Efficient Data Collection for Wireless Networks:
Delay and Energy Tradeoffs

Chao Ren†, XuFei Mao‡, Xiang-Yang Li*, Ping Xu†, GuoJun Dai∗

Abstract—In this paper, we study efficient data collection in wireless sensor networks. We present efficient distributed algorithms with approximately the minimum delay, or the minimum number of messages to be sent by all nodes, or the minimum total energy costs by all nodes. We analytically prove that all proposed methods are either optimum or within constants factor of the optimum. We then investigate the possibility of designing one universal method such that the delay, the messages sent by nodes, and the total energy costs by all nodes are all optimum or within constants factor of optimum. Given a method $\A$ for data collection let $\varrho_M$, $\varrho_M$, and $\varrho_E$ be the approximation ratios of $\A$ in terms of time complexity, message complexity, and energy complexity respectively. We show that, for data collection, there are networks of $n$ nodes and maximum degree $\Delta$, such that $\varrho_M = \Omega(\Delta)$ for any algorithm.

Index Terms—Time complexity, message complexity, energy, sensor networks, data collection.

I. INTRODUCTION

For wireless sensor networks, the ultimate goal is to collect the data (either the raw data or in-network-processed data or both) from a number of targeted wireless sensors to some sink nodes and then perform some further analysis at sink nodes. Convergecast is the common many-to-one communication pattern used for such sensor network applications. In this paper, we study some fundamental complexity problems for data collection in wireless sensor networks.

Data collection is to collect the set of data items $A_i$ stored in each individual node $v_i$ to the sink node. In this paper, we first design efficient algorithms whose complexity is asymptotically same as (or within a certain factor of) the complexity of the optimum for data collection. The complexity of a problem is defined as the worst case cost (time, message or energy) by the best algorithm. Studying the complexity of a problem is often challenging. Data collection and aggregation has been extensively studied in the community of networking and database. Surprisingly, little is known about the complexity tradeoffs of this operation.

In [10], five distributive aggregations max, min, count, sum and average are carried out efficiently on a spanning tree. Subsequent work did not quite settle the time complexity, the message complexity and the energy complexity of data collection and aggregation, nor the tradeoffs among these three possibly conflicting objectives. The closest results are [7]–[9]. All assume a complete wireless network, which is usually not true in practice. Furthermore, to the best of our knowledge, no fundamental results on the tradeoffs among the time complexity, message complexity, and energy complexity were known before this work.

To the best of our knowledge, we are the first to study the tradeoffs among the message complexity, time complexity, and energy complexity for data collection; we are the first to present lower bounds (and matching upper-bounds for some cases) on the message complexity, time complexity, and energy complexity for data collection in wireless networks. The main contributions of this paper are as follows.

We design algorithms whose time complexity and message complexity are within constant factors of the optimum. The minimum energy data collection can be done using minimum cost shortest path tree. We further show that no data collection algorithm can achieve approximation ratio $\varrho_M$ for message complexity and $\varrho_E$ for energy complexity with $\varrho_M \cdot \varrho_E = o(\Delta)$. We then prove that our data collection algorithm has energy cost within a factor $O(\Delta)$ of the optimum while its time and message complexity are within $O(1)$ of the corresponding optimum. Thus, our method achieves the best tradeoffs among the time complexity, message complexity and energy complexity.

The rest of the paper is organized as follows. In Section II, we first present our wireless sensor network model, define the problems to be studied in this paper, and then briefly review the connected dominating set. We present several efficient methods for data collection in Section IV and we study the complexity tradeoffs of distributed data collection in Section V. We review the related works in Section VI and conclude the paper in Section VII.

