8-2009

Link Scheduling for Scalable Data Aggregation

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

Majid Sarrafzadeh
University of California, Los Angeles

Link Scheduling for Scalable Data Aggregation

Jamie Macbeth
Department of
Computer Science
University of California
Los Angeles, California 90095
Email: macbeth@cs.ucla.edu

Majid Sarrafzadeh
Department of
Computer Science
University of California
Los Angeles, California 90095
Email: majid@cs.ucla.edu

***Abstract*—We explore the link scheduling optimization problem in the context of scalable in-network data aggregation, extending results for broadcast networks to routing in general networks. The primary vehicle for resource preservation is transmission suppression. For certain types of queries, nodes can avoid transmitting records if they can locally infer that their data is not needed to execute the query. We introduce a novel protocol paradigm for *duplicate-insensitive exemplary monotonic* (e.g. `MIN` and `MAX`) data aggregation queries.**

Performance of query execution in these networks is measured through collective expected number of transmissions in the network, and is linked to the minimum and maximum connectivity (δ, Δ) of nodes in general graphs. Nodes can reduce transmissions and save power by forcibly broadcasting partial results to the network during data collection. An algorithm running in time $O(nm)$ is presented which achieves $O((n \ln n \ln \Delta)/\delta)$ expected transmissions.

***Index Terms*—In-network data aggregation, link scheduling, wireless sensor networks.**

I. Introduction

Sensor networks and distributed computing systems are often starved for communication bandwidth and energy resources as they perform tasks and collect data. Even when they are not, there is a regular impetus to attempt to conserve these resources for efficiency. These systems require short delays, reduced power consumption and reduced network and interconnect bandwidth use.

In-network data aggregation attempts to preserve resources and reduce delays in distributed computing systems by performing part of the computation locally to reduce communication between nodes. Data aggregation techniques explore a key trade-off; should we use the interconnect to relay more data to a central location for processing, or should we use local node computing resources to perform some simplification before data transport?

We are most interested in scalable types of distributed computations and data queries; they run tasks and collect data on all nodes, but, on average, only require a small number of nodes to transmit data or results to the node that issued the query. Specifically, we study the communication patterns of scalable data aggregation using a protocol model, and attempt to minimize interconnect bandwidth usage by solving a link scheduling problem [1] [2]. Data aggregation is performed when nodes transmit and relay messages multiple hops to a base station node according to the link schedule, and conserve resources by avoiding transmissions.

This paper is organized as follows: Section II discusses related work. Section III discusses the network model, data model, and aggregation query model with the aid of simple examples. In Sections IV and V the link scheduling data aggregation optimization problem is posed. Section VI discusses our proposed link scheduling algorithm and its performance, and future work is discussed in the final section.

II. Related Work

In TAG [3], a declarative language similar to SQL is developed as a method of issuing queries for sensor network data. Also, a taxonomy is introduced that can be used to describe queries and their qualities. Although our results are independent of the exact programming interface to the distributed system, we use a SQL-like language and the TAG query taxonomy as a means to describe the types of aggregation queries and processes under consideration. TAG has a tree-based routing framework that is sensitive to transmission failures. Cougar [4] is a similar declarative programming based system for sensor data aggregation. In contrast to TAG and Cougar, we explore scalability and optimal conservation of network resources for particular types of queries.

Other work has been done on reducing communication costs in distributed computing systems for methods of data acquisition. Fasolo, Rossi, Widmer and Zorzi [5] have an excellent survey of data aggregation for wireless sensor networks. Jain, Chang and Wang [6] consider data collection as a multiple stream management problem and Kalman filters as a filtering solution for conserving resources. Deshpande, Guestring, Hong and Madden [7] exploit cost disparities in correlated data attributes in multi-predicate query planning. In [8] the authors exploit both the broadcast nature of node transmissions and spatial correlations of nodes for energy efficiency. In [9], scalable in-network data aggregation is considered, but only in the medium access control domain.

