via CSMA is displayed in Figure 2.

For the subsequent experiments where performance is measured as a function of the number of data nodes, we simulate the performance for two values of G and a range of values of n. We chose $G = 3$, which roughly minimizes the delay for a network of 100 data nodes, and $G = 1$. Results for non integral values of n appear in results to represent values of n for which multiple values of the non integral transmission probability g were selected.

Figure 4 shows the query response delay of LAS-CSMA and LAS-TDMA as a function of the number of data nodes. Figures 5 and 6 are the average record transmissions as a function of n and the average listens per data node as a function of n respectively. For LAS-TDMA, plots of the analytical results above are given. For LAS-CSMA, simulation data are given for $G = 1$ and $G = 3$.

7. DISCUSSION

Figure 1 compares the MAX query response delay characteristics of LAS-CSMA to a version of CSMA where data nodes don't curb their transmissions by listening. As expected, the delay to successfully receiving all query response

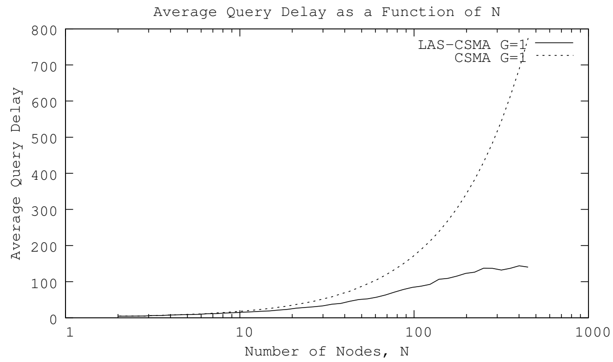

Figure 1: Simulated query response delay of LAS-CSMA and CSMA as a function of the number of data nodes, n.

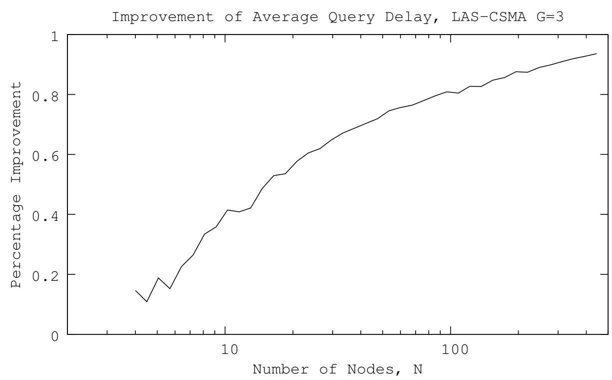

Figure 2: Percentage Improvement in Simulated query response delay of LAS-CSMA as a function of the number of data nodes, n.

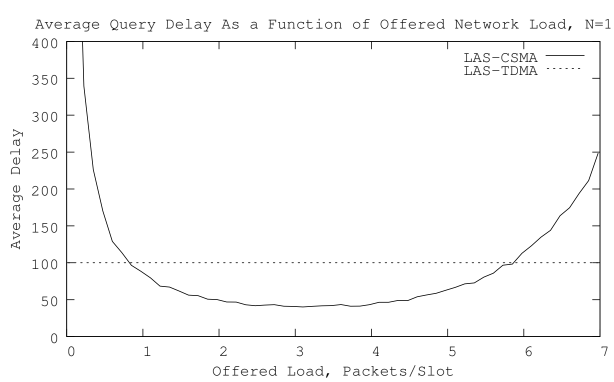

Figure 3: Simulated query response delay of LAS-CSMA and LAS-TDMA as a function of offered load.

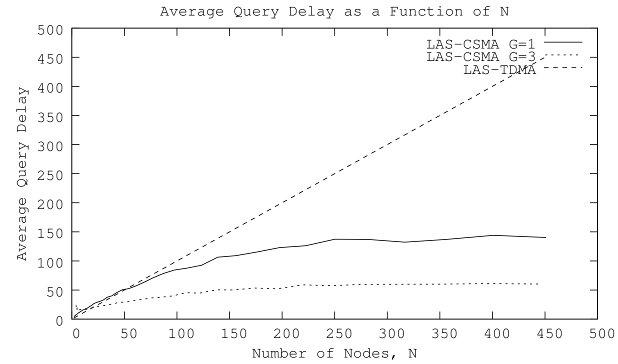

Figure 4: Simulated query response delay of LAS-CSMA and LAS-TDMA as a function of the number of data nodes, n.

