A Framework for Extending the Synergy between MAC Layer and Query Optimization in Sensor Networks*

Vladimir I. Zadorozhny
Info Science & Tele Dept.
University of Pittsburgh
Pittsburgh, PA 15260
vladimir@sis.pitt.edu

Panos K. Chrysanthis
Computer Science Dept.
University of Pittsburgh
Pittsburgh, PA 15260
panos@cs.pitt.edu

Prashant Krishnamurthy
Info Science & Tele Dept.
University of Pittsburgh
Pittsburgh, PA 15260
prashant@sis.pitt.edu

Abstract

Queries in sensor networks are expected to produce results in a timely manner and for long periods, as needed. This implies that sensor queries need to be optimized with respect to both response time and energy consumption. With these requirements in mind, we develop novel cross-layer optimization techniques that utilize information about how the medium access control (MAC) layer operates while processing queries in large scale sensor network environments. The central framework of our approach is a Data Transmission Algebra that uniformly captures the structure of data transmissions along with their constraints and requirements. Our framework enables both qualitative analysis and quantitative cost-based optimization of sensor queries. We illustrate the effectiveness of our framework by developing a collision-aware scheduler and evaluating it experimentally.

1 Introduction

We are rapidly moving towards a world that is networked to an unprecedented scale where every device and appliance will have computing and communications capabilities and smart sensor networks will be deployed widely. A large part of the information infrastructure is evolving towards *large-scale wireless* sensor networks, e.g., information tracking systems such as airport security infrastructure, monitoring of children in metropolitan areas, product transition in warehouse networks, fine-grained weather measurements, etc. All of these tasks require efficient mechanisms for querying the sensor data and getting the result of the query in a timely manner. Typical sensor query execution maps into a tree-like data delivery pattern where a responding sensor node sends its data to a neighbor node which transmits it further to the next node towards the requesting node (the root). The data combined from all relevant sensors may be quite large and will require very high data transmission rates to satisfy time constraints. Meanwhile, limitations on sensor node resources like battery power imply that excessive transmissions in response to sensor queries can lead to premature network death.

Several techniques have been proposed to alleviate the problem of limited power at the network level such as energy-efficient routing, clustering and transmission scheduling [12, 25, 11, 6]. Sensor database research has also looked into sensor query processing strategies to minimize the query response time and reduce energy consumption that include sampling [16], prediction [10], approximation [5], and in-network query processing (or aggregation) [2, 15, 21]. With the same goal in mind, our research makes an effort to fuse the techniques and methods currently used in the two different areas of databases and networking. We believe that there is a natural convergence towards combining sensor query processing and lower layer network protocols that can systematically be explored in order to enable efficient operation of sensor networks.

In this paper we introduce an integrated approach to sensor query processing that utilizes performance and functional trade-offs between the query processing schemes, and the medium access control (MAC) layer. An examination of the reasons that affect both energy consumption and response time reveals that (a) data transmission *collisions* represent a major source of energy waste in wireless communication; (b) unnec-

*This research was partially supported by NSF awards IIS0219909, ANI-0123705, and a University of Pittsburgh CRDF award.

Proceedings of the First Workshop on Data Management for Sensor Networks (DMSN 2004), Toronto, Canada, August 30th, 2004.
http://db.cs.pitt.edu/dmsn04/

essary amounts of *active time* for the sensors, due to lack of synchronization among data transmissions, is another major source of wasted energy in sensor networks; and (c) multi-rate data transmissions can have a considerable impact on the energy versus time tradeoff.

We propose a Data Transmission Algebra (DTA) that can capture the information about how the MAC layer operates while processing sensor queries. That is, the DTA can uniformly capture the structure of data transmissions, their constraints and their requirements. Our framework enables both qualitative analysis and quantitative cost-based optimization of sensor queries. Further, it allows the automatic generation and evaluation of alternative routing trees for a given set of queries and network configurations.

Using our framework, we have been able to develop novel *cross-layer optimization techniques*. An example of such an optimization discussed in this paper is *collision-aware query scheduling* that minimizes simultaneous transmissions that interfere with each other. As opposed to other schemes which assume that the MAC layer handles collisions in an appropriate manner, our collision-aware query scheduling reduce the amount of retransmissions and thus saves energy by explicitly considering data transmission collisions.

In realizing the DTA within an efficient query processor and optimizer, we are implementing a novel structure, *a pervasive catalog* that maintains highly available and accurate query statistics and other relevant network run-time information (i.e., meta-data). Such information includes current network topology, processing and transmission delays, collision domains, data rates, and current distribution of already aggregated and materialized data. We evaluate the effectiveness of our framework and the efficiency of the optimization algorithms experimentally.

In Section 2, we set the stage for our framework and overview closely related work. In Section 3, we introduce DTA and its application to cost-based query scheduling. In Section 4, we discuss the challenges in building a pervasive catalog infrastructure. We present the results of our experimental evaluation in Section 5 and discuss the applicability of our approach in Section 6.

2 Background and Related Work

Packet collisions are a major source of energy waste in wireless local communications [14]. Collisions occur when two or more nodes transmit at the same time in an area where both transmissions will have sufficient signal strength at the receiver node. When a collision occurs packets are corrupted and discarded unless there is some sort of capture [18].