III. Network and Query Model

We assume a protocol communication model in which stations are represented by nodes in an undirected simple graph $G = (V, E)$. Two nodes a and b share an edge $(a, b) \in E$ if they are within the fixed transmission range. The set of nodes

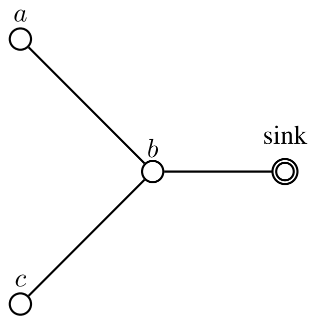

Fig. 1. Nodes may required to forward records in data aggregation, but they may be able to transmit short summaries instead of all data.

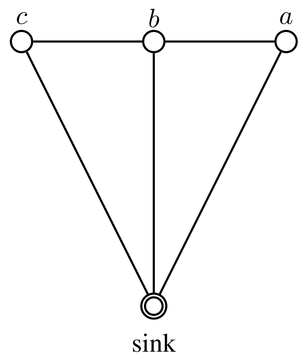

Fig. 2. Each node can listen to the others' transmissions and possibly choose not to transmit at all.

with a link to a is denoted $N(a)$ and is called the neighborhood of a. All transmissions by a node are locally broadcast to its neighbors and all nodes use a single channel or interconnect. For simplicity we assume that all transmissions are successful–no retransmissions or ACKs are required.

In the network, ordinary nodes are called *data nodes* and a special node called the *sink* or *base station node* collects data from the rest of the nodes in the network. A connected undirected simple graph $G = (V, E)$ with a special node s labeled as the sink, $\mathscr{A} = (V, E, s)$, will be called a *data aggregation network*.

Data collection occurs when data nodes, either directly or via forwarding, transmit data records to the sink node. In sensor network systems for data aggregation, queries are issued explicitly using a declarative language and distributed over the network. In other systems, queries may be implicit in the programming of individual nodes. We disregard the problem of *query distribution* here and focus on the *collection* phase, assuming that all nodes have already received the query packet. We say that data nodes respond to the query by sending or forwarding *data records*.

A. Examples

A few simple examples illustrate the possible savings in total power, network bandwidth and delay if proper link scheduling reduces the average number of node transmissions. Fig. 1 depicts a network in which data must be forwarded to the sink node; data records from nodes a and c must pass through node b to reach the sink, s. For some data queries, node b can listen to a and c and forward a summary of the received data.

In Fig. 2, assume that the query seeks for the extremum of the records stored at nodes a, b, and c. This is much unlike our previous example in which data from two nodes had to be forwarded to the sink, and thus all nodes were required to transmit. In this case, because all three nodes can directly reach the sink node, some of them may be able to avoid transmitting altogether while the record needed to satisfy the query is still received successfully by the sink. Each node in this example can listen to the others' transmissions and possibly choose not to transmit its own record.

B. Query Taxonomy

For many applications and data aggregation operations the data nodes are not required to send all data to the sink and summary or transmission suppression is possible. In our discussion, we refer to the TAG query taxonomy [3]. TAG uses a declarative query language similar to SQL to form and execute queries. We use similar SQL-like operators here.

In this study we focus on query response for *duplicate-insensitive exemplary monotonic* aggregation queries. These are best exemplified by `MAX` and `MIN` operators in SQL-like query languages, and we hope to exploit their data redundancy and nonnecessity when we design the link schedule. Without loss of generality, in later discussions we refer to `MAX` in lieu of both `MAX` and its dual, `MIN`.

C. Data Model

We assume that just before each data aggregation query is sent, each node's record value is chosen independently and identically distributed from a set R with a total order, denoted by $\geq$. Each record value is chosen to be unique, and the node record values are represented by a relation, $r : V \to R$, which is one-to-one. For every possible choice of r, a unique extremum record exists, and each node is equally likely to have the extremum record.

By symmetry, the probability that a certain node $\alpha \in V$ holds the extremum record is $P[r(\alpha) > r(x), \forall x \in V] = 1/|V|$. For any subset $S \subseteq V$ of nodes, each node is equally likely to be have the extreme record, and the probability that a certain node $\alpha \in S$ holds the extremum record is $P[r(\alpha) > r(x), \forall x \in S] = 1/|S|$.

D. Effective Forwarding

Assume that a node α has received a set of records from the network that were transmitted to it by neighboring nodes in response to a `MAX` query. α can avoid transmitting if its record value is less than some value that it has received, but must transmit if its record value is higher in the total order than all other values of which it is aware.

