Deterministic Distributed Data Aggregation under the SINR Model

Nathaniel Hobbs[1], Yuexuan Wang[1], Qiang-Sheng Hua[1], Dongxiao Yu[2], and Francis C.M. Lau[2]

[1] Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, P.R. China
hobbs.nathaniel@gmail.com, {amywang,qshua}@mail.tsinghua.edu.cn
[2] Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong
{dxyu,fcmlau}@cs.hku.hk

Abstract. Given a set of nodes $\mathcal{V}$, where each node has some data value, the goal of data aggregation is to compute some aggregate function in the fewest timeslots possible. Aggregate functions compute the aggregated value from the data of all nodes; common examples include *maximum* or *average*. We assume the realistic physical (SINR) interference model and no knowledge of the network structure or the number of neighbors of any node; our model also uses physical carrier sensing. We present a *distributed* protocol to compute an aggregate function in $O(D + \Delta \log n)$ timeslots, where D is the diameter of the network, Δ is the maximum number of neighbors within a given radius and n is the total number of nodes. Our protocol contributes an exponential improvement in running time compared to that in [18].

Keywords: SINR interference model, data aggregation, physical carrier sensing.

1 Introduction

In this paper, we concentrate on minimizing latency when performing *data aggregation*, a fundamental operation in wireless networks. Informally, our problem is to, given a set of nodes distributed in a two-dimensional Euclidean plane, compute an aggregate function (e.g. a maximum or average, see below for formal definition) on the input data from all nodes in the network, and let every node be aware of this value in as little time as possible. A practical, real world application of data aggregation would be to compute an average temperature in a wireless sensor network, for example.

To put things into context, we adopt the physical SINR (signal to interference plus noise ratio) model. In the SINR model, the signal of a message from a sender propagates through the Euclidean plane continuing on into infinity, but fades with distance. A transmission is said to be successful if and only if the signal power at the intended recipient is sufficiently strong against background noise and the received signal power (seen as interference) of concurrent transmissions.

M. Agrawal, S.B. Cooper, and A. Li (Eds.): TAMC 2012, LNCS 7287, pp. 385–399, 2012.

Centralized algorithms for data aggregation under the SINR model have been widely studied [20,17,18,9]; however, more realistic *distributed* algorithms have not yet received significant attention. To the best of our knowledge, the only distributed data aggregation algorithm under SINR is given in [18]. Under stronger restrictions on the initial knowledge that nodes have and aided by physical carrier sensing, our protocol offers an exponential improvement in running time over [18].

Our algorithm is deterministic and confines nodes to only one piece of knowledge: a polynomial estimate of the total number of nodes in the network. Our distributed protocol makes extensive use of physical carrier sensing: the ability to sense when the shared channel used by all nodes is occupied. Technically, we develop a novel node election process using physical carrier sensing and make new use of the maximal independent set algorithm in [27], which may be of independent interest.

1.1 Our Contribution and Techniques

This paper presents an algorithm that is able to calculate (distributive and algebraic) aggregate functions in a deterministic, *distributed* way under the physical (SINR) model. When all nodes can perform physical carrier sensing, have access to a global clock and synchronously wake up, we show that an aggregate function can be computed in $O(D + \Delta \log n)$ timeslots. A trivial lowerbound for any data aggregation protocol is $\Omega(D + \Delta)$ timeslots [19], so our algorithm is at worst an $O(\log n)$ approximation to an optimal solution. To the best of the authors' knowledge, the current best distributed data aggregation algorithm is presented in [18] which has a running time that depends on the logarithm of the ratio of the longest and shortest links in the network. In the worst case this could require $\Omega(n)$ timeslots; our algorithm can therefore offer an exponential improvement. Our algorithm makes new use of finding maximal independent sets and we also present a practical technical novelty of a node election process aided by the use of physical carrier sensing.

1.2 Related Previous Work

The SINR model is more realistic than graph-based models (e.g. the protocol interference model), as shown both experimentally and theoretically [6,21,24]. In a seminal work, Moscibroda and Wattenhofer first abstracted and researched the *connectivity problem* in wireless networks in the context of the SINR model [23]. Their work on centralized connectivity algorithms with arbitrary power assignments (e.g. non-uniform power) was subsequently expanded on in [25,22,7]. Related to the connectivity problem, i.e. creating a connected spanning tree on a given set of nodes in a minimal number of timeslots, is that of *scheduling*: schedule a given a set of communication links in as few timeslots as possible. The scheduling problem was shown to be NP-complete in [4] (an $O(\log n)$ approximation was given in [5]). Halldorsson et al. proved hardness results for One-Shot scheduling (i.e. scheduling as many links as possible in a single timeslot) with uniform power (i.e. where every node transmits with the same, fixed power) in [10,11]. Kesselheim extended the result to the power control version of the problem in [15].

Among results for data aggregation in graph theoretic models, Li et al. studied the problem in the decentralized setting in [19]: an algorithm whose resulting schedule was shown to be within a constant factor of the optimal is given. A number of centralized algorithms have also been proposed [29,28,14].

There have been several results for centralized data aggregation algorithms under the SINR model. Li et al. studied the problem in [20] using uniform power with an asymptotically optimal scheduling algorithm of latency $O(\Delta+R)$, where R is the radius of the network and Δ the maximum number of neighbors of any node. Recently, this work was expanded on in [17], where an asymptotically optimal algorithm was produced for geometric minimum latency data aggregation in the dual power model. In the same paper, the authors show that no algorithm can have an approximation ratio better than $\Omega(\log n)$ in metric SINR as well as the NP-hardness of minimum latency data aggregation under geometric SINR. Hua et al. study the minimum latency link scheduling problem for arbitrary directed acyclic networks under both precedence and SINR constraints in [13] where they show hardness results and give approximation algorithms. In [18], Li et al. presented an algorithm with latency $O(\log^3 n)$, assuming that the transmission power of each node is large enough to cover the maximum distance in the network. Halldorsson et al. very recently presented a centralized algorithm for scheduling using non-uniform power that gives a constant approximation to an optimal scheduling [9]. In the same work, he shows that any algorithm that uses oblivious power assignments will require $\Omega(n)$ time slots for connectivity under certain node distributions.