II. PRELIMINARIES

A. Network Model

In this paper, we mainly focus on the complexities of data collection in wireless sensor networks. Thus, for simplicity, we assume a simple and yet general enough wireless sensor network model that is widely used in the community. We
assume that there are $n + 1$ wireless sensor nodes $V = \{v_0, v_1, v_2, \cdots, v_n\}$ that are deployed in a certain geographic region, where $v_0$ is the sink node. Each wireless sensor node corresponds to a vertex in a graph $G$ and two vertices are connected iff their corresponding sensor nodes can communicate directly. The graph $G$ is called the communication graph of this sensor network. We assume that links are “reliable”: when a node $v_i$ sends some data to a neighboring node $v_j$, the total message cost is only 1, although in practice node $v_i$ may need re-transmit several times. In some of the results, we further assume that all sensor nodes have a communication range $r$ and a fixed interference range $R = \Theta(r)$. For simplicity, we assume that $r = 1$, i.e., is normalized to one unit. In other words, the communication graph $G$ is a Unit Disk Graph (UDG).

Let $h(v_i, v_j)$ be the hop number of the minimum hop path connecting $v_i$ and $v_j$ in graph $G$, and $D(G)$ be the diameter of the graph, i.e., $D(G) = \max_{v_i, v_j} h(v_i, v_j)$. Here, we assume that $D(G) \geq 2$. If $D(G) = 1$, then the graph $G$ is simply a completed graph and all questions studied in this paper can either be trivial or have been solved [7]–[9]. For a graph $G$, we denote its maximum degree as $\Delta(G)$. When each node $v_i$ has $n_i$ data items, we define the weighted degree, denoted as $\tilde{d}_{v_i}(G)$, of a node $v_i$ in graph $G$ as $n_i + \sum_{v_j, v_k \in \mathcal{G} \cap V} n_j$. The maximum weighted degree of a graph $G$, denoted as $\Delta(G)$, is defined as $\max_{v_i} \tilde{d}_{v_i}(G)$.

Each wireless node is able to monitor the environment, and collect some data (such as temperature). Assume that $A = \{a_1, a_2, \cdots, a_N\}$ is a totally ordered multi-set of $N$ elements collected by all $n$ nodes. Here, $N$ is the cardinality of set $A$. Each node $v_i$ has $n_i$ amount of raw data, denoted as $A_i \subset A$. Since $A$ is a multi-set, $A_i \cap A_j = \emptyset$ for $i \neq j$ and $A = \bigcup_{i=1}^{n} A_i$. Then $\langle A_1, A_2, \cdots, A_n \rangle$ is called a distribution of $A$ at sites of $V$. We assume that one packet (i.e., message) can contain one data item $a_i$, the node ID, plus additional constant number of bits, i.e., the packet size is at the order of $\Theta(\log n + \log U)$, where $U$ is the upper-bound on values of $a_i$. Such a restriction on the message size is realistic and needed, otherwise a single convergecast would suffice to accumulate all data items to the sink which will subsequently solve the problems easily. We consider a TDMA MAC schedule and assume that one time-slot duration allows transmission of exactly one packet.

If energy consumption is to be optimized, we assume that the minimum energy consumption by a node $u$ to send data correctly to a node $v$, denoted as $E(u, v)$, is $c_1 \cdot \|u - v\|^\alpha$, where $c_1$ (normalized to 1 hereafter) and $\alpha \geq 2$ are constants depending on the environment. We assume that each wireless sensor node can dynamically adjust its transmission power to the minimum power needed.

For data queries in WSNs, we often need to build a spanning tree $T$ of the communication graph $G$ first for pushing down queries and propagating back the intermediate results. Given a tree $T$, let $H(T)$ denote the height of the tree, i.e., the number of links of the longest path from root to all leave nodes. The depth of a node $v_i$ in $T$, denoted as $d_T(v_i)$, is the length of the path from the root to $v_i$. The subtree of $T$ rooted at a node $v_i$ is denoted as $T(v_i)$, the parent node of $v_i$ is denoted as $p_T(v_i)$, and the set of children nodes of a node $v_j$ is denoted as $\text{Child}(v_j)$.

### B. Problems and Complexities

We mainly study the time complexity, message complexity, and energy complexity of data collection in wireless sensor networks.