However, α's neighbors may have received records transmitted to them that weren't transmitted to α. Assume that α's neighbor β receives a transmission from a third node, γ, and α and γ do not share an edge. If β transmits its record, $r(\beta)$, then

$r(\beta) > r(\gamma)$. If α transmits after receiving β's transmission, $r(\alpha) > r(\beta)$, and, transitively, $r(\alpha) > r(\gamma)$. γ's record is *effectively forwarded* by β's transmission. The set of records that have been directly transmitted or effectively forwarded to a node α are called the *local effective aggregate set* of α.

IV. Transmission Designations

A. Transmission Schedules

Given a data aggregation network $\mathscr{A} = (V, E, s)$, a *transmission schedule* or *link schedule* is a relation $\sigma : V \to \mathbf{N}$ of nodes to TDMA-like transmission slots. In response to a `MAX` query, it is assumed that all nodes will summarize and transmit a single record in any scheduled transmission slot, and that all records are identical in size. Correspondingly, the slot size is the time needed to transmit a single record.

When convenient, the schedule is expressed as a sequence $\sigma = \langle v_1, v_2, v_3, ...v_k\rangle$ where $v_i \in V$. All nodes, including the sink node, can be scheduled to transmit. Nodes can appear repeatedly in the schedule where $v_i = v_j$ for some $i \neq j$. Simplifying the formulation, only one node is allowed to transmit at any given time in the schedule. Aggregated data flows through the network given transmissions made according to the schedule.

B. Transmission Rules

In the second example, we would like to save nodes the trouble of transmitting. However, in the first example we see that nodes need to be forced to transmit under certain circumstances in order for the final result to get to the sink. These cases illustrate why we need transmission rules.

The transmission rules govern whether or not and what a node should transmit based on information gained from transmissions made earlier in the schedule by other nodes. We assign each node one of two transmission rules:

1) **The Always-Transmit (ATX) rule:** Nodes always transmit an extremum of the local effective aggregate set–its own record value and the record values collected from neighboring nodes–independent of its own calculated or measured data.
2) **The Transmit-If-Best (TXB) rule:** Nodes transmit their own record value only if their record value is an extremum of the local effective aggregate set.

The Transmit-If-Best rule is a simple mechanism for nodes to avoid transmitting, while the ATX rule can be used for nodes that are required to forward records towards the sink. We specify transmission rules in an aggregation network as the set of always transmit nodes. Given a data aggregation network $\mathscr{A}$, a subset of the nodes $\tau \subseteq V$ is an *ATX set* for $\mathscr{A}$. A pair consisting of a schedule and an ATX set $\mathscr{D} = (\sigma, \tau)$ is a *transmission designation* for $\mathscr{A}$.

C. Repeatedly Scheduled Nodes

We add an additional, universal transmission rule to accommodate for nodes that may be scheduled to transmit multiple times.

1) **The Never-Repeat-Same-Record rule:** Nodes in ATX mode that appear repeatedly in the schedule do not transmit the same record twice. Nodes may only transmit repeatedly if the extremum of the local effective aggregate set has changed since their last scheduled transmission.

Nodes in transmit-if-best mode only transmit the first time they are scheduled, and only when their record is the extremum of the local effective aggregate set.

D. Statistics of Transmission Designation

Given a data aggregation network $\mathscr{A}$ and a transmission designation $\mathscr{D}$, let T_v be a random variable indicating the number of times that node v transmits during the execution of the schedule. The probability that v transmits k times is $P[T_v = k]$. If node v is only appears once (or not at all) in the schedule, let T_v be a shorthand for $T_v = 1$ and T_v^c stand for $T_v = 0$.

A random variable $\hat{T} = \sum_v T_v$ represents the combined number of transmissions performed by all nodes during the course of the schedule. The expectation value $E[\hat{T}]$ is taken over all possible orderings of record values chosen for the nodes.

Let's consider the designation $\mathscr{D} = (\langle b, a, c\rangle, \emptyset)$ for the example in Fig. 2. Node b transmits with 100% probability, and now nodes a and c, both receiving b's broadcast, each transmit with probability 1/2. Here $E[\hat{T}] = 2$.

E. Feasibility

We require that, for a `MAX` query, the extreme record always arrives at the sink after a single execution of the transmission schedule. This must hold for every possible ordering of records stored at the data nodes. We call this condition *feasibility*.