Under the SINR model, to the best of our knowledge, the only decentralized data aggregation algorithm was given by Li et al. in [18]. When every node in the network knows its position, the network diameter, the number of neighbors, and has access to a global clock, [18] gives a distributed algorithm to perform data aggregation whose running time depends on the logarithm of the ratio of the longest and shortest links in the network, which may be $\Omega(n)$ in the worst case.

2 Model and Related Terminologies

2.1 Model

We consider a set of nodes $\mathcal{V} := \{x_1, x_2, \cdots, x_n\}$ distributed arbitrarily in the Euclidean plane. The Euclidean distance between two nodes $x_i, x_j \in \mathcal{V}$ is denoted by $d(x_i, x_j)$. A directed link λ_{ij} represents a communication request from sender x_i to receiver x_j, where the length of $\lambda_{ij} = d(x_i, x_j)$.

As in [8], we assume that transmissions are put into synchronized slots of equal length. All communication among nodes are done in synchronized rounds and nodes wake up and begin the execution of the algorithm synchronously. In any given time slot t, a node x can either transmit or receive a message, but not both. In every time slot, a *power assignment* is assigned to every node in a time slot, and is non-zero for node $x \in V$ if and only if x is to transmit a message in time slot t. Formally, a power assignment P_t is a function $P_t \mapsto \mathbb{R}^+$. A *schedule* $\mathcal{S} = (P_1, P_2, \cdots, P_{|\mathcal{S}|})$ is a sequence of $|\mathcal{S}|$ power assignments.

Considering a link λ_{sr} in the network, with sender x_s transmitting at power $P_s(r)$ to receiver x_r, the SINR at x_r is:

$$SINR_{x_r}(x_s, x_r) = \frac{\frac{P_s(r)}{d(x_s,x_r)^\alpha}}{N + \sum_{x_i,x_j \in \mathcal{V} \setminus \{x_s\}} \frac{P_i(j)}{d(x_i,x_r)^\alpha}} \tag{1}$$

where $N > 0$ is the ambient noise, $2 < \alpha \leq 6$ is the path loss exponent (the amount that the signal from x_s degrades over distance), and $\sum_{x_i,x_j \in \mathcal{V} \setminus \{x_s\}} \frac{P_i(j)}{d(x_i,x_r)^\alpha}$ is the total signal strength (viewed as interference at x_r) of other concurrently transmitting senders. Receiver x_r is said to have successfully received the transmission from x_s if and only if the $SINR_{x_r}(x_s, x_r)$ exceeds a given threshold $\beta \geq 1$.

We consider data aggregation in the unknown neighborhood model, i.e. nodes have no knowledge of the number of neighboring nodes within any given radius from themselves. Nodes do, however, have a polynomial estimate (specifically an upperbound) to the total number of nodes in the network. This assumption will not affect the asymptotic time bounds of our proposed algorithm compared to when the exact total number of nodes is known. Each node has a unique ID from the interval $[1, n]$ using the same number of bits, i.e. each node has a unique $\log n$ bit identifier, where nodes with smaller IDs pad their lower order bits by a prefix of 0s.

Every node is equipped with the ability to perform physical carrier sensing provided by a Clear Channel Assessment (CCA) circuit [26]. This is a natural assumption, as physical carrier sensing is widely used in wireless protocols such as Zigbee and Wi-Fi (IEEE 802.15.4 and 802.11 standards, respectively) [31]. For a given threshold T, a node can sense if the shared channel is occupied if the power sensed at that node by a transmitting neighbor is greater than or equal to T. A carrier sensing range R_s [3] is mapped from the carrier sensing power threshold T: $R_s = (\frac{P}{T})^{1/\alpha}$ where P is the transmission power. A node x_i can carrier sense a node x_j if and only if the distance between x_i and x_j is no larger than R_s.

In the absence of other concurrently transmitting nodes, let $R_{\mathsf{max}} = (\frac{P_{\mathsf{max}}}{\beta \cdot N})^{(1/\alpha)}$ be the maximum distance that a successful transmission can be from by a sender transmitting at power P_{max}. For a given distribution of nodes, Δ is the maximum number of nodes that lie within radius R_{max} centered around any node. If a node x_j is within the transmission radius of sender x_i, we say that the two nodes are within one "hop" from one another. Let $h(x_i, x_j)$ be the minimum hop distance between two nodes x_i and x_j. We define the diameter D of the network to be $D = max_{x_i,x_j} h(x_i, x_j)$.

2.2 Related Terminologies

AGGREGATION FUNCTION: In general, there are three classes of aggregation functions [12]: distributive (e.g. maximum, minimum, sum, count), algebraic (e.g. variance, average), and holistic (e.g. k^{th} largest/smallest). Our algorithm is only concerned with distributive and algebraic aggregate functions, and does not apply to holistic functions as in [16], Kuhn et al. proved that the decentralized computation of holistic functions is strictly harder than distributive or algebraic ones. As in [20], we define an aggregation function f to be *distributive* if for every pair of disjoint data sets X_1, X_2, $f(X_1 \cup X_2) = h(f(X_1), f(X_2))$ for some function h. For example, when f is *sum*, h can also be set to *sum*.

An *algabraic* aggregation function is defined as a combination of k distributive functions, k a constant, i.e. $f(X) = h(g_1(X), g_2(X), \cdots, g_k(X))$. For example, when f is *average*, then $k = 2$, g_1 and g_2 are the distributive functions $g_1(X) = \sum_{x_i \in X} x_i, g_2(X) = \sum_{x_i \in X} 1$ and $h(g_1, g_2) = g_1/g_2$. We assume that an algabraic function f is given in formula $h(g_1, g_2, \cdots, g_k)$, so we can just compute $g_i(X)$ distributively for $i \in [1, k]$ and $h(g_1, g_2, \cdot, g_k)$ at each node once all data has arrived.