The complexity measures we use to evaluate the performance of a given protocol are worst-case measures. The message complexity (and the energy complexity, respectively), of a protocol is defined as the maximum number of total messages (the total energy used, respectively) by all nodes, over all inputs, i.e., over all possible wireless networks $G$ of $n$ nodes (and possibly with additional requirement of having diameter $D$ and/or maximum nodal degree $\Delta$) and all possible data distributions of $A$ over $V$. The time complexity is defined as the lapsed time from the time when the first message was sent to the time when the last message was received. The lower bound on a complexity measure (e.g., message complexity) is the minimum complexity (e.g., message complexity) required by all protocols that answer the queries correctly. The approximation ratio $q_T$ (resp. $q_M$ and $q_E$) for an algorithm denotes the worse ratio of the time complexity (resp. message complexity and energy consumption) used by this algorithm compared to an optimal solution over all possible problem instances.

Here we assume that TDMA MAC is used for channel usage. Obviously, the complexity depends on the TDMA schedule policy $S$. Let $X(v_i, t)$ denote whether node $v_i$ will transmit at time slot $t$ or not. Then a TDMA schedule policy $S$ is to assign 0 or 1 to each variable $X(v_i, t)$. A TDMA schedule should be interference free: no receiving node is within the interference range of the other transmitting node. In other words, if the schedule is define for tree $T$, for any time slot $t$, if $X(v_i, t) = 1$, then $X(v_j, t) \neq 1$ for any node $v_j$ such that $p_T(v_i)$ is within the interference range of $v_j$.

Data collection is an operation to collect the set of raw data items $A$ from all sensor nodes to the sink node. It can be done by building a spanning tree $T$ rooted at the sink $v_0$, and sending the data at every node $v_i$ to the root node along the unique path in the tree. Clearly, the message complexity of data collection along $T$ is $\sum_{i=1}^{n} n_i \cdot d_T(v_i)$. The energy complexity, defined as the total energy needed by all nodes for completing an operation, of data collection using $T$ is $\sum_{i=1}^{n} \left[ E(v_i, p_T(v_i)) \right] \cdot \sum_{v_j \in T(v_i)} n_j$.

The TDMA schedule should also be valid in the sense that every datum in the network will be relayed to the root. In other words, in tree $T$, when node $v_i$ sends a datum to its parent $p_T(v_i)$ at the time slot $t$, node $p_T(v_i)$ should relay this datum at some time-slot $t' > t$. The largest time $D$ such that there exists a node $v_i$ with $X(v_i, D) = 1$ is called the time complexity of this valid schedule. Time $D$ is also called the mark-span of the schedule $S$. Generally, a schedule $S$ can be defined as assigning 0 or 1 to variable $X(v_i, t, k)$: it is 1 if and only if node $v_i$ will relay datum $a_k$ at time slot $t$. Clearly, a schedule $S$ is valid for data collection of $A$ using tree $T$, iff for every node $v_i$ and time slot $t$, $\sum_{v_j \in \text{Child}(v_i)} \sum_{b=1}^{T(v_j, b) +}$
\[ n_i \geq \sum_{b=1}^{t} X(v_i, b). \] Here \( \sum_{v_j \in \text{Child}(v_i)} \sum_{b=1}^{t-1} X(v_j, b) + n_i \)

is the total number of data items node \( v_i \) has seen so far till time slot \( t \) and \( \sum_{b=1}^{t} X(v_i, b) \) is the total number of data items that have been relayed by node \( v_i \) so far till time slot \( t \). Then the time-complexity optimizing data collection problem is to find a spanning tree \( T \) and a valid, interference-free schedule \( S \) such that the mark-span is minimized.

In this paper, we study the complexity and efficient algorithm for data collection in wireless sensor networks. To address each of these problems, we usually first build a spanning tree \( T \) and then decide a interference-free and valid schedule of nodes activities such that certain complexity measure is optimized. However, our lower bound and approximation argument do not depend on the communication graph used, which may not be a tree.