Figure 1 elaborates on the concept of the *Collision Domain (CD)* in typical wireless systems such as IEEE 802.11. Assume that a sensor $n1$ wishes to initiate

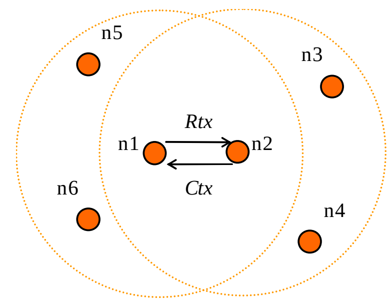


Figure 1: Collision domain of two communicating nodes

transmission to sensor $n2$. Initially, $n1$ sends a request for transmission (Rtx) (called request to send or RTS in 802.11) to $n2$. All other nodes in its transmission range ($n5$ and $n6$ in Figure 1) become aware of the request and remain silent until $n1$ ends the transmission to $n2$. The period of silence is based on virtual carrier sensing where information in the Rtx is used to determine how long they should back off. Note that sensors $n3$ and $n4$ do not sense the Rtx and could potentially transmit at the same time either to $n2$ or to each other resulting in collisions. To prevent this from happening, sensor $n2$ replies to $n1$ with a confirmation (Ctx) (called clear-to-send or CTS in 802.11). This time, the nodes in the transmission range of $n2$ ($n3$ and $n4$ in Figure 1) hear the Ctx and do not transmit until the end of the transmission from $n1$ to $n2$. In this scenario, the nodes $n3$, $n4$, $n5$, and $n6$ belong to the same collision domain. In general, any two communicating nodes ni and nj specify a *collision domain* CD(ni,nj) that can be defined as the union of transmission ranges of ni and nj.

Another way of eliminating collisions is to create an orthogonal transmission mechanism whereby a central authority, such as a base station allocates specific time slots for nodes to transmit based on reservation or polling [19] that will be similar to time division multiple access (TDMA). This however requires a centralized synchronization mechanism that could be fairly complex to implement, consume significant overhead for signaling and be difficult to implement in a multi-hop scenario. Although collisions, overhearing, and idle listening are major sources of energy waste in wireless multi-hop network, control traffic overhead is a significant factor in the energy consumption that should also be taken into account [24]. This can be achieved by efficient methods of wireless meta-data management [7, 26].

An important open research direction related to our work is developing intelligent cost-based strategies for switching nodes to sleep mode to minimize energy consumption [27, 22, 24, 4]. In [28] the authors proposed a cross-layer design for power management. The term "cross layer" here refers to a power management layer utilizing knowledge about route setup and packet forwarding. In-network aggregation has also been proposed to save energy by reducing the amount of communications at the expense of extra computation [15, 23]. TAG [15] and Cougar [23] generate query routing trees in a way similar to what we consider in this paper. TiNA [21] is a middleware layer sitting on top of either TAG or Cougar. TiNA employs query semantics (and in particular, Quality of Data) and can reduce energy consumption significantly, by eliminating redundant data transmissions. However, none of these schemes considered data transmission collisions to reduce the amount of retransmissions and thus save energy. All of these schemes assume that the MAC layer handles collisions. Unlike TAG, Cougar, and techniques similar to TiNA or GaNC [3], our approach employs query and network metadata to generate query plans and routing trees that avoid collisions and maximize sleep time, while balancing response time and energy consumption.

3 Query Scheduling using DTA

We develop an algebraic framework that allows a sensor query optimizer to arrange concurrent data transmissions in the query tree so as to avoid collisions. The idea is that the query optimizer generates a schedule for data transmissions that is disseminated to each node in the query evaluation tree. As opposed to TDMA-like policies, the schedule is a suggested strategy that avoids collisions but it is up to individual node to decide how to behave within a set of constraint intervals specified by the schedule. In the event that a node cannot follow the schedule to avoid collisions, collisions are handled by the MAC layer. Thus, instead of delegating the collision resolution solely to the MAC layer, our framework utilizes query semantics to coordinate transmissions between sensor nodes.

3.1 Data Transmission Algebra

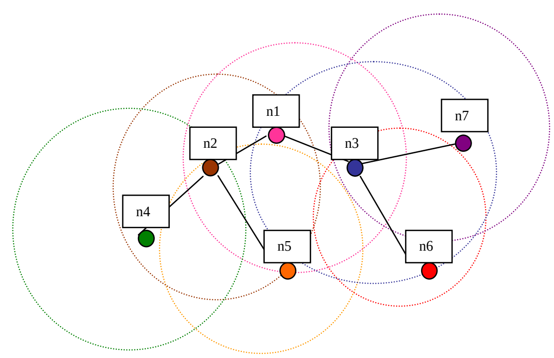
We define a Data Transmission Algebra (DTA) that efficiently enables such query scheduling. The DTA consists of a set of operations that take transmissions between wireless sensor nodes as input and produce a schedule of transmissions as the result. We call a one-hop transmission from sensor node ni to node nj an *elementary transmission* (denoted $ni \sim nj$). We also use a special symbol, *null*, that denotes a *completed* (or *empty*) transmission. Each transmission $nj \sim nj$, which is not empty is associated with a collision domain CD(ni, nj) as defined in Section 2. A transmission schedule is either an elementary transmission, or a composition of elementary transmissions using operations of the DTA as described below. The DTA includes three basic operations that combine two transmission schedules A and B:

1. ***order(A,B)*** $\equiv o(A,B)$. This is a strict order operation, that is, schedule A must be executed before B.

2. ***any(A,B)*** $\equiv a(A,B)$. This is an overlap operation that allows schedules A and B to be executed concurrently.