Hereafter, we use the aggregate function *maximum* as an example for the sake of intelligibility, but any distributive aggregation function could easily be chosen.

Definition 1. *(MAXIMAL INDEPENDENT SET (MIS)): In a graph $G = (V, E)$, V a set of vertices and E edges, a set S is a maximal independent set if every edge of graph G has at least one endpoint in S and every vertex not in S has at least one neighbor in S.*

Definition 2. *(CONNECTED DOMINATING SET (CDS)): For a graph $G = (V, E)$, a subset V' of V is a dominating set if for all vertices $x_i \in V \setminus V'$, there exists an adjacent node $x_j \in V'$. Nodes in V' are called* dominators*, whereas nodes in $V \setminus V'$ are called* dominatees*. A subset C of V is a connected dominating set (CDS) if C is a dominating set and C induces a connected subgraph. By definition, an MIS is a dominating set.*

3 Data Aggregation Algorithm

Each node begins with some value. Our protocol will have every node in the network become aware of the highest value among them. The role of the dominators will be to collect data from their respective dominatee neighbors and then disseminate the highest value among them to all other nodes in the CDS.

We construct a CDS by first conducting an "election" process, where nodes decide whether or not they are dominators. The collection of these dominators will "cover" all nodes in the network, i.e. for all non-dominator nodes there exists a dominator within a certain radius R_{collect}. The dominators will be such that the distance to the closest neighboring dominator will not be more than three hops away with respect to R_{collect} (c.f. Fig. 1). We will ensure that that the dominating set is connected by allowing dominators to transmit to all nodes within a range $R_{\mathsf{CDS}} = 3 \cdot R_{\mathsf{collect}}$.

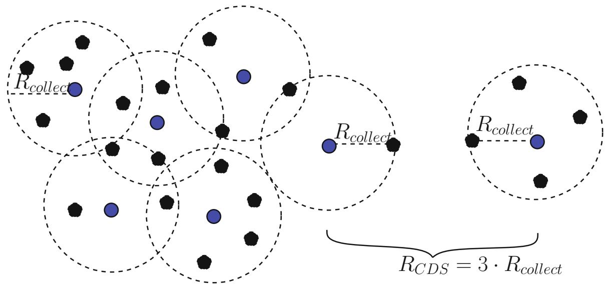


Fig. 1. An example of a MIS with respect to R_{collect}. The nodes at the center of the disks are dominators.

3.1 Algorithm Overview

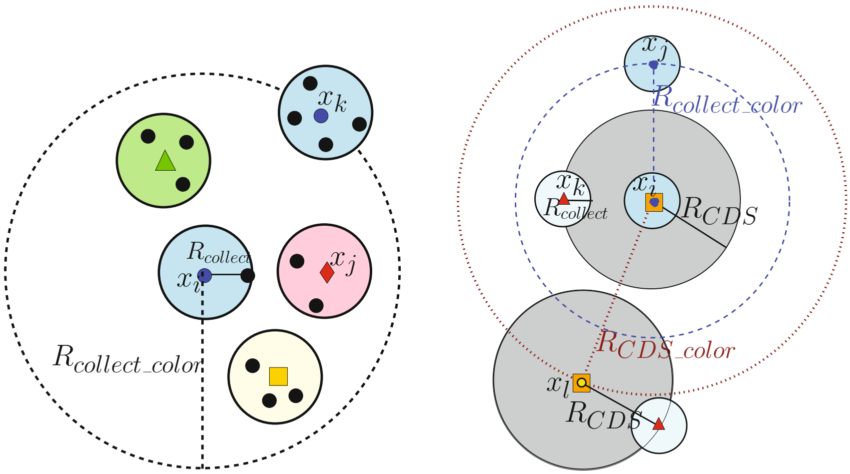
We model our network as a "unit" disc graph with respect to an elaborately chosen scaling factor. When adopting a uniform power assignment, the graph can be modelled as a "unit" disc graph $G = (V, E, R_{\mathsf{CDS}})$, where an edge $\lambda_{ij} \in E$ exists between x_i and x_j if and only if $d(x_i, x_j) \leq R_{\mathsf{CDS}}$. It should be noted that our model differs from traditional unit disc graphs because contention for access to the shared wireless channel can cause interferences when trying to receive messages. Because of this, we adopt the novel MIS algorithm that utilizes a collision detection based method presented in [27]. The running time of this algorithm is $O(\log n)$, will be denoted by t_{MIS}, and is known by all nodes.

In our algorithm, we find maximal independent sets in order to accomplish three different goals. The first MIS (performed with respect to R_{collect}) is computed to select which nodes will be in the dominating set, i.e. the *dominators*. Each node not in this initial MIS, but which lies within a disc of radius R_{collect} around some dominator, will be the *dominatee* of that dominator.

In order to deal with eventual wireless interferences, we expand on a coloring method used by Yu et al. in [30]. A second round of MISes (a constant number, each with respect to $R_{\mathsf{collect\_color}}$) will be used to color the dominators; dominators of the same color ($color_1$) can successfully send/receive a transmission to/from one of their dominatees (at a distance R_{collect}) in a single timeslot. See Fig. 2(a) for reference. In order to ensure that only one dominatee of any given dominator sends in any given timeslot, the dominator will use $O(\log n)$ timeslots to perform a binary search to grant the dominatee with the highest ID permission to send its data. A logarithmic number of timeslots, then, will suffice for every dominator in the network to successfully receive a message from one of their respective dominatees.