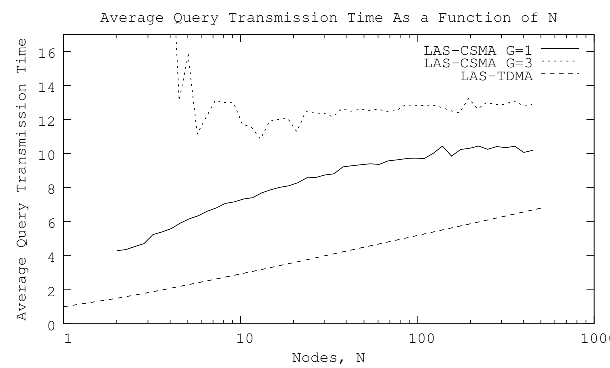

Figure 5: Per-data-node simulated transmissions of LAS-CSMA and LAS-TDMA as a function of the number of data nodes, n.

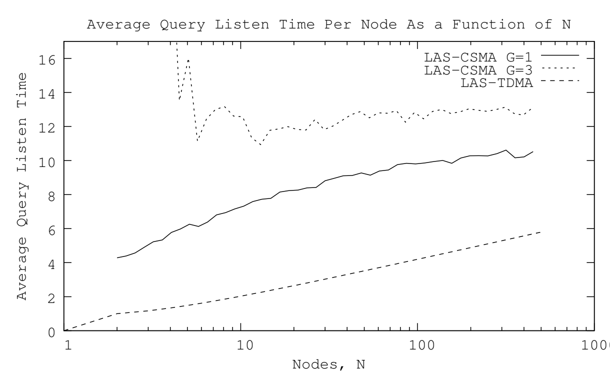

Figure 6: Per-data-node simulated listens of LAS-CSMA and LAS-TDMA as a function of the number of data nodes, n.

transmissions is roughly linear for CSMA, and logarithmic for LAS-CSMA, making the improvement quite dramatic for networks of 100 data nodes or greater. The percentage improvement is displayed in Figure 2. For networks of 20 data nodes LAS-CSMA reduces the average delay by 58% in comparison to CSMA, and for networks of 100 data nodes, it reduces the average delay by 80%.

The delay to receiving the final message in response to a MAX query appears to grow slightly more slowly than a logarithmic function of n, which fits our approximation well. As stated, a major factor missing from the analysis above is the fact that the throughput of the random scheduling depends on the offered load of the responding data nodes. As the query response progresses in time, the offered load decreases as the responding data nodes hear the responses of their counterparts and realize that their transmissions aren't necessary to calculate the MAX. The effect of this is that there is less contention for the network later in the query response. This can increase the throughput if the contention is high to start with, but, near the end of the query response, throughput may decrease due to low utilization. Ideally, we would increase transmission probabilities as the query response progresses to maintain the maximum throughput throughout.

The number of transmissions made (including failed transmissions due to collisions) for LAS-CSMA appears to increase at a rate slightly slower than a logarithmic function of n (Figure 5), which fits our approximation. According to our approximation above the total listen cycles summed over all of the data nodes in the network should grow as $O(n)$. In Figure 6 we plot the number of listens per data node in the network logarithmically as a function of n. For $G = 3$ the per-data-node growth is nearly constant for more than 10 nodes, but for $G = 1$ the per-data-node growth seems to be slightly slower than logarithmic, not constant. Again, we expect that, in our simulations, as the query progresses over time, contention for the network changes. This may be causing the logarithmic growth that we have observed for LAS-CSMA when $G = 1$.