3. ***choice(A,B)*** $\equiv c(A,B)$. This is a non-strict order operation that either schedules A before B, or puts B before A. Thus, $c(A,B) \equiv (o(A,B) \vee o(B,A))$.

As an example of DTA operations consider the query tree in Figure 2 which was generated for some query Q. This shows an *initial* DTA specification that reflects the basic constraints of the query tree. The circles represent the ranges of the sensor nodes. For the purposes of this example, we assume that the transmission power is constant and the nodes are stationary. The initial specification consists of a set of strict order and overlap operations. For instance, operation $O1$ specifies that transmission $n2 \sim n1$ occurs after $n4 \sim n2$ is completed. This constraint reflects the query tree topology. Operation $A1$ specifies that $n4 \sim n2$ can be executed concurrently with $n6 \sim n3$, since neither $n3$ nor $n6$ belong to CD($n4$,$n2$), and neither $n4$ nor $n2$ are in CD($n6$,$n3$).



Initial DTA specification:

O1: *o*(n4~n2, n2~n1) *A2:* *a*(n4~n2, n7~n3)
O2: *o*(n5~n2, n2~n1) *A3:* *a*(n4~n2, n3~n1)
O3: *o*(n6~n3, n3~n1) *A4:* *a*(n5~n2, n6~n3)
O4: *o*(n7~n3, n3~n1) *A5:* *a*(n5~n2, n7~n3)
A6: *a*(n5~n2, n3~n1)
A7: *a*(n6~n3, n2~n1)
A8: *a*(n7~n3, n2~n1)

Figure 2: Query tree and initial DTA Specification

Each operation of the initial specifications defines a simple transmission schedule consisting of two elementary transmissions. The DTA introduces a set of transformation rules that can be used to generate more complex schedules from the initial specification. Figure 3 shows examples of DTA transformation rules $R1$-$R6$, and illustrates how these rules apply towards generating more complex schedules $A9$, $A10$ and $A11$ from the initial specification in Figure 2. $A9$ schedules

Example DTA transformation rules:

R1: *o*(A,B) ≠ *o*(B,A)
R2: *a*(A,B) = a(B,A)
R3: *c*(A,B) = *c*(A,B)
R4: *a*(A,B) & *a*(A,C) = *a*(A, *c* (B,C))
R5: *c*(A, *c*(B,C)) & *o*(A,B) = *c*(*o*(A,B), C)
R6: *c*(*c*(B,C), A) & *o*(B,A) & *o*(C,A) = *o*(*c*(B,C), A)

Example of DTA transformations:

A1,A2,R4 imply:
A9: *a*(n4-n2, *c*(n6~n3, n7~n3));
A3, A9, R4 imply:
A10: *a*(n4-n2, *c*(*c*(n6~n3, n7~n3), n3~n1));
A10,O3,O4,R6 imply:
A11: *a*(n4-n2, *o*(*c*(n6~n3, n7~n3), n3~n1));

Figure 3: Example of DTA transformations

schedule	cost
ni~nj	Tp(*ni*)+Ttx(*ni~nj*)+Tp(*nj*)
o(A,B)	cost(A)+cost(B)
a(A,B)	max(cost(A),cost(B))
c(A,B)	cost(A)+cost(B) – Tf

Figure 4: Estimating costs of schedules

three elementary transmissions, while each of $A10$ and $A11$ schedules four elementary transmissions.

None of the simple or complex transmission schedules considered so far include all elementary transmissions of the query tree, so we call them *partial schedules*. Our goal is to generate DTA expressions for *complete schedules*. A complete schedule includes all elementary transmissions of the query tree. Below we introduce a cost model for optimizing data transmissions in order to generate complete and efficient schedules.

Figure 4 shows simple cost estimation expressions for each of the DTA expressions. In this case, the cost corresponds to the execution time associated with a particular schedule. For clarity of presentation we ignore energy consumption at this point. For example, the execution time of elementary transmission $ni \sim nj$ consists of local processing times Tp at nodes ni and nj plus the time Ttx required for transmitting data from ni to nj.

The execution time of strict order of schedules A and B is the sum of execution times of A and B. For overlapping schedules A and B, the execution time would be the maximum of the execution times of A and B. Finally, the execution time of the choice between A and B is the same as the execution time of the strict order minus a predefined time factor Tf. Tf indicates that in general, the optimizer prefers the choice operation over strict order, since the latter restricts flexibility of the optimizer in query scheduling. We ignore propagation times as they are negligible in this case.

3.2 Scalable DTA Scheduling

Basic DTA scheduling may be expensive due to its combinatorial nature. The number of alternative schedules grows at least exponentially with the number of sensor nodes and elementary transmissions participating in a query. In order to decrease this complexity, we developed heuristic-based pruning methods that eliminate suboptimal alternatives. We also explored randomized algorithms to cope with the expected complexity of queries in large scale sensor networks. Randomized algorithms [13] are scalable techniques to solve complex combinatorial optimization problems that search for a solution in a large space of all possible solutions. Each solution is associated with application-specific costs. Randomized algorithms will search for a solution with the minimal cost by performing random walks in the solution space via a series of valid moves. In our case possible solutions are DTA schedules.