### III. CONNECTED DOMINATING SET

A number of our methods will be based on a “good” connected dominating set (CDS) that has bounded degree \( d \) and bounded hop spanning ratio. Here a subgraph \( H \) of \( G \) is a connected dominating set if (1) graph \( H \) is connected, and (2) the set of vertices of \( H \) is a dominating set, i.e., for every node \( v \in G \setminus H \), there is a neighboring node \( u \in H \), i.e., \( uv \in G \). Notice that in this paper we treat CDS as a graph which includes the edges between dominators and connectors in CDS. A subgraph \( H \) of \( G \) has a bounded spanning ratio if for every pair of nodes \( u \) and \( v \) in \( H \), the distance (hop or weighted distance) of the shortest path connecting \( u \) and \( v \) in \( H \) is at most a constant times of the distance of the shortest path connecting them in graph \( G \).

A number of methods have been proposed in the literature to construct such a good CDS. See [1], [2] for more details. A simple method is to partition the deployment region into grid of size \( r/\sqrt{2} \), select a node (called dominator) from each cell if there is any, and then find nodes (called connectors) to connect every pair of dominators that are at most 3-hops apart. Then the diameter of the CDS is at most a constant times of the original diameter of graph \( G \). The complexity of building a good CDS at each node is only \( O(d \log d) \), where \( d \) is the number of neighbors of that node. We assume the availability of a good CDS hereafter.

Given a graph \( G = (V, E) \), let \( C = (V_C, E_C) \) be a connected dominating set of \( G \) where \( V_C \) is the set of dominators and connectors and \( E_C \) is the edges between dominators and connectors. For a node \( v \in V_C \), let \( T_C \) be a BFS tree of \( C \). For a node \( v \in V \setminus V_C \), we define a unique dominator \( d(v) \) which is the one having the shortest hop distance to the sink \( v_0 \).

**Definition 1 (Data Communication Tree (DCT)):** For a graph \( G \) and its CDS, the data communication tree \( T \) is defined as follows: \( T = (V, T_C \cup \{ vd(v) \mid v \in V \setminus V_C \}) \).

Given a data communication tree, an aggregate operation consists of (possibly repeated) two phases: a propagation phase where the query demands are pushed down into the sensor network along the tree; and an aggregation phase where the aggregated values are propagated up from the children to their parents. We discuss some properties of the data communication tree.

**Theorem 1:** Let \( G \) and \( C \) be a graph and its CDS respectively. The data communication tree \( T \) has following properties:

1) \( \Delta(T_C) \leq d \).
2) For any edge \( e \in E_T \), let \( I(e) \) be the set of edges in \( T_C \) that have interferences with \( e \), then \( |I(e)| \leq c \cdot d \cdot \Delta(G) \) for some constant \( c \) depending on \( R/r \).

**Proof:** The first property directly comes from the property of the CDS \( C \). For any edge \( e = uv \in E_T \), either \( u \) or \( v \) will be in \( C \) based on our construction. Assume \( u \in V(C) \). For all edges having interferences with \( e \), both end nodes should be within distance \( 2r + R \) from \( u \). Since \( R = \Theta(r) \) and the CDS has a constant-bounded degree, there are at most a constant number of nodes of \( C \) within this range. On the other hand, all edges of \( T \) have at least one node in \( C \). Then it is easy to show that \( |I(e)| \leq (R+2r)^2 \cdot d \cdot \Delta(G) \) by an area argument.

All our methods will be based on a good CDS and using data clustering: given a good CDS, for a node \( v \in V \setminus V_C \), it sends the data items to its dominator \( d(v) \) in a TDMA manner.