A third round of MISes (again, a constant number, but with respect to $R_{\mathsf{CDS\_color}}$) will be used to color dominators in such a way that dominators of the same color ($color_2$) can send messages to neighboring dominators (as far as R_{CDS} away) in order to disseminate values across the entire network. See Fig. 2(b) for reference.

Because no node knows the value of Δ, we cannot have dominators wait to collect data from *all* of their dominatees before disseminating values across the



(a) A coloring resulting from a round of MISes with respect to $R_{\mathsf{collect\_color}}$. The nodes in the center of the disks are dominators, the rest dominatees. By Lemma 2. x_i and x_k may collect data from a dominatee at the same time.

(b) Completed Preprocessing. By Lemma 2 and 3, x_i and x_j can exchange messages with their dominatees. Similarly, by Lemma 2, x_i and x_l can successfully broadcast to all other nodes within radius R_{CDS} at the same time.

Fig. 2.

network. We therefore carry out the aggregation process in rounds where, in a single round, dominators first collect data from just *one* of their respective dominatees then transmit values to neighboring dominators. The number of rounds this will take is unknown to the dominators. In our analysis, however, we bound the number of rounds required. In addition, because, while dominators are collecting data from their dominatees a logarithmic number of timeslots are required, it would be disadvantageous for dominators who wish to disseminate values throughout the CDS to wait for the collecting dominators to finish. We therefore restrict the dominatee collection process to be performed during even time slots, allowing dominators to perform the dissemination process during odd timeslots.

3.2 Algorithm

The entire data aggregation algorithm is broken up into three separate parts. Algorithm 1 defines the main data aggregation algorithm. It begins by running the preprocessing subroutine defined in Algorithm 2 where the dominating set is defined and colors are given. Algorithm 1 then goes on to ensure that dominators collect data from their dominatees and disseminate data throughout the CDS. Algorithm 3 is a subroutine that allows dominators to elect their dominatee of the highest node ID who has not sent yet to collect data from.

Algorithm 1. Data Aggregation

Initially, $max\_val = initial\_val, sent\_value = null, dominator = \mathsf{FALSE}, elected\_dominatee = null, color_1 = null, color_2 = null, new\_max = \mathsf{FALSE}$

Run Algorithm 2 to decide if dominator, and if so, get colors
loop
  \\\\ **Even timeslots dominators collects value from dominatees.**
  if Timeslot even **then**
    for $i = 0$ to $\mathsf{collector\_colors}$ **do**
      $elected\_dominatee$ = result of Algorithm 3 on input $color = i$.
      if $dominator = \mathsf{TRUE}$ and $color_1 = i$ and $elected\_dominatee \neq null$ **then**
        Transmit with power P_{collect} data request to $elected\_dominatee$ and listen for one timeslot. If value received greater than max_val, update max_val and set $new\_max = \mathsf{TRUE}$.
      else
        Listen for one timeslot and if receive request for node ID that matches your own to send value, send at power P_{collect} and set $sent\_val = \mathsf{TRUE}$.
  \\\\ **Odd timeslots dominators broadcast their** max_val
  if Timeslot odd **then**
    for $i = 0$ to $\mathsf{CDS\_colors}$ **do**
      if $dominator = \mathsf{TRUE}$ and $color_2 = i$ and $new\_max = \mathsf{TRUE}$ **then**
        Transmit max_val with power P_{CDS}, set $new\_max = \mathsf{FALSE}$.
      else
        Listen for one timeslot. If receive value greater than max_val then update max_val and set $new\_max = \mathsf{TRUE}$.

Algorithm 2. Preprocessing Subroutine

Elect dominators
Perform MIS algorithm [27] with power $P_{\mathsf{dominator}}$
if In MIS **then**
  $dominator = \mathsf{TRUE}$
Color dominators for successful dominatee data collection
for $i = 0$ to $\mathsf{collector\_colors}$ **do**
  if $dominator = \mathsf{TRUE}$ and $color_1 = null$ **then**
    Use t_{MIS} timeslots to perform MIS algorithm in [27] with power $P_{\mathsf{collect\_color}}$
    if In MIS **then** $color_1 = i$
  else Stay quiet for t_{MIS} timeslots.
Color dominators for successful CDS transmission
for $i = 0$ to $\mathsf{CDS\_colors}$ **do**
  if $dominator = \mathsf{TRUE}$ and $color_2 = null$ **then**
    Use t_{MIS} timeslots to perform MIS algorithm [27] with power $P_{\mathsf{CDS\_color}}$
    if In MIS **then** $color_2$ = i
  else Stay quiet for t_{MIS} timeslots.

Algorithm 3. Dominatee Election Subroutine

Require: *color*
Ensure: *elected_dominatee*

$L' = 0, L = \lceil n/2 \rceil, R = n-1, elected\_dominatee = null$
if $dominator = \mathsf{TRUE}$ and $color_1 = color$ **then**
    Transmit with power P_{collect} request for nodes with node IDs in range $[0, n-1]$ to reply, then listen for one timeslot. If sense occupied channel via physical carrier sensing, execute the while loop. Else, stay quiet for $2\log n$ timeslots and return *null*.
    while $L \neq R$ **do**
        Transmit with power P_{collect} request for nodes with node IDs in range $(L, R]$ to reply, then listen for one timeslot.
        if sense node response via physical carrier sense **then** $L' = L, L = \lceil (L+R)/2 \rceil$
        else $R = L, L = \lceil (L'+L)/2 \rceil$
    Return L.
else
    for $2(\log n + 1)$ timeslots **do**
        if $dominator = \mathsf{FALSE}$ and $sent\_value = \mathsf{FALSE}$ **then**
            Listen. If receive request for ID, then transmit with power $P_{\mathsf{dominator}}$ if ID in range.
        else Stay quiet.
    Return *null*.

We define many parameters for our algorithm, most of them fairly contrived. Their intricacy largely stems from a method we use to bound interferences in Lemma 2. The parameters have been calculated so that our methods will work. We define the following parameters for our algorithm, and some intuition regarding them follows.