M1.	Choice commutativity	c(X,Y) ↔ c(Y,X)
M2.	Overlap commutativity	a(X,Y) ↔ a(Y,X)
M3.	Choice associativity	c(c(X,Y),Z) ↔ c(X,c(Y,Z))
M4.	Overlap associativity	a(a(X,Y),Z) ↔ a(X,a(Y,Z))
M5.	Order associativity	o(o(X,Y),Z) ↔ o(X,o(Y,Z))
M6.	A/C exchange	a(X,c(Y,Z)) → c(a(X,Y),Z)
M7.	Left A/O exchange	a(X,o(Y,Z)) → o(a(X,Y),Z)
M8.	Right A/O exchange	a(X,o(Z,Y)) → o(Z, a(X,Y)
M9.	C/A exchange	c(a(X,Y),Z) → a(X,c(Y,Z)), provided ***any(X,Z)*** holds
M10.	Left O/A exchange	o(a(X,Y),Z) → a(X,o(Y,Z)), provided ***any(X,Z)*** holds
M11.	Right O/A Exchange	o(Z, a(X,Y) → a(X,o(Z,Y)), provided ***any(X,Z)*** holds

Figure 5: Valid moves between DTA Schedules

Figure 5 represents valid moves between DTA schedules. Here *any(S1,S2)* is relation between two DTA schedules $S1$ and $S2$ defined recursively as follows:

any(X, Y) if $a(X, Y)$ or $a(Y, X)$.
any$(X,a(Y,Z))$ if *any*(X, Y) and *any*(X,Z).
any$(X,c(Y,Z))$ if *any*(X, Y) and *any*(X,Z).
any$(X,o(Y,Z))$ if *any*(X, Y) and *any*(X,Z).

Different randomized algorithms employ different moving strategies and stopping conditions. Some of the most well-known randomized optimization algorithms are Iterative Improvement (II), Simulated Annealing and Two-Phase Optimization [13]. We explore performance of each of them for the purpose of scalable DTA scheduling. In Figure 6, we illustrate how DTA scheduling can utilize II algorithm.

```
Procedure II () {
minS = Sser ;
  while (not stopping_condition) do {
    S = random DTA schedule
    while (local_minimum(S)) do {
      S' = random DTA schedule
           in neighbors(S)
      if cost(S') < cost(S)  then  S=S'
    }
    if cost(S) < cost(minS)  then  minS=S
  }
return(minS)
}
```

Explanation of variables and parameters:
- ***minS*** – current DTA schedule with minimal cost;
- ***Sser*** – random serial DTA schedule;
- ***S*** – random initial DTA schedule;
- ***neighbors(S)*** – a set of schedules that can be generated from S via one valid move;
- ***stopping_condition*** – number of considered initial schedules;
- ***local_minimum(S)*** – a number of neighbors of S to be tested, of which none has lower cost than S. If the test is successful, S is considered to be a local minimum

Figure 6: II Algorithm for DTA Scheduling

3.3 Impact of multi-rate transmissions

Multi-rate transmission is supported in the new generation of standards for wireless local communications (such as 802.11a/b/g) as well as in evolving future technologies. Under these standards, it is possible for nodes to transmit at different data rates depending on signal quality. Usually, signal quality degrades with distance (although this is not the only reason) [18]. The path loss (that is dependent on the environment and frequency), the modulation scheme, the transmission power and the receiver sensitivity influence the data rates that can be provided for a given quality (bit error rate or packet error rate). For instance, consider phase shift keying (PSK) based modulation schemes. In the case of PSK, the number of bits/symbol will affect the bit error rate. Consider binary PSK (BPSK), quaternary PSK (QPSK), 8-PSK and 16-PSK that transmit 1, 2, 3 and 4 bits per symbol respectively. The energy per bit to the noise power spectral density ratios required by these modulation schemes to achieve a bit error rate of 10^{-5} are respectively 10, 10, 13.5 and 18 dB [20]. Note also that compared to BPSK, QPSK, 8-PSK and 16-PSK can transmit data at 2, 3 and 4 times higher rates in the same bandwidth. For actual products based on 802.11, similar properties apply. Assuming a constant standard transmission power, an 802.11 based node may be able to transmit data at 11 Mbps to another node that is 90 ft away, but only at 5.5 Mbps to another node that is 150 ft away[1] or 2 Mbps to a node that is 210 ft away using 802.11b technology. If the transmission power is increased or the environment is open space, the range of transmission at 11 Mbps could be increased. In outdoor areas, the distances up to which certain data rates can be achieved will be different. For example, a data rate of 11 Mbps can be achieved if the nodes are separated by 200m, 5.5 Mbps if the separation is between 200 and 300 m, and 2 Mbps if it is between 300 and 600m[2].

Alternatively, by reducing the transmission power, the range can be reduced while keeping the data rate at say 2 Mbps. Reducing the range also implies that the collision domain is shrunk allowing the possibility of concurrent transmissions between different sensor nodes. This brings up interesting opportunities for creating minimal cost query schedules. Our query optimizer estimates the transmission power, data rates, and order of transmission of sensor nodes that minimizes costs in multiple ways. We discuss such scenarios next.

Certain sensor nodes may be low on battery power and if this information is known, it would be advantageous to reduce their transmission power and range to prolong the network life. There may be sensor nodes that have sufficient energy and could increase their transmission power for a certain period of time to bypass some hops and directly reach the node that initiated the query. In this case the DTA will utilize a cost model that takes into account both response time and energy consumption while trading certain degree of concurrency (i.e., number of operations/transmissions that can overlap in the initial specification) for increasing the speed of some transmissions.