Formally, given a data aggregation network and a transmission designation, a node a is a *feasible node* if it appears in the schedule and:

1) There is an edge from a to the sink node, or:
2) If there is an edge from a to some ATX feasible node b which is scheduled to transmit later than a.

A transmission designation is called *feasible* if all nodes in G are feasible under the designation. A node is said to be feasible at slot i if it has a feasible transmission scheduled at some slot $j > i$.

Lemma 1. Given a data aggregation network $\mathscr{A} = (V, E, s)$ and a transmission designation $\mathscr{D} = (\sigma, \tau)$, if a feasible node $a \in \tau$ receives the extreme record, $r(v)$ such that $r(v) \geq r(u), \forall u \in V$, it will eventually be forwarded to the sink.

Proof: The theorem holds for the node a in the last scheduled slot; if $a \in N(s)$, a is feasible, and $a \in \tau$, it will transmit the extreme record directly to the sink if it receives it.

Assume that the theorem holds for all feasible transmissions scheduled after slot i. Let a be the node scheduled to transmit in slot i. If $a \in \tau$ and a is feasible, then, by the definition of feasibility, it must have an edge to some node b that is feasible

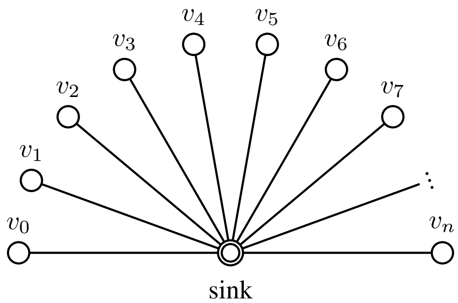

Fig. 3. Repeatedly scheduling a "broadcaster" node to transmit partial results can reduce the number of transmissions overall.

in a later slot. By the transmission rules, if a owns or receives the extreme record, it will transmit that record to be received by b. Therefore the theorem holds for slot $i-1$. By induction, the theorem holds throughout the schedule. ∎

Theorem 1. Given a data aggregation network, $\mathscr{A} = (V, E, s)$, and a feasible transmission designation, $\mathscr{D} = (\sigma, \tau)$, the extreme record is eventually transmitted to the sink.

Proof: If an ATX node owns the extreme record, it is always transmitted to another feasible ATX node, and thus is transmitted to the sink, by Lemma 1. If a TXB node owns the extreme record, by TXB transmission rules it transmits this record locally. Because it is a feasible node, a feasible ATX node receives the record, and it is forwarded to the sink by Lemma 1. ∎

V. Link Scheduling Data Aggregation

Next we pose an optimization problem based on the fact that we can schedule nodes to transmit in a way that will reduce the expected number of transmissions overall.

The Data Aggregation Link Scheduling Optimization Problem. Given a data aggregation network $\mathscr{A} = (V, E, s)$, find a transmission designation $\mathscr{D} = (\sigma, \tau)$ that is feasible and minimizes $E[\hat{T}]$. Expectation is taken over all possible orderings of record values $r : V \to R$ stored at the data nodes.

A. Sink Broadcasting

Consider the data aggregation network given in Fig. 3. In this network all of the non-sink nodes are symmetrically equivalent and none can transmit to each other. As a result, $E[\hat{T}]$ only depends on the scheduling of the sink. For simplicity we assume that the sink has no data record.

We construct a transmission schedule in which the sink node transmits in between the transmissions of data nodes with period $\phi + 1$. The transmission designation is the following:

$$\begin{aligned}\mathscr{D} = (\langle & v_1, v_2, v_3, ..., v_\phi, \text{sink}, \\ & v_{\phi+1}, v_{\phi+2}, ..., v_{2\phi}, \text{sink}, \\ & v_{2\phi+1}, ..., v_n\rangle, \\ & \{\text{sink}\}).\end{aligned}$$