1. **constants**: **(i)** $\mathsf{collector\_colors} = (2[96\beta(2^{\alpha-1}+\frac{\alpha-1}{\alpha-2})]^{\frac{1}{\alpha}}+1)^2$, **(ii)** $\mathsf{CDS\_colors} = (6[96\beta(2^{\alpha-1}+\frac{\alpha-1}{\alpha-2})]^{\frac{1}{\alpha}}+1)^2$
2. **Radii**: **(i)** $R_{\mathsf{collect}} = \min\left\{\frac{(N\beta/T)^{(1/\alpha)}R_{\mathsf{max}}}{3[96\beta(2^{\alpha-1}+\frac{\alpha-1}{\alpha-2})]^{\frac{1}{\alpha}}}, \frac{1}{3}\cdot(\frac{1}{2})^{\frac{1}{\alpha}}R_{\mathsf{max}})\right\}$ **(ii)** $R_{\mathsf{CDS}} = 3R_{\mathsf{collect}}$, **(iii)** $R_{\mathsf{collect\_color}} = [96\beta(2^{\alpha-1}+\frac{\alpha-1}{\alpha-2})]^{\frac{1}{\alpha}}R_{\mathsf{collect}}$, **(iv)** $R_{\mathsf{CDS\_color}} = [96\beta(2^{\alpha-1}+\frac{\alpha-1}{\alpha-2})]^{\frac{1}{\alpha}}R_{\mathsf{CDS}}$
3. **Powers**: **(i)** $P_{\mathsf{dominator}} = TR^{\alpha}_{\mathsf{collect}}$, **(ii)** $P_{\mathsf{collect\_color}} = TR^{\alpha}_{\mathsf{collect\_color}}$, **(iii)** $P_{\mathsf{CDS\_color}} = TR^{\alpha}_{\mathsf{CDS\_color}}$, **(iv)** $P_{\mathsf{collect}} = 2N\beta R^{\alpha}_{\mathsf{collect}}$, and **(v)** $P_{\mathsf{CDS}} = 2N\beta R^{\alpha}_{\mathsf{CDS}}$

$\mathsf{collector\_colors}$ and $\mathsf{CDS\_colors}$ are the number of colors needed to color all dominators for successful dominatee data collection and CDS data dissemination, respectively.

R_{collect} is the radius in which dominators are intended to collect data from their respective dominatees. R_{CDS} is the radius with respect to which our graph is connected and also the furthest distance any dominator is to its closest neighboring dominator. $R_{\mathsf{collect\_color}}$ (resp. $R_{\mathsf{CDS\_color}}$) is the minimum distance between two dominators that share the same $color_1$ (resp. $color_2$); that is, it is the "buffer" distance between two simultaneously collecting (resp. disseminating) dominators.

All our power assignments are static. $P_{\mathsf{dominator}}$ is the power level used when electing domintators; it is also used by dominatees for the physical carrier sensing binary search. $P_{\mathsf{collect\_color}}$ (resp. $P_{\mathsf{CDS\_color}}$) is the power level used when coloring dominators (i.e. performing an MIS) with respect to $R_{\mathsf{collect\_color}}$ (resp. $R_{\mathsf{CDS\_color}}$). P_{collect} is the power level used when exchanging messages between dominators/dominatees. P_{CDS} is the power level used when dominators are transmitting to neighboring dominators that compose the CDS. Strictly speaking, we define our powers in relation to P_{max}. Formally, we let $\max\{P_{\mathsf{CDS\_color}}, P_{\mathsf{CDS}}\} = P_{\mathsf{max}}$, so R_{CDS} will be a constant fraction of R_{max}.[1]

Lastly, we define the following node attributes: *dominator*, a Boolean to define if node is a dominator (TRUE) or dominatee (FALSE). $color_1$ (resp. $color_2$), all nodes that have the same color can simultaneously broadcast within radius R_{collect} (resp. R_{CDS}). *initial_val* is the initial value that the node begins with (e.g. initial temperature). *max_val* is the current maximum value thus received by the node. *sent_val* is a Boolean used by dominatees to keep track of whether they have sent their *initial_val* to their respective dominator.

4 Analysis

In this section we will give a detailed analysis of our algorithm, show its correctness and bound its running time.

Lemma 1. *([27])The total time to compute a MIS in each stage is $t_{MIS} = O(\log n)$ and each node computing it knows whether or not they are in it.*

Recall that t_{MIS} is known by all nodes in advance of their execution of Algorithm 1.

Lemma 2. *Dominators that have the same $color_1$ (resp. $color_2$) can successfully broadcast a message to all nodes within the disc of radius R_{collect} (resp. R_{CDS}) centered around them in the same timeslot.*

Proof. We have a set of dominators that all share the same $color_1$. Let x_i be some such dominator and x_j one of its dominatees. Recall $d(x_i, x_j) \le R_{\mathsf{collect}}$. We claim that no matter how many other dominators of $color_1$ transmit, that x_j can successfully receive a message sent by x_i.

Using a method first developed by Moscibroda et al. in [23], and expanded on in [30], we use a "ring method" to show that interferences are bounded. Because all simultaneously transmitting dominators lie at distance at least $R_{\mathsf{collect\_color}}$ from each other, then discs of radius $R_{\mathsf{collect\_color}}/2$ centered at each such dominator do not overlap. Let $\mathsf{R}_l = \{x_k : lR_{\mathsf{collect\_color}} \le d(x_i, x_k) \le (l{+}1)R_{\mathsf{collect\_color}}\}$. Notice now, that all discs of radius $R_{\mathsf{collect\_color}}/2$ in R_l around the dominators are completely contained within the extended ring $\mathsf{R}_l^+ = \{x_k : lR_{\mathsf{collect\_color}} - R_{\mathsf{collect\_color}}/2 \le d(x_i, x_k) \le (l+1)R_{\mathsf{collect\_color}} + R_{\mathsf{collect\_color}}/2\}$. See Fig. 4 for reference.