We compare LAS-CSMA, which represents a worst case scenario for contention-based MAC protocols, to LAS-TDMA, representing fixed schedule protocols where the delay is always n slots. For TDMA we must always wait until the schedule ends, in case the last data node in the schedule has a needed record for the result. LAS-CSMA provides slightly longer query delays than a fixed schedule protocol for very small networks, but improves to perform a great deal better as we approach networks of 100 data nodes. At 100 nodes, the average delay of LAS-CSMA with $G = 3$ is less than half that of LAS-TDMA.

One might expect that, due to collisions and network contention, LAS-CSMA would generate many more transmissions than LAS-TDMA for a network of moderate size. LAS-CSMA generates 2 to 4 more transmissions per query response, on average, than LAS-TDMA in our simulations when the normalized offered load, G, is 1. LAS-CSMA generates 8 or more extra transmissions when $G = 3$. Per data node, LAS-CSMA uses 3 to 6 slots more time listening than LAS-TDMA when $G = 1$ and about 8 slots more when $G = 3$. This seems counterintuitive given that the delays are shorter for LAS-CSMA, and listening time might be strongly tied to the total delay. It can be explained by the fact that many nodes will be listening for a long period at the beginning of the query response when the contention is greatest.

This leaves us with many important trade-offs to consider when we design a data or sensor network for good performance in response to query requests. Listening to the responses on the channel from other data nodes as part of the response pattern is essential in cases where minimizing transmissions is important. Transmissions may be very expensive energetically if data nodes need to boost transmission power to reach the base station from long-range. Our non-transmission technique is beneficial if channel bandwidth is high-priced or limited, or if transmissions need to be curbed to reduce channel interference.

On average, with LAS-CSMA, slightly more than 9 effective transmissions can satisfy a query request over a network of 100 data nodes when $G = 1$. A little more than 5 are necessary for a similar query network using LAS-TDMA. Generally LAS-TDMA seems to perform a good deal better than LAS-CSMA in reducing data nodes' transmissions in response to queries. On the other hand, in cases where reducing query delay is the chief concern, the offered load of LAS-CSMA can be adjusted to reduce the average delay to receiving the query result. But this is done at the expense of having more collisions and more energy used in transmissions.

8. FUTURE WORK

Our results demonstrate that in-network data aggregation benefits greatly when data nodes are able to broadcast to each other, and able to listen to each other's broadcasts during the process. For systems where transmission is energy intensive, the data nodes on the network are better able to conserve power. The aggregation process can also reduce the amount of time needed to perform the query response operation. The benefits are a direct consequence of the redundancy in the data being collected. Our results are relevant not only to sensor network applications, but also to any broadcast network where data aggregation operations take place.

We assume a perfect channel with successful receipt of all transmissions, an unrealistic simplifying assumption. This is particularly important because our protocol tolerates lack of transmission. Our future work will provide a way for our contention-based and conflict-free protocols to deal with interference and packet losses, and analyze the effect of losses on performance.

Another future goal is to explore the integration of our technique with other power-saving and contention-limiting schemes. We would like to compare the performance of our scheme to TAG-style tree routing, and explore how well it coexists with other MAC-layer schemes for energy-efficiency– like sleep-wake schedules and collision avoidance. For a given spatial deployment of wireless data nodes, this raises the question of whether performance and power dissipation are better or worse when data nodes boost power or perform multi-hop routing.

The development of software programmable network stacks makes it possible to test the viability of our system in real-world situations. The introduction of software-defined radios [15] encourages the development and testing of novel wireless MAC protocols that integrate the application level objectives with the lower-level network and data-link layers. Prototype implementations which incorporate novel MAC protocol concepts and test them in a realistic environment are an important aspiration.

9. REFERENCES

[1] M. Ali, U. Saif, A. Dunkels, T. Voigt, K. Römer, K. Langendoen, J. Polastre, and Z. A. Uzmi. Medium access control issues in sensor networks. *SIGCOMM Comput. Commun. Rev.*, 36(2):33–36, 2006.

[2] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. Srivastava. Participatory sensing. *ACM Sensys World Sensor Web Workshop*, 2006.

[3] I. Demirkol, C. Ersoy, and F. Alagoz. MAC protocols for wireless sensor networks: A survey. *IEEE Communications Magazine*, 44(4):115–121, Apr. 2006.