The first group of ϕ nodes in the schedule $\{v_1, v_2, v_3, ..., v_\phi\}$ all transmit with probability 1, since they are unable to hear each other. The sink also transmits with probability 1. Every data node hears the sink's transmission, and $(i-1)\phi$ record values will be effectively forwarded to the ith group of nodes. Each node in the ith group transmits with probability $1/((i-1)\phi + 1)$. There are $\lceil n/\phi \rceil$ such groups. We first calculate the expected number of transmissions for the data nodes:

$$\begin{aligned}E[\sum_{v \neq \text{sink}} T_v] &= \phi + \phi\left(\frac{1}{\phi+1}\right) + \phi\left(\frac{1}{2\phi+1}\right) + ... \\ &\leq \phi\left(1 + \frac{1}{\phi} + \frac{1}{2\phi} + ... + \frac{1}{\lceil n/\phi \rceil - 1}\right) \\ &= \phi + H_{\lceil n/\phi \rceil - 1} \\ &\leq \phi + H_{\lfloor n/\phi \rfloor}\end{aligned}$$

Where H_n denotes the nth harmonic number.

In each of the sink's scheduled transmissions, it only transmits if it has heard a record value higher in the total order since its last transmission (via the never-repeat-same-record rule). The likelihood that this happens in the sink's ith scheduled transmission is the same as the likelihood that any node transmits in the ith group. The probability that any of the ϕ nodes in the ith group has a higher record value than all the nodes in previous groups is $1/i$. The sink is scheduled to transmit after each of $\lfloor n/\phi \rfloor$ groups, and has

$$E[T_{\text{sink}}] = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + ... + \frac{1}{\lfloor n/\phi \rfloor} = H_{\lfloor n/\phi \rfloor}$$

expected transmissions. Combining and using a simple bound (see Havil [10]) for the harmonic number, $H_n < 1 + \ln n$:

$$\begin{aligned}E[\hat{T}] &= E[T_{\text{sink}}] + E[\sum_{v \neq \text{sink}} T_v] \\ &\leq \phi + 2H_{\lfloor n/\phi \rfloor} \\ &< \phi + 2 + 2\ln\left(\frac{n}{\phi}\right).\end{aligned} \quad (1)$$

VI. Constructing Scalable Link Schedules

We now discuss general solutions to the data aggregation link scheduling problem. Initially, we make an observation about the characteristics of of a feasible set of ATX nodes for a graph.

Lemma 2. Given a connected dominating set $C \subseteq V$, a transmission designation can be constructed in time $O(n+m)$ that makes all nodes in C feasible. If σ is any schedule of only C nodes in which nodes are ordered non-increasing in hop distance from the sink, then all nodes in C are feasible under $\mathscr{D} = (\sigma, C)$.

Proof: All nodes in $C \cap N(s)$ (hop distance 0) are feasible by definition. Inductively, assume that all nodes at hop distance i are feasible. All nodes at hop distance $i+1$ are earlier in the schedule than those at hop distance i, and all nodes in the schedule are in the always-transmit set C. Therefore, all nodes at hop distance $i+1$ are feasible. Since C is connected, all nodes in C are feasible. The schedule can be constructed in time $O(n+m)$ by performing a BFS in the subgraph induced by $C \cup s$ starting at s. ∎

Lemma 3. We are given a connected dominating set $C \subseteq V$ and any schedule fragment f. All nodes in f are feasible in the transmission designation $\mathscr{D} = (f', C)$ where f' is a schedule with σ as in Lemma 2 appended to f.

Proof: The schedule σ contains only nodes in the connected dominating set C. All nodes in C are feasible, in ATX mode, and are scheduled after the nodes in f. All nodes in f have an edge to a node in C because C is a dominating set. Thus, all nodes in f are feasible. ∎

A. Scalable Schedules for General Graphs

In this section we present a method for constructing schedules for general graphs. We do this by recreating the sink-broadcaster example. This is done in a way very reminiscent of ad-hoc routing techniques which create a virtual network backbone [11]. Whereas in ad-hoc routing the virtual backbone is used to efficiently compute and update routes, we will use the backbone to broadcast partial results throughout the network.

To do this we first construct a connected dominating set that serves as the basis for an ATX set. A "collector" schedule fragment σ_c schedules the ATX nodes to collect all transmitted records and transmit them to the sink. It is constructed by ordering the transmissions according to hop distance from the sink, with furthest nodes first.