**Lemma 2:** Given a good CDS of the graph \( G \), data clustering can be done in time \( O(\tilde{\Delta}(G)) \).

**Proof:** We use the communication tree \( T \) to do data clustering. For a node \( v \in V \setminus V_C \), assume that the edge \( vd(v) \) interferes with an edge \( ud(u) \). Then dominator nodes \( d(u) \) and \( d(v) \) are within distance at most \( R + 2r \). Thus, there are at most \( (R+2r)^2 \cdot d \cdot \Delta(G) \) such dominator nodes. Thus, the total number of data items of all nodes \( u \) such that \( ud(u) \) interferes with \( vd(v) \) is at most \( (R+2r)^2 \cdot d \cdot \Delta(G) = \Theta(\tilde{\Delta}(G)) \). Hence, every such edge \( vd(v) \) can be scheduled to transmit \( n_i \) times in \( \Theta(\tilde{\Delta}(G)) \) time-slots using a simple greedy coloring method that colors the nodes sequentially using the smallest available color.

After data clustering, all data elements are clustered in \( T_C \). In other words, each node \( v_i \) in the connected dominating set now will have data from all nodes dominated by \( v_i \). The data clustering asymptotically does not incur additional cost for time complexity and message complexity when \( n_i = O(1) \). Notice that the total number of messages for data clustering is \( \sum_{v_i \in V_C} n_i \).

### IV. EFFICIENT DATA COLLECTION

In this section, we design efficient methods for collecting data in wireless sensor networks.

**A. Minimize Message**

We first study the data collection with the minimum number of messages. When all links are reliable, i.e., we only need one message to send a packet from a node \( u \) to a node \( v \) over a link \((u, v)\), we should collect any source data from a source node \( v_i \) to the sink node \( v_0 \) over the path with the minimum number of relay nodes, i.e., with minimum hop number. Thus,

**Theorem 3:** Data collection can be done with minimum number of messages \( \sum_{i=1}^{n} n_i \cdot h(v_i, v_0) \) using a Breadth First Search (BFS) tree with root \( v_0 \) if all links are reliable.
B. Minimize Energy

We then study the data collection with the minimum energy cost. Apparently, for any element, it should follow the minimum energy cost path from its origin to the sink node $v_0$ in order to minimize the energy consumption. So minimizing the energy is equivalent to the problem of finding the shortest paths from the sink to all nodes (where the link cost is the its energy cost), which clearly can be done in time $O((m+n \log n)$ for a communication graph of $n$ nodes and $m$ links. We call the tree formed by minimum energy path from the root to all nodes as the minimum energy path tree (MEPT). Let $E(v_i, v_0)$ be the energy cost of the path from $v_i$ to $v_0$ with the minimum energy cost. Thus, we have

Theorem 4: Data collection can be done with minimum energy cost $\sum^n_{i=1} n_i \cdot E(v_i, v_0)$ using a MEPT tree with root $v_0$ if a link $(u, v)$ has an energy cost $E(u, v)$.

Clearly, when links are not reliable, we have to take into account the energy cost in retractions. In other words, we need use $E(u, v)/p(u, v)$ as the expected energy cost of a link $(u, v)$.

C. Minimize Time Delay

Then we study the time complexity of data collection. Notice that, the transmissions of nearby nodes should be in different time slots to avoid the interferences. We assume that all links are reliable hereafter.

Algorithm 1 presents our efficient data collection method based on the connected dominating set C. The constructed CDS has a degree at most a constant $d$, and similar to Theorem 1, all nodes in CDS can be scheduled to transmit once in constant $\beta = \Theta(d)$ time-slots without causing interferences to other nodes in CDS. We take $\beta$ time-slots as one round.

First, the data elements from each dominatee node (a node not in C) are collected to the corresponding dominator node in the connected dominating set C. Here the dominatee nodes that are one hop away from the sink node $v_0$ will directly send the data to $v_0$. Notice that this can be done in time-slots $O(\Delta(G))$ which is proved in Lemma 2.

Now we only consider the dominator nodes and the breadth-first-search spanning tree $T_C$ of nodes in CDS rooted at the sink $v_0$. Every edge in the tree $T_C$ will be scheduled exactly once in each round. For simplicity, we do not schedule sending an element more than once in the same round. At any round, nodes in CDS push one datum item to its parent node until all data are received by $v_0$.