Figure 7 illustrates this idea with two simple transmission scenarios. In scenario **(a)**, transmissions $n4 \sim n2$ and $n5 \sim n3$ can occur concurrently, which is reflected by the overlap operations $A1$ in the corresponding initial DTA specification. By increasing transmission power of sensor $n4$ (scenario **(b)**), the opportunity of transmitting $n4 \sim n2$ and $n5 \sim n3$ concurrently disappears, which results in a more restricted DTA specification. However, the gain in $n4 \sim n2$ transmission speed, as well as a possibility for $n4$ to transmit directly to $n1$ can overcome the lack of concurrency in scenario **(b)** under certain circumstances. Apparently, in this case $n4$ would spend more energy to complete its transmission. We are extending the DTA cost model to capture the tradeoffs between transmission speed, transmission power and degrees of concurrency in sensor query processing. Assuming general modulation schemes and suitable ranges of transmit powers we plan to compare the results with measurements with real products like 802.11 and Bluetooth.

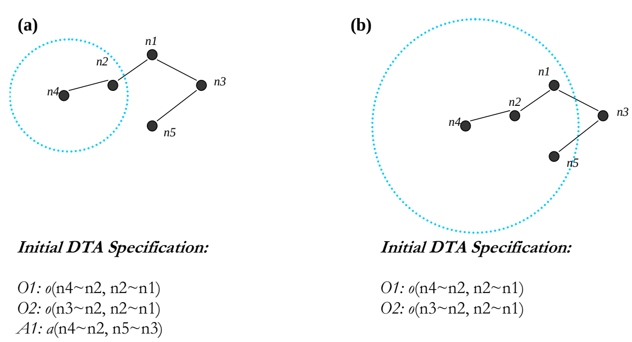


Figure 7: Explanation of the tradeoff between power, speed and concurrency

[1] These numbers are based on measurements in indoor areas by Atheros [1].

[2] These numbers are based on product information by Firetide [9].

4 Pervasive Catalog for DTA Query Scheduling

In order to support DTA query scheduling the optimizer should rely upon highly available and accurate query statistics and other relevant network meta-data including current network topology, processing and transmission delays, collision domains and current distribution of pre-aggregated and materialized data. Such query statistics and network meta-data should be stored in a highly available distributed repository with varying freshness, precision and availability requirements. Design and implementation of such a repository together with an appropriate signaling system is a considerable challenge. In this section we report our on-going research on designing a *pervasive catalog system (PCat)* that implements such a meta-data repository.

We are considering three basic catalog implementation alternatives: (1) *centralized scheme*, where all the statistics metadata is maintained in a central node accessible through a base station (2) *distributed scheme*, where each node maintains its own metadata statistics, and (3) *hybrid scheme*, where some sensor nodes maintain their own statistics and host statistics about other nodes and sub-networks.

Centralized Scheme. In a centralized scheme, the root node is a base station (BS) with a large broadcast area and unlimited power supply since it is presumably a fixed node and located in an opportunistic location. The BS maintains the statistics on processing and transmission delays, the network topology, and collision domains. The synchronization of the participating nodes can be easily achieved, since every node listens to the same BS. The BS performs query scheduling using DTA and broadcasts the resulting schedule to every node in the network. For this purpose, out-of band signaling or periodic beacons can be employed. Note that sensor nodes need to only receive this information, but need not transmit information directly to the BS as this may require large transmit powers and incur large energy consumption.

Distributed Scheme. In this scheme, each wireless node maintains statistics meta-data about itself. We consider only local sensor processing times (Tp), and the transmission time to a parent node (Ttx). A query can be submitted at a root node of a routing tree and then it can propagate down the tree to every node. After receiving a query, each child node in the lowest level provides its statistics, i.e., processing and transmission times (delays) to their parent (Figure 8 - top). Then, the parent node performs query scheduling for each child node using the DTA in order to minimize collisions and the active time for the parent's receiver. The parent node returns this schedule to its children (Figure 8 - bottom). After scheduling its children, the parent node estimates and sends its own processing and transmission delay information to an upper level parent node. Then the same process propagates up the routing tree until it reaches the root node. The above process can vary depending on actual query and network statistics. For example, the transmission time of the latest node can be fixed and transmissions for the remaining nodes should be scheduled ahead of the latest node.

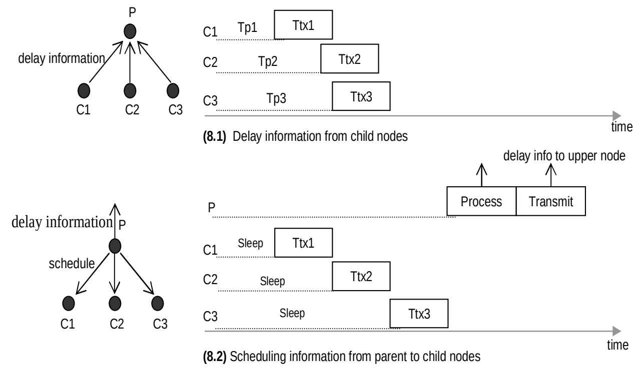


Figure 8: Distributed Query Scheduling

Hybrid Scheme. Under the hybrid scheme, every node in the sensor network is associated with its own statistics metadata, and some of the nodes can additionally host statistics meta-data (perhaps more summarized) about a subnet of devices in their local meta-data repository. Hybrid PCat implements adaptive distribution granularity that minimizes control and meta-data traffic, as well as energy consumption while providing certain level of meta-data accuracy and freshness. It can be tuned for either maximum lookup or update performance and levels in between. In this way PCat is implementing different tradeoffs between data availability, freshness and precision, ranging from purely distributed schemes to a purely centralized scheme.