A "broadcaster" schedule fragment σ_b is the reverse of σ_c, and, assuming that the sink has just transmitted the best record value heard so far, it broadcasts this record value to the entire network. We construct a schedule that periodically collects data from ϕ nodes, and broadcasts it to nodes remaining to transmit. An algorithm to generate this schedule is in Fig. 4. It schedules nodes in the following order:

1) ϕ non-ATX nodes which have not yet been scheduled are scheduled to transmit first.
2) The "collector" schedule fragment σ_c is scheduled.
3) The sink is scheduled to transmit, beginning the broadcast.
4) The "broadcaster" schedule fragment σ_d is scheduled.
5) The schedule components in 1-4 are repeated until no non-ATX nodes remain.

B. Expected Number of Transmissions

We conservatively bound the $E[\hat{T}]$ as follows:
In the ith iteration, where $i \geq 1$:

1) ϕ non-ATX nodes each transmit with probability $\leq 1/((i-1)\phi+1)$.
2) $|\tau|$ nodes each transmit with probability $\leq 1/i$ in the collection stage.
3) The sink transmits with probability $\leq 1/i$.
4) $|\tau|$ nodes each transmit with probability $\leq 1/i$ in the broadcast stage.

Input: $\mathscr{A} = (V, E, s)$
Output: A feasible $\mathscr{D} = (\sigma, \tau)$

```
τ ← a connected dominating set of G
Perform BFS of subgraph τ ∪ s starting at s
σ_d ← τ in least hop-distance order
σ_c ← τ in greatest hop-distance order
S ← V − τ − {s}
σ ← ⟨⟩
φ ← 2|τ| + 1
while S ≠ ∅ do
  if |S| > φ then
    Let X ⊂ S, |X| = φ
  else
    Let X = S
  end if
  Append X to σ
  Append σ_c to σ
  S ← S − X
  if S ≠ ∅ then
    Append s to σ
    Append σ_d to σ
  end if
end while
```

Fig. 4. An algorithm to generate a transmission designation to solve the scalable data aggregation link scheduling problem.

$E[\hat{T}]$ can thus be bound to

$$\begin{aligned} E[\hat{T}] &\leq \phi + (1 + 2|\tau| + 1)\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + ...\right) \\ &= \phi + (2|\tau| + 2)H_{\lfloor n/\phi \rfloor}. \end{aligned}$$

It is minimized for $\phi = 2(|\tau| + 1)$ where it is

$$\begin{aligned} E[\hat{T}] &\leq 2(|\tau| + 1)\left(1 + H_{\lfloor n/\phi \rfloor}\right) \\ &\leq \phi\left(2 + \ln\left(\frac{n}{\phi}\right)\right). \end{aligned} \quad (2)$$

C. Algorithms

We make use of the following lemma to bind $E[\hat{T}]$ using the transmission designation given by our algorithm.

Lemma 4. For any connected graph G where the minimum degree of any node is $\delta(G) \geq k$, the connected domination number $\gamma_c(G) \leq 3\lfloor n/(k+1) \rfloor - 2$.

Proof: See [12]. ∎

This gives us a bound on the size of the minimum connected dominating set, $\gamma_c \in O(n/\delta)$. If this set is used to construct τ and, consequently, the schedule fragments σ_c and σ_d, $E[\hat{T}]$ will be bound as well.

Theorem 2. If $\phi = 2(|\tau| + 1)$, $\phi \ll n$, and a minimum connected dominating set is used to construct σ_c and σ_d, $E[\hat{T}]$ can be bound asymptotically to

$$E[\hat{T}] \in O\left(\frac{n \ln n}{\delta}\right). \quad (3)$$

Also, the resulting transmission designation is feasible. If $\delta(n) \in \Omega(n)$ then

$$E[\hat{T}] \in O(\ln n), \tag{4}$$

consistent with the result given in [9] for broadcast networks.

Proof: From (2). The designation is feasible by Lemma 3. Lemma 4 gives the bound on $|\tau|$. ■

The ATX set τ can be calculated by constructing a connected dominating set for the graph G. The connected dominating set problem is NP-Complete [13], but Guha and Khuller [14] give a $\ln \Delta + 3$ approximation algorithm for connected dominating set where Δ is the maximum degree of any node in the graph. This algorithm runs in time $O(nm)$.

The collector and broadcaster schedule segments which use the set τ can be constructed using a BFS from the sink node, running in $O(n+m)$. Non-ATX nodes for each of the iterations can be chosen arbitrarily.