Algorithm 1 Efficient Data Collection Using CDS

Input: A CDS C with bounded degree $d$, tree $T_C$.

1: Every node $v_i$ sends its data to its dominator node $d(v_i)$.
2: for $t=1$ to $N$ do
3: for each node $v_i \in V_C$ do
4: If node $v_i$ has data not forwarded to its parent node in $T_C$, node $v_i$ sends a new data to its parent in round $t$.

Theorem 5: Given a connected wireless network $G$, data collection can be done in time $\Theta(N)$ with $\Theta(\sum^n_{i=1} n_i h(v_i, v_0))$ messages.

Proof: From Lemma 2, in $O(\Delta(G))$ time-slots, the data elements from each dominatee node are collected to the corresponding dominator node in the connected dominating set. We show that after $N + H(T_C)$ rounds, all elements can be scheduled to arrive in the root, where $H(T_C)$ is the height of the BFS tree $T_C$. Algorithm 1 illustrates our method to achieve this.

A CDS node $v$ is in level $i$ if the path from $v$ to $v_0$ in BFS tree $T_C$ has $i$ hops. A level $i$ is said to be occupied at a time instance if there exists one CDS node from level $i$ that has at least one data. Assume that originally all levels $i \in [1, H(T_C)]$ are occupied, after collecting data from all dominatee nodes.

We will show that each round the root will get at least one data item if there are data in the network. We essentially will show that the occupied levels are continuous, i.e., before each round $t$, there is $L_t$ such that all levels in $[1, L_t]$ are occupied and levels in $[L_t+1, H(T_C)]$ are not. We prove this by induction. This is clearly true for round 1. Assume that it is true for round $t$. Then in round $t$, for each level $i \in [1, L_t-1]$, every node in level $i+1$ will send its data to its parent in level $i$. Then every level $i \in [1, L_t-1]$ will have data for sure before round $t+1$. Then clearly, $L_{t+1} = L_t$ if some nodes in level $L_t$ still have some data; otherwise we set $L_{t+1} = L_t - 1$. Consequently, root will get at least one data item for each round whenever there are data in the network. Since there are at most $N$ data items, Algorithm 1 will take at most $N$ rounds, i.e., $O(N)$ time-slots because each round is composed of constant $\beta$ time-slots.

When not all levels are occupied initially, then it is easy to show that after at most $H(T_C)$ rounds, the occupied levels will be continuous. Hence, the collection can be done in at most $N + H(T_C)$ rounds. Notice that $H(T_C) = \Theta(D(G))$. Consequently, the total time-slots are at most $O(\Delta(G)) + O(N + D) = O(N)$ since $\Delta(G) \leq N$.

On the other hand, for any data collection algorithm, it needs at least $N$ time slots since the sink can only receive one data item in one time slot and there are $N$ data items.

The total number of messages used by the algorithm is of course at most $4 \sum^n_{i=1} n_i h(v_i, v_0)$ as the element at node $v_i$ is relayed by at most $4 \times h(v_i, v_0)$ nodes in CDS (since $h(v_i, v_0) \geq 2$). Obviously any algorithm needs at least $\sum^n_{i=1} n_i h(v_i, v_0)$ messages.