[1] We assume that graph $G(V, E, R_{\mathsf{CDS}})$ is connected.

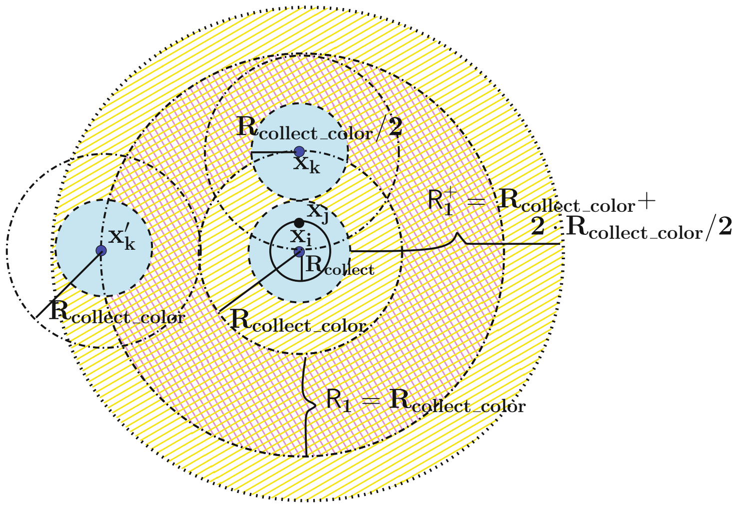


Fig. 3. A simple layout the rings R_1 and R_l^+ surrounding a dominator x_i. The disks surrounding dominators of the same *color*$_1$ of radius $R_{\mathsf{collect}}/2$ do not overlap, e.g. with dominator x_i and x_k. By Lemma 2, x_j can successfully receive a transmission from dominator x_i, despite simultaneously transmitting dominators of the same *color*$_1$ as x_i (like x_k and x'_k).

We bound the interference by dominators in these rings on x_i's dominatee x_j. Denote the interference received by unwanted dominator x_k on x_j as I_j^k. Then the interference $I_j^{R_l}$ on x_j by all unwanted senders of the same *color*$_1$ as x_i in ring R_l is at most:

$$\begin{aligned} I_j^{R_l} &= \sum_{x_k \in \mathsf{R}_l \text{ with given } color_1} I_j^k \\ &\leq \frac{Area(\mathsf{R}_l^+)}{Area(Disc(R_{\mathsf{collect\_color}}/2))} \cdot \frac{P_{\mathsf{collect}}}{(lR_{\mathsf{collect\_color}} - R_{\mathsf{collect}})^\alpha} \\ &= \frac{\pi((l+1)R_{\mathsf{collect\_color}} + R_{\mathsf{collect\_color}}/2)^2 - \pi(lR_{\mathsf{collect\_color}} - R_{\mathsf{collect\_color}}/2)^2}{\pi(R_{\mathsf{collect\_color}}/2)^2} \cdot \frac{P_{\mathsf{collect}}}{(lR_{\mathsf{collect\_color}} - R_{\mathsf{collect}})^\alpha} \\ &= \frac{8(2l+1)P_{\mathsf{collect}}}{(lR_{\mathsf{collect\_color}} - R_{\mathsf{collect}})^\alpha} < \frac{48P_{\mathsf{collect}}}{(l-1/2)^{\alpha-1}R^\alpha_{\mathsf{collect\_color}}} \end{aligned} \tag{2}$$

The last inequality comes from the fact that $R_{\mathsf{collect\_color}} > 2R_{\mathsf{collect}}$ (recall that $R_{\mathsf{collect\_color}} = [96\beta(2^{\alpha-1} + \frac{\alpha-1}{\alpha-2})]^{\frac{1}{\alpha}} R_{\mathsf{collect}}$). We can now bound the total interference at a dominatee x_j of x_i by simultaneously transmitting domintators:

$$\begin{aligned} I_j &= \sum_{l=1}^{\infty} \frac{48P_{\mathsf{collect}}}{(l-1/2)^{\alpha-1}R^\alpha_{\mathsf{collect\_color}}} \leq \frac{48P_{\mathsf{collect}}}{R^\alpha_{\mathsf{collect\_color}}} \sum_{l=1}^{\infty} \frac{1}{(l-1/2)^{\alpha-1}} \\ &= \frac{48P_{\mathsf{collect}}}{R^\alpha_{\mathsf{collect\_color}}}(2^{(\alpha-1)} + \sum_{l=2}^{\infty} \frac{1}{(l-1/2)^{\alpha-1}}) \leq \frac{48P_{\mathsf{collect}}}{R^\alpha_{\mathsf{collect\_color}}}(2^{(\alpha-1)} + \frac{\alpha-1}{\alpha-2}) \leq N \end{aligned} \tag{3}$$

By the value of P_{collect}, the SINR at x_j from its transmitting dominator x_i is: $SINR_{x_j}(x_i, x_j) \geq \frac{P_{\mathsf{collect}}/R^{\alpha}_{\mathsf{collect}}}{I_j+N} \geq \beta$. Therefore, all simultaneously transmitting dominators of the same $color_1$ can successfully broadcast a message to their respective dominatees.

The proof for successful transmission of dominators with $color_2$ is similar, and is omitted for brevity. □

Lemma 3. *Dominatees within the disc of radius R_{collect} centered around their respective dominators (of the same $color_1$) can transmit a message to those dominators successfully in the same timeslot.*

Proof. This lemma is the converse of Lemma 2. That is to say, all dominators of $color_1$ are now receievers and one of their respective dominatees is now the sender.

If there are two dominators of the same $color_1$ x_i and x_k with respective dominatees x_j and x_l (c.f. Fig. 3), then we know by Lemma 2 that both x_j and x_l can both successfully receive a message from their dominator. The interference received by a dominatee x_j by sending dominator x_k is equal to the amount of interference at x_k when their roles are reversed.