Theorem 3. Using Guha and Khuller's algorithm provides a dominating set (and thus an ATX set) with $|\tau| \in O(\ln \Delta/\delta)$. Assuming that δ and Δ are large and $|\tau|$ remains small, Algorithm 1 constructs a transmission designation with expected transmissions

$$E[\hat{T}] \in O\left(n \ln n \cdot \frac{\ln \Delta}{\delta}\right). \tag{5}$$

Assuming that $\delta(n)$ and $\Delta(n) \in \Omega(n)$:

$$E[\hat{T}] \in O\left(\ln^2 n\right). \tag{6}$$

Proof: The proof follows from (2) and Lemma 4. ■

VII. Conclusion

We have developed a broadcast link scheduling algorithm and a protocol to efficiently route messages for data aggregation tasks. Under time-division-multiplexing medium access, the link schedules transport messages needed to execute queries that have minimal data requirements. Simultaneously, the algorithm generates a set of transmission rules that allows nodes to avoid transmitting unneeded messages based on messages they have received. The transmission rules couple with a link scheduling and routing strategy to attempt to minimize resource usage by reducing unnecessary transmissions throughout the network. A decentralized version of this algorithm may work well for mobile ad-hoc networks.

The current work, while providing a simple framework to understand routing messages in data aggregation, makes a number of unrealistic assumptions and constraints that we would like to relax in future work. Because our transmission schedules only allow a single node to transmit at a time, we don't achieve the same efficiency in time as when there are concurrent transmissions in the network.

If many stations are allowed to transmit concurrently, we require an appreciation of how interference affects successful reception of messages. Dealing with retransmission and acknowledgement are important future considerations. A physical (SINR) interference model [15] should be employed to explore efficient usage of the channel to reduce delays in data aggregation. Use of similar schemes under a random-access CSMA-like medium access control regime should be considered.

Our study assumes that data generated at individual nodes has no structure and that all initial data record distributions are equally probable. Better algorithms will likely produce designations with even less expected transmissions when a structured model for sensor measurements and data exists.

References

[1] R. Cruz and A. Santhanam, "Optimal routing, link scheduling and power control in multihop wireless networks," *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 1, pp. 702–711 vol.1, March-3 April 2003.

[2] B. E. Hajek and G. H. Sasaki, "Link scheduling in polynomial time," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, 1988.

[3] 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.*, vol. 36, no. SI, pp. 131–146, 2002.

[4] Y. Yao and J. Gehrke, "The Cougar approach to in-network query processing in sensor networks," *SIGMOD Rec.*, vol. 31, no. 3, pp. 9–18, 2002.

[5] 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]*, vol. 14, no. 2, pp. 70–87, April 2007.

[6] A. Jain, E. Y. Chang, and Y. F. Wang, "Adaptive stream resource management using kalman filters," in *SIGMOD Conference*, 2004, pp. 11–22.

[7] A. Deshpande, C. Guestrin, W. Hong, and S. Madden, "Exploiting correlated attributes in acquisitional query processing," *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pp. 143–154, April 2005.

[8] L. Wang and A. Deshpande, "Predictive modeling-based data collection in wireless sensor networks," in *EWSN*, ser. Lecture Notes in Computer Science, R. Verdone, Ed., vol. 4913. Springer, 2008, pp. 34–51.

[9] J. Macbeth and M. Sarrafzadeh, "Scalable medium access control for in-network data aggregation," in *DIAL M-POMC '08: Proceedings of the fifth international workshop on Foundations of mobile computing*. New York, NY, USA: ACM, 2008, pp. 13–22.

[10] J. Havil, *Gamma: exploring euler's constant*. Princeton, NJ: Princeton Univ. Press, 2003.

[11] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets," *Communications, 1997. ICC 97 Montreal, 'Towards the Knowledge Millennium'. 1997 IEEE International Conference on*, vol. 1, pp. 376–380 vol.1, Jun 1997.

[12] T. W. Haynes, S. Hedetniemi, and P. Slater, *Fundamentals of Domination in Graphs*. CRC, January 1998.

[13] M. R. Garey and D. S. Johnson, *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co Ltd, January 1979.

[14] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.

[15] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, Mar 2000.