V. COMPLEXITY TRADEOFFS FOR DATA COLLECTION

One may want to design a universal data collection method whose time-complexity, message-complexity and energy-complexity are all within constant factors of the optimum. Observe that Algorithm 1 is a constant approximation for both time-complexity and message-complexity. However, it is not a constant approximation for energy-complexity. Consider the following line network example: $n+1$ nodes are uniformly distributed in a line segment $[0, r]$; Sink $v_0$ is the leftmost node and node $v_i$ is at position $i \cdot r/n$ and has one data item. Here we assume $r=1$. See Figure 1 for illustration. Assume the energy cost for a link $uv$ is $||uv||^2$. Then the minimum cost
data collection is to let node \( v_i \) send all its data to node \( v_{i-1} \). The total energy cost is \( \sum_{i=1}^{n} i \cdot \frac{1}{n^2} \approx 1/2 \). While the energy cost of collecting data via CDS is \( \sum_{i=1}^{n} (\frac{i}{n})^2 \approx n/6 \). On the other hand, the total number of messages of the minimum-energy data collection scheme is \( n(n-1)/2 \) and the time slots used by this scheme is also \( \Theta(n^2) \); both of which are \( \Theta(n) \) times of the corresponding minimum.

Consider any data collecting algorithm \( A \). Let \( q_M \) and \( q_E \) be the approximation ratio for the message-complexity and energy-complexity of algorithm \( A \). We show that there are graphs of \( n \) nodes such that \( q_M \cdot q_E = \Theta(n) \).

**Lemma 6:** Assume the energy cost for supporting a link \( uv \) is \( ||uv||^2 \). For any data collection algorithm \( A \), there are graphs of \( n \) nodes, such that \( q_M \cdot q_E = \Omega(n) \).

**Proof:** Consider the line graph example defined previously. For a node \( v_i \), assume that the data collection path is composed of \( k_i \) hops and the length of the \( k_i \) links are \( x_{i,1}, x_{i,2}, \ldots, x_{i,k_i} \). Then \( \sum_{j=1}^{k_i} x_{i,j} = \frac{i}{n} \). The total energy cost, denoted as \( e_i \), of such data collection path is \( e_i = \sum_{j=1}^{k_i} x_{i,j}^2 \geq (\frac{\sum_{j=1}^{\lceil k_i \rceil} x_{i,j}}{k_i})^2 \).

Thus,

\[ e_i \cdot k_i \geq \left( \frac{i}{n} \right)^2. \]

Obviously, the total number of messages is \( \sum_{i=1}^{n} k_i \) and the total energy cost is \( \sum_{i=1}^{n} e_i \). We will use the Hölder’s inequality: for positive \( a_i \) and \( b_i, p > 0, q > 0 \) with \( \frac{1}{p} + \frac{1}{q} = 1 \), we have

\[ \left( \sum_{i=1}^{n} a_i^p \right)^{\frac{1}{p}} \left( \sum_{i=1}^{n} b_i^q \right)^{\frac{1}{q}} \geq \left( \sum_{i=1}^{n} a_i \cdot b_i \right). \]

Equivalently, \( \left( \sum_{i=1}^{n} a_i \right)^{\frac{1}{p}} \left( \sum_{i=1}^{n} b_i \right)^{\frac{1}{q}} \geq \left( \sum_{i=1}^{n} a_i^p \right)^{\frac{1}{p}} \left( \sum_{i=1}^{n} b_i^q \right)^{\frac{1}{q}}. \) Then

\[ \left( \sum_{i=1}^{n} k_i \right) \left( \sum_{i=1}^{n} e_i \right) \geq \left( \sum_{i=1}^{n} \sqrt{e_i} \cdot \sqrt{k_i} \right)^2 \geq \left( \sum_{i=1}^{n} \frac{i}{n} \right)^2 \geq \frac{(n-1)^2}{4}. \]

Clearly, the minimum number of messages is \( n \) for any scheme and the minimum energy cost is \( 1/2 \) for any scheme. Thus, we have \( q_M \cdot q_E \geq (n-1)^2/(2n) = \Theta(n) \).

Notice that we generally assumed that the energy cost for supporting a link \( uv \) is \( ||uv||^\alpha \). Then we can show that

\[ (q_M)^{\alpha-1} q_E \geq \frac{n^{\alpha-1}}{2^{\alpha-1}}. \]
can finish the data collection operation in $O(N + h)$ rounds, hence $O(\Delta^2(N + h))$ time slots. On the other hand, any data collection algorithm will take $\Omega(N + h^2)$ time slot. Hence, data collection using MEPT is $g_E = O(\Delta(G)^4)$.