Formally, the interference received by a dominatee x_j received by a foreign dominator x_k is $\frac{P_k}{d(x_j,x_k)^{\alpha}}$. Clearly, if x_j sends at power $P_j = P_k$, then the interference received at x_k will be identical. □

Lemma 4. *Dominators performing a binary search in Algorithm 3 can successfully sense responses from dominatees using physical carrier sensing.*

Proof. This proof is omitted for brevity.

Lemma 5. *The number of dominators contained in a disc of radius $R_{\mathsf{collect\_color}}$ (resp. $R_{\mathsf{CDS\_color}}$) is a constant no more than* $\mathsf{collector\_colors} = (2[96\beta(2^{\alpha-1} + \frac{\alpha-1}{\alpha-2})]^{\frac{1}{\alpha}} + 1)^2$ *(resp.* $\mathsf{CDS\_colors} = (6[96\beta(2^{\alpha-1} + \frac{\alpha-1}{\alpha-2})]^{\frac{1}{\alpha}} + 1)^2$*).*

Proof. The proof follows from a simple area argument is omitted for brevity.

Lemma 6. *Each dominator can elect a dominatee with the highest ID that has not sent yet and then collect a message from it in $O(\log n)$ timeslots.*

Proof. In line 3 of Algorithm 3, one timeslot is used for all dominators of $color_1$ to transmit a message to see if any dominatees respond; by Lemma 2, this will be successfully received by their respective dominatees. Another timeslot is used to wait for a reply; by Lemma 4, any reply will be sensed. If there is none, then dominators wait for $2\log n$ timeslots for others to finish. In the i^{th} iteration of the while loop on line 4 of Algorithm 3, dominators that sensed a response broadcast a message requesting dominatees with a range covering $1/2^i$ of all total node IDs to respond. This takes one timeslot to accomplish, and because only dominators of the same $color_1$ perform this broadcast, by Lemma 2, all of their respective dominatees will successfully receive it. These dominators listen for one timeslot to allow the dominatees to respond; dominatees within the ID range who have not sent their information yet use this timeslot to reply with a message at power

$P_{\mathsf{dominator}}$ and by Lemma 4, the dominatees respective dominators will be able to sense this response. The range is then halved and the process repeated. After $\log n$ iterations of this process (each taking two timeslots), the dominatee with the highest ID will be discovered by each corresponding dominator. This binary search, then, takes a total of $2 \log n + 2$ timeslots.

Another combined two timeslots are used in line 8 and 10 of Algorithm 1 for dominators of the same $color_1$ to collect the actual data value of their respective elected dominatees. By Lemma 2 (resp. Lemma 3), each dominatee (resp. dominator) is able to receive the message successfully.

For the set of dominators of the same $color_1$ to successfully receive a message from a dominatee, then, uses $2 \log n + 4$ timeslots. The for loop in line 5 of Algorithm 1 has dominators of all (constantly many) colors perform this collection. Thus, $O(\log n)$ timeslots are sufficient for each dominator to successfully collect data from one of their respective dominatees. □

Theorem 1. *For all placements of nodes in the plane, there exists a schedule using $O(D + \Delta \log n)$ timeslots for the highest value in the network to be known by all nodes.*

Proof. Each node knows in advance the amount of time required ($t_{MIS} = O(\log n)$) to perform the MIS algorithm in [27]. By Lemma 1, The election of dominators takes t_{MIS} timeslots. A constant number of executions, collector_colors (resp. CDS_colors), of the MIS algorithm in [27] are needed to color the dominators with their $color_1$ (resp. $color_2$) for the dominatee data collection (resp. data dissemination) process by Lemma 5. By Lemma 1, each coloring needs $O(\log n)$ timeslots. The total running time of the preprocessing subroutine in Algorithm 2 is therefore $O(\log n)$.

In each iteration of the loop in line 2 of Algorithm 1, by Lemma 6, each dominator will be able to successfully collect a data item from one of its dominatees in $O(\log n)$ timeslots. After Δ iterations of this loop, the maximum value of any node in the network will be contained in the dominating set, requiring a total of $O(\Delta \log n)$ timeslots.

In the same iteration of the loop, by Lemma 2, each dominator of the same $color_2$ will be able to broadcast its current highest value to neighboring dominators, and by Lemma 5 the number of colors to iterate through is constant. Because dominators disseminating data do not have to wait for dominators collecting data to perform their binary search (as they occur during different timeslot intervals), each dominator will be able to successfully broadcast a message to all nodes in the disc of radius R_{CDS} surrounding them in a constant number of timeslots. At this point because R_{CDS} is a constant fraction of R_{max}, at most $O(D)$ timeslots are needed for the highest value to be disseminated throughout the network.

The entire execution time required for the aggregate function to be computed and known by all nodes in the network is therefore $O(D + \Delta \log n)$ and the theorem follows. □

5 Conclusions

In this paper, under the SINR interference model aided by physical carrier sensing, we present a distributed, deterministic algorithm for computing (distributive and algabraic) aggregate functions in wireless networks. With no knowledge beyond a polynomial estimate of the number of nodes in the network, our decentralized protocol computes an aggregate function and ensures the result is obtained by every node in the network using only $O(D + \Delta \log n)$ timeslots. In particular, aided by the use of physical carrier sensing, our protocol can outperform the distributed data aggregation technique used in [18] by an exponential factor despite the fact our protocol is more limited in its initial knowledge of the network. As a future work, the study of distributed data aggregation using non-uniform powers could be particularly meaningful as they have been shown to have significant effects on reducing time complexity in some cases [2,23]. Another natural future direction would be to investigate distributed data aggregation algorithms under SINR computing holistic aggregate functions. Extending our work to cases without physical carrier sensing abilities and/or under the asynchronized communication model, when nodes do not share a global clock, would also be beneficial directions of future study.