5 Experiments and Analysis

In this section, we discuss the first results of the evaluation of our framework. First, we show the potential performance gains of DTA schedules. These are generated by a basic DTA scheduler that enumerates all possible schedules exhaustively. Second, given that such a DTA scheduler does not scale, we evaluated the performance of an Iterative Improvement (II) algorithm for DTA scheduling that is capable of handling large query trees. Finally, we compared DTA scheduling with 802.11 MAC in order to put our results in a better perspective.

5.1 Behavior of the DTA schedules.

In order to evaluate our approach, we implemented a basic DTA scheduler in Arity Prolog. Here, we report on the behavior of the DTA scheduler for a medium complexity query tree involving ten sensor nodes with overlapping collision domains. Processing and transmission costs were generated randomly using Gaussian distributions.

The basic DTA scheduler generated schedules stage by stage starting from initial schedules with two elementary transmissions (stage 1). Stage 2, 3 and 4 represent schedules with 3, 4 and 5 scheduled transmissions. Stage 5 includes complete schedules covering all elementary transmissions of the query tree.

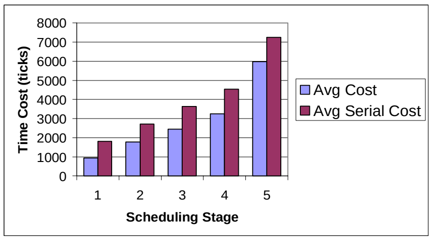


Figure 9: Comparison of DTA scheduling with serial scheduling

Figure 9 shows the average query execution time for different scheduling stages. We compare the DTA scheduling with a *serial scheduling strategy* that performs elementary transmissions sequentially. For each scheduling stage we report the average execution time of all its schedules. We observe that at each scheduling stage, the approach that uses DTA considerably outperforms serial scheduling.

Figure 10 reports on the average benefit that each scheduling stages gains from concurrent transmissions. Intuitively, the benefit is part of the time cost that the DTA scheduler is able to "hide" scheduling some transmissions concurrently. The benefit is defined recursively for each of DTA operations. The benefit of a(X,Y) is equal to minimum of costs cost(X) and cost(Y). For the rest of the DTA operations the benefit is equal to zero. Thus, any serial schedule has a zero benefit.

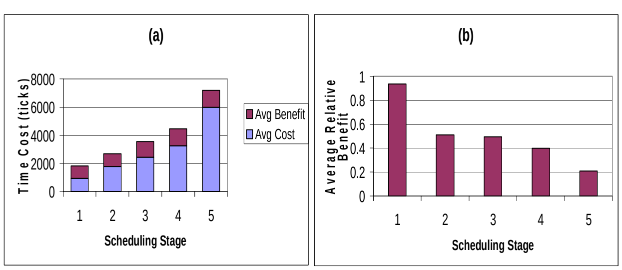


Figure 10: Time cost (a) and relative benefit (b) of DTA scheduling

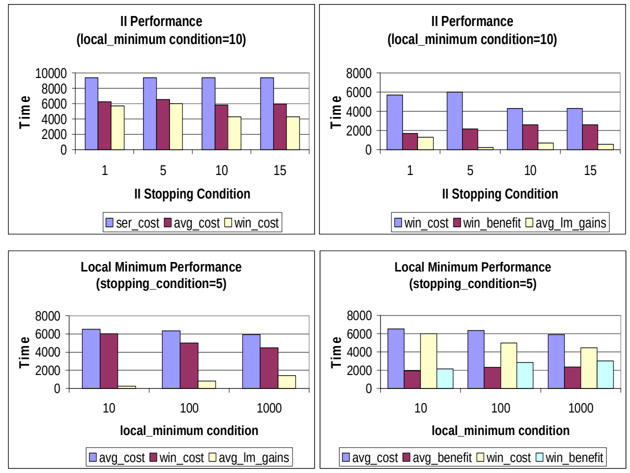


Figure 11: Performance of II-based DTA Scheduler

Figure 10(a) compares values of average time cost and average benefit for each scheduling stage. With the increase of the number of transmissions the benefit grows, but not as fast as the time cost. Figure 10(b) plots the average relative benefit as a percentage of the overall average time cost per scheduling stage. We observe that for simple initial concurrent schedules the benefit is almost equal to the time cost. This is an expected behavior. Elementary transmissions have comparable time costs. By scheduling them concurrently, DTA hides on average one half of the time cost of their serial execution. However, for complete schedules (stage 5) the average relative benefit is as low as 0.2, which means that only 20% of the total serial cost has been hidden. This is also an expected behavior, since complete schedules are composed of non-elementary transmissions (sub-schedules) with higher variance in their time cost. Thus, it is more challenging for the DTA scheduler to hide time costs of non-elementary sub-schedules.

5.2 Evaluation of the II-based DTA Scheduler

Figure 11 shows some of our experiments that evaluated the performance of the Iterative Improvement (II) algorithm for DTA scheduling. It reports average time cost and benefit of all considered schedules (*avg_cost* and *avg_benefit*) and time cost and benefit of the winner schedule chosen by II algorithm (*win_cost* and *win_benefit*). In addition to costs and benefits of the schedules, we also report a value of average gain received from the local minimum phase of the algorithm (*avg_lm_gains*). The local minimum gain occurs when II algorithm improves a random initial schedule via given number of random moves. This number should be no greater than the local minimum condition.