We construct a network example, in which the MEPT has delay $\Omega(\Delta^2)$ times of the optimum. Consider a rectangle $uwvz$ with side-length $||uw|| = p \cdot r$ and $||uz|| = p\Delta^2/8(1 - \epsilon)$. There are $p + 1$ nodes $u = u_1, u_2, \ldots, u_{p+1} = v$ uniformly distributed over the segment $uw$ and $q = p\Delta^2/8 - 1$ nodes $v_1, v_2, \ldots, v_q$ uniformly distributed over the rest of the 3 segments. Then it is easy to show that the MEPT path connecting $u$ and $v$ is $u_1v_1v_2\ldots v_qv$, with $q = p\Delta^2/8 - 1$ hops. Obviously, the path $u_1u_2\ldots u_p$ connecting $u$ and $v$ has the least delay $p$.

VI. RELATED WORK

Most existing convergecast methods [3], [6], [14] are based on a tree structure and with minimum either energy or data latency as the objective. For example, [14] first constructs a tree using greedy approach and then allocates DSSS or FHSS codes for its nodes to achieve collision-free, while [3], [6] uses TDMA to avoid collisions. In [3], the authors did not give any theoretical tradeoffs between energy cost and latency. Zhang and Huang [16] proposed a hop-distance based temporal coordination heuristic for adding transmission delays to avoid collisions. They studied the effectiveness of packet aggregation and duplication mechanisms with such convergecast framework. Kemelman and Kowalski [9] proposed a randomized distributed algorithm for convergecast that has the expected running time $O(\log n)$ and uses $O(n \log n)$ times of minimum energy in the worst case, where $n$ is the number of nodes. They also showed the lower bound of running time of any algorithm in an arbitrary network is $\Omega(\log n)$. However, they assume that all nodes can dynamically adjust its transmission power from 0 to any arbitrary value and a data message by a node can contain all data it has collected from other nodes. In [4], Chu et al. studied how to provide approximate and bounded-loss data collection in sensor networks instead of accurate data. Their method used replicated dynamic probabilistic models to minimize communication from sensor nodes to the base station.

To significantly reduce communication cost in sensor networks, in-network aggregation has been studied and implemented. In TAG (Tiny AGrgregation service) [10], besides the basic aggregation types (such as $count$, $min$, $max$, $sum$, average) provided by SQL, five groups of possible sensor aggregates are summarized: distributive aggregates (e.g., $count$, $min$, $max$, $sum$), algebraic aggregates (e.g., average), holistic aggregates (e.g., median), unique aggregates (e.g., count distinct), and content-sensitive aggregates (e.g., fixed-width histograms and wavelets). Notice that the first two groups aggregates are very easy to achieve by a tree-based method. To overcome the severe robustness problems of the tree approaches [10], [11], [15], multipath routing for in-network aggregation has been proposed [5], [13]. Then recently Manjhi et al. [12] combined the advantages of the tree and multi-path approaches by running them simultaneously in different regions of the network. In [7], Kashyap et al. studied a randomized (gossip-based) scheme using which all the nodes in a complete overlay network can compute the common aggregates of min, max, sum, average, and rank of their values using $O(n \log \log n)$ messages within $O(\log n \log \log n)$ rounds of communication. Kempe et al. [8] earlier presented a gossip-based method which can get the average in $O(\log n)$ rounds with $O(n \log n)$ messages.

VII. Conclusion

There are still a number of interesting questions left for future research. One is to design efficient algorithms when each node will produce a data stream. The second challenge is what is the best algorithm when we do not require that the found data item to be precise, i.e., we allow certain relative errors, or additive errors on the found answer. We also need to study the lower bound on energy cost and design energy efficient algorithm for other data operations, such as aggregation, selection, top-k query, and other holistic queries such as most frequent items, number of distinctive items.

REFERENCES