Acknowledgements. This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00302, the National Natural Science Founda- tion of China Grant 61103186, 61073174, 61033001, 61061130540, the Hi-Tech research and Development Program of China Grant 2006AA10Z216, and Hong Kong RGC-GRF grants 714009E and 714311.

References

1. Chen, X., Hu, X., Zhu, J.: Minimum Data Aggregation Time Problem in Wireless Sensor Networks. In: Jia, X., Wu, J., He, Y. (eds.) MSN 2005. LNCS, vol. 3794, pp. 133–142. Springer, Heidelberg (2005)
2. Fanghanel, F., Kesselheim, T., Racke, H., Vocking, B.: Oblivious Interference Scheduling. In: PODC (2009)
3. Fu, L., Liew, S., Huang, J.: Effective Carrier Sensing in CSMA Networks under Cumulative Interference. In: INFOCOM (2010)
4. Goussevskaia, O., Oswald, Y.A., Wattenhofer, R.: Complexity in Geometric SINR. In: Mobihoc (2007)
5. Goussevskaia, O., Wattenhofer, R., Halldorsson, M., Welzl, E.: Capacity of Arbitrary Wireless Networks. In: INFOCOM (2009)
6. Gronkvist, J., Hansson, A.: Comparison Between Graph-based and Interference Based STDMA Scheduling. In: Mobihoc (2001)
7. Gu, Z., Wang, G., Hua, Q.-S., Wang, Y.: Improved Minimum Latency Aggregation Scheduling in Wireless Sensor Networks under the SINR Model. In: CWSN (2011)
8. Gupta, P., Kumar, P.R.: The Capacity of Wireless Networks. IEEE Transactions on Information Theory (2000)
9. Halldorsson, M.M., Mitra, P.: Wireless Connectivity and Capacity. In: SODA (2012)

10. Halldórsson, M.M., Wattenhofer, R.: Wireless Communication Is in APX. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5555, pp. 525–536. Springer, Heidelberg (2009)
11. Halldorssson, M.M., Wattenhofer, R.: Computing Wireless Capacity (2010) (unpublished manuscript)
12. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2006)
13. Hua, Q.-S., Wang, Y., Yu, D., Tan, H.: Minimum Latency Link Scheduling for Arbitrary Directed Acyclic Networks under Precedence and SINR Constraints. Journal of Interconnection Networks 12(1-2), 87–107 (2011)
14. Huang, S.C.-H., Wan, P., Vu, C.T., Li, Y., Yao, F.: Nearly Constant Approximation for Data Aggregation Scheduling in Wireless Sensor Networks. In: INFOCOM (2007)
15. Kesselhelm, T.: A Constant-factor Approximation for Wireless Capacity Maximization with Power Control in the SINR Model. In: SODA (2011)
16. Kuhn, F., Locher, T., Wattenhoffer, R.: Tight Bounds for Distributed Selection. In: SPAA (2007)
17. Lam, N.X., An, M.K., Huynh, D.T., Nguyen, T.N.: Minimum Latency Data Aggregation in the Physical Interference Model. MSWiM (2011)
18. Li, H.-X., Wu, C., Hua, Q.-S., Lau, F.C.-M.: Latency-minimizing Data Aggregation in Wireless Sensor Networks under Physical Interference Model. Ad Hoc Networks (2012); Minimum-Latency Aggregation Scheduling in Wireless Sensor Networks under Physical Interference Model. MSWiM (2010)
19. Li, X.-Y., Wang, Y., Wang, Y.: Complexity of Data Collection, Aggregation, and Selection for Wireless Sensor Networks. IEEE Transactions on Computers (2011)
20. Li, X.-Y., Xu, X.H., Wang, S.G., Tang, S.J., Dai, G.J., Zhao, J.Z., Qi, Y.: Efficient Data Aggregation in Multi-hop Wireless Sensor Networks under Physical Interference Model. In: MASS (2009)
21. Maheshwari, R., Jain, S., Das, S.R.: A Measurement Study of Interference Modeling and Scheduling in Low-power Wireless Netowrks. In: SenSys (2008)
22. Moscibroda T.: The Worst Case Capacity of Wireless Sensor Networks. In: IPSN (2007)
23. Moscibroda, T., Wattenhofer, R.: The Complexity of Connectivity in Wireless Networks. In: INFOCOM (2006)
24. Moscibroda, T., Wattenhofer, R., Weber, Y.: Protocol Design Beyond Graph-based Models. In: Hotnets (2006)
25. Moscibroda, T., Wattenhofer, R., Zollinger, A.: Topology Control Meets SINR: the Scheduling Complexity of Arbitrary Topologies. In: Mobihoc (2006)
26. Scheideler, C., Richa, A., Santi, P.: An O(log n) Dominating Set Protocol for Wireless Ad-hoc Networks under the Physical Interference Model. In: Mobihoc (2008)
27. Schneider, J., Wattenhofer, R.: What Is the Use of Collision Detection (in Wireless Networks)? In: Lynch, N.A., Shvartsman, A.A. (eds.) DISC 2010. LNCS, vol. 6343, pp. 133–147. Springer, Heidelberg (2010)
28. Wan, P.-J., Huang, S.C.-H., Wang, L.X., Wan, Z.Y., Jia, X.H.: Minimum-latency Aggregation Scheduling in Multihop Wireless Networks. In: Mobihoc (2009)
29. Yu, B., Li, J., Li, Y.: Distributed Data Aggregation Scheduling in Wireless Sensor Networks. In: INFOCOM (2009)
30. Yu, D., Wang, Y., Hua, Q.-S., Lau, F.C.M.: Distributed Local Broadcasting Algorithms in the Physical Interference Model. In: DCOSS (2011)
31. Zheng, J., Jamalipour, A.: Wireless Sensor Networks: a Networking Perspective. Wiley-IEEE Press, Hoboken (2009)