[4] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 14(2):70–87, April 2007.

[5] J. Gao, L. Guibas, N. Milosavljevic, and J. Hershberger. Sparse data aggregation in sensor networks. In *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 430–439, New York, NY, USA, 2007. ACM.

[6] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann. Impact of network density on data aggregation in wireless sensor networks. In *ICDCS '02: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, page 457, Washington, DC, USA, 2002. IEEE Computer Society.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, pages 56–67, 2000.

[8] R. Jafari, A. Encarnacao, A. Zahoory, F. Dabiri, H. Noshadi, and M. Sarrafzadeh. Wireless sensor networks for health monitoring. In *MOBIQUITOUS '05: Proceedings of the The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 479–781, Washington, DC, USA, 2005. IEEE Computer Society.

[9] K. Kredo II and P. Mohapatra. Medium access control in wireless sensor networks. *Computer Networks*, 51(4):961–994, Mar. 2007.

[10] S. S. Kulkarni and M. U. Arumugam. TDMA service for sensor networks. In *ICDCSW '04: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04)*, pages 604–609, Washington, DC, USA, 2004. IEEE Computer Society.

[11] K. Langendoen. Medium access control in wireless sensor networks. In H. Wu and Y. Pan, editors, *Medium Access Control in Wireless Networks, Volume II: Practice and Standards*. Nova Science Publishers, Inc., 2007.

[12] G. Lu, B. Krishnamachari, and C. S. Raghavendra. An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. *IPDPS*, 13:224a, 2004.

[13] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: a Tiny AGgregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):131–146, 2002.

[14] P. Naik and K. Sivalingam. A survey of MAC protocols for sensor networks. In T. Znati, K. Sivalingam, and C. Raghavendra, editors, *Wireless Sensor Networks*, pages 93–107. 2004.

[15] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald. Softmac - flexible wireless research platform. In *Fourth Workshop on Hot Topics in Networks (HotNets-IV)*, November 2005.

[16] R. Pon, M. A. Batalin, V. Chen, A. Kansal, D. Liu, M. H. Rahimi, L. Shirachi, A. Somasundra, Y. Yu, M. M. Hansen, W. J. Kaiser, M. B. Srivastava, G. S. Sukhatme, and D. Estrin. Coordinated static and mobile sensing for environmental monitoring. In V. K. Prasanna, S. S. Iyengar, P. G. Spirakis, and M. Welsh, editors, *DCOSS*, volume 3560 of *Lecture Notes in Computer Science*, pages 403–405. Springer, 2005.

[17] R. Rom and M. Sidi. *Multiple access protocols: performance and analysis*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[18] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An analysis of a large scale habitat monitoring application. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 214–226, New York, NY, USA, 2004. ACM.

[19] N. A. Vasanthi and S. Annadurai. Energy efficient sleep schedule for achieving minimum latency in query based sensor networks. In *SUTC '06: Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing - Vol 2 - Workshops*, pages 214–219, Washington, DC, USA, 2006. IEEE Computer Society.

[20] Y. Yao and J. Gehrke. The Cougar approach to in-network query processing in sensor networks. *SIGMOD Rec.*, 31(3):9–18, 2002.

[21] W. Ye and J. Heidemann. Medium access control in wireless sensor networks. In T. Znati, K. Sivalingam, and C. Raghavendra, editors, *Wireless Sensor Networks*, pages 73–91. 2004.

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

[23] R. G. Yonggang Jerry Zhao and D. Estrin. Computing aggregates for monitoring wireless sensor networks. In *The First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA 03)*, Anchorage, AK, USA, May 11 2003.

[24] V. I. Zadorozhny, P. K. Chrysanthis, and P. Krishnamurthy. A framework for extending the synergy between mac layer and query optimization in sensor networks. In *DMSN '04: Proceeedings of the 1st international workshop on Data management for sensor networks*, pages 68–77, New York, NY, USA, 2004. ACM.

[25] J. Zhao, R. Govindan, and D. Estrin. Computing aggregates for monitoring wireless sensor networks, 2003.