The upper left graph in Figure 11 illustrates a consistent improvement of II performance as we increase the values of the stopping condition with fixed local minimum condition of 10. We also provide a time cost

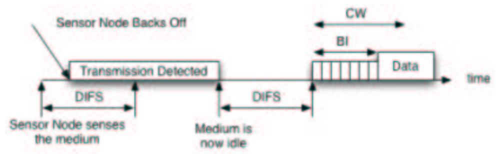


Figure 12: Back-off in 802.11 MAC

of a serial schedule (*ser_cost*) as a reference point and a worst case scenario.

The upper right graph also reports on benefit and local minimum gains of the winner schedule. While we observe steady increase of the benefit value, the local minimum gain behaves quite sporadically. This is an expected behavior, since for each value of II stopping condition we set the same local minimum condition. Thus, in general we should expect a random value of *avg_lm_gains*.

In order to explore the performance of the local mimimum phase we plot the cost, benefits and local minimum gains for different values of the local minimum conditions (lower two graphs of the Figure 11). We observe that the performance of the II algorithm consistently improves as we increase the values of the the local minimum conditions.

In summary, our experiments showed that II algorithm scales well for large query trees and demonstrates reasonable performance with proper parameter settings.

5.3 Comparison of DTA scheduling with 802.11 MAC

We note that 802.11-like transmissions may be faster than simple serial schedules considered above under lightly loaded conditions, but would still be slower than DTA. For example, in Figure 2, let us assume that the MAC layer independently operates and the query optimizer creates no schedule. For this topology, there could be concurrent transmissions $n4 \sim n2$ or $n5 \sim n2$ and $n6 \sim n3$ or $n7 \sim n3$. However, there is no guarantee which of these will occur first. Consider the contention between $n4 \sim n2$ and $n5 \sim n2$. Suppose the medium is idle and both $n4$ and $n5$ sense it as idle at the same time upon receiving the query. They will both wait for a time called distributed inter-frame space (DIFS) and transmit the packet simultaneously, resulting in a collision. If they sense the channel at slightly different times, one of the nodes will transmit first resulting in the second node backing off as shown in Figure 12.

Suppose node $n4$ was able to transmit first. Node $n5$ will back-off and wait till node $n4$ completes its transmission. After node $n4$ completes its transmission, $n5$ will wait for an additional time equal to DIFS and anywhere between 1 and 7 slots each of duration 20 μs before it attempts transmission. The number of slots (called the back-off interval - BI) will be selected randomly in a window (called the contention window - CW). In case there is a collision, the CW is doubled. This doubling occurs each time there is a collision (resulting in up to an increase of 1024 times). If there are several sensor nodes in the same collision domain, that need to transmit data, the process would result in some collisions and considerable additional waiting time. A similar scenario happens between nodes $n6$ and $n7$. The number of collisions would also depend upon network topology and the type of queries (how large the traffic will be at given points in the sensor network).

We believe DTA scheduling would reduce collisions and improve the energy savings. Collisions result in completely wasted energy. In addition, during the backoff slots, sensor nodes will be continuously monitoring the medium resulting in wasted energy consumption. We have also ignored the acknowledgment process at the MAC layer in this preliminary analysis. Currently we are implementing simulations in OPNET Modeler [17] to test the degree of time and energy savings that DTA would provide over regular 802.11-like transmissions

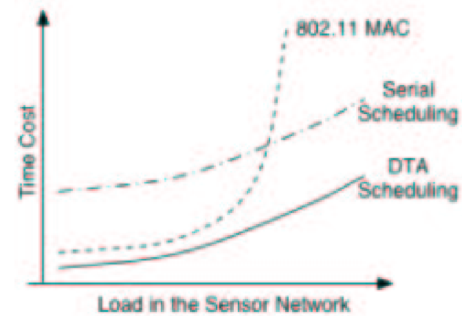


Figure 13: Time Costs with different scheduling schemes

Figure 13 represents the expected relationship between 802.11-MAC, serial and DTA-based transmissions. As discussed above 802.11-like transmissions may be faster than simple serial schedules considered above under lightly loaded conditions, but would still be slower than DTA. For higher network loads and more complex sensor queries the performance of 802.11-MAC considerably degrades comparing to serial and DTA scheduling. DTA will always outperform serial scheduling. Our preliminary simulation results support this assumption. Currently we are undertaking a comprehensive study of different query scheduling options.

6 Discussion on DTA Applicability

In this paper, we use the IEEE 802.11 standard as the basis for the medium access control mechanism as we are considering large scale sensor networks that may need to transmit large amounts of data over fairly long distances. For lower data rates (on the order of a few kbps) and smaller ranges, a more suitable mechanism is the newly proposed IEEE 802.15.4 standard [29] for low-rate wireless personal area networks. We note that this mechanism also employs CSMA/CA for medium access although the details are different.

In explaining the DTA and in the simulations, we use a circular coverage area for each node. In reality, the radio propagation conditions determine the shape of the coverage area and this will be irregular. We do note that circular coverage areas are commonly used as approximations and also for mathematical tractability. They do provide us with insights as to how a proposed mechanism may perform. Moreover, the DTA does not depend on the shape of the collision domains, but rather on the knowledge of what transmissions from what nodes are likely to collide. For this, it is sufficient if the interference characteristics of sensor nodes are known a priori. In a fixed topology with a small number of nodes, it is easy to determine such characteristics and obtain knowledge of the collision domains. In a dense network, this could be a problem. While we do not address this problem, there have been research attempts to provide location information of sensor nodes. For routing purposes, nodes need to determine what their neighbors are and the number of hops required to reach a sensor node can provide us with equivalent information.

Finally, it is worth pointing out that our framework is not limited to tree-like data patterns, but is also capable of capturing broader data dissemination paradigms such as *wave scheduling* [8].

7 Conclusions

We introduced a novel algebraic framework for specifying and analyzing data transmissions along with constraints imposed by a query in wireless sensor networks. Our framework enables flexible cross-layer query optimization techniques that utilize information about the MAC layer. The query optimization results in reduction in energy consumption, which increases the lifetime and effectiveness of the network, to produce the expected Quality of Data in a timely manner. We also introduced the necessary infrastructure, a pervasive catalog that provides our framework with highly available and accurate query statistics and relevant network meta-data.

Currently we are undertaking a comprehensive experimental and theoretical study of our framework. It includes the implementation and testing of our framework in simulated and real-world settings, as well as exploring its completeness and complexity characteristics.

Acknowledgments

We would like to thank our students Mohamed Sharaf, Divyasheel Sharma and Chih-kuang Lin as well as the anonymous reviewers for their thoughtful comments. Special thanks to Alexandros Labrinidis whose constructive and insightful suggestions significantly improved the DTA framework.

References

[1] Atheros Communications. *Whitepaper: 802.11 Wireless LAN Performance.* (available at http://atheros.com/), April 2003.

[2] P. Bonnet, J. Gehrke and P. Seshadri. Towards Sensor Database Systems. *Proc. of MDM Conf.*, 2001

[3] J. Beaver, M. A. Sharaf, A. Labrinidis, and P. K. Chrysanthis. Location-Aware Routing for Data Aggregation for Sensor Networks. *Proc. of Geo Sensor Networks Workshop*, 2003

[4] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. SPAN: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. Proc. of ACM MobiCom Conf., 2001

[5] J. Considine, F. Li, G. Kollios and J. Byers. Approximate Aggregation Techniques for Sensor Databases. *Proc. of IEEE ICDE Conf.*, 2004

[6] U. Cetintemel, A. Flinders, Y. Sun. Power-Efficient Data Dissemination in Wireless Sensor Networks. *Proc. of ACM MobiDE Workshop*, 2003

[7] P. K. Chrysanthis and V. Zadorozhny. From Location Databases to Pervasive Catalog. *Proc. of MDDS Workshop*, 2002

[8] A. Demers, J. Gehrke, R. Rajaraman, N. Trigoni and Y. Yao. Energy-Efficient Data Management for Sensor Networks: A Work-In-Progress Report. *Proc. of 2nd IEEE Upstate New York Workshop on Sensor Networks*, 2003.

[9] Firetide Inc. *Specifications of the HotPoint 1000S Wireless Mesh Router, Datasheet.* (available at: http://www.firetide.com/images/User_FilesImages/documents/HP1000S_DS_a104.pdf)

[10] S. Goel and T. Imielinski. Prediction-based monitoring in sensor networks: Taking lessons from MPEG. *Computer Comm. Review*, 31(5), 2001.

[11] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. *Proc. of HICSS Conf.*, 2000

[12] J. Heidemann, F. Silva, C. Intanagonwiwat, R.Govindan, D. Estrin and D. Ganesan. Building efficient wireless sensor networks with low-level naming. Proc. of ACM SOSP, 2001

[13] Y. E. Ioannidis and Y. Kang. Randomized algorithms for optimizing large join queries. *Proc. of ACM SIGMOD Conf.*, 1990

[14] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen. A Survey of Energy Efficient Network Protocols for Wireless Networks. *Wireless Networks*, 7(4), 2001

[15] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong. TAG: A tiny aggregation service for ad hoc sensor networks. *Proc. of OSDI*, 2002

[16] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong. The Design of an Acquisitional Query Processor for Sensor Networks. *Proc. of ACM SIGMOD Conf.*, 2003

[17] www.opnet.com

[18] K. Pahlavan and A. Levesque. *Wireless Information Networks.* John Wiley and Sons, 1995

[19] K. Pahlavan and P. Krishnamurthy. *Principles of Wireless Networks: A Unified Approach.* Prentice Hall, 2002

[20] J. Proakis. *Digital Communications.* McGraw Hill, 2001

[21] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis. TiNA: A Scheme for Temporal Coherency-Aware in-Network Aggregation. *Proc. of ACM MobiDE Workshop*, 2003

[22] C. Schurgers, V. Tsiatsis and M. Srivastava. STEM: Topology Management for Energy Efficient Sensor Network. *Prov. of IEEE Aerospace Conf.*, 2002

[23] Y.Yao and J.E. Gehrke. The Cougar approach to in-network query processing in sensor networks. *SIGMOD Record*, 31(3), 2002

[24] W. Ye, J. Heidemann and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. *Proc. of IEEE INFOCOM*, 2002

[25] M. Younis, M. Youssef and K. Arisha. Energy-aware routing in cluster-based sensor networks. *Proc. of MASCOTS*, 2002

[26] V. Zadorozhny and P. K. Chrysanthis. Location-Based Computing. In *Telegeoinformatics: Location-Based Computing and Services*, Taylor and Francis Books, 2003

[27] R. Zheng, J. Hou and L. Sha. Asynchronous Wakeup for Ad Hoc Networks: Theory and Protocol Design. *Proc. of ACM MobiHoc*, 2003

[28] R. Zheng and R. Kravets. On-demand Power Management for Ad-Hoc Networks. *Proc. of IEEE INFOCOM Conf.*, 2003

[29] IEEE Std 802.15.4. *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs).* IEEE Computer Society, October 2003