Minimum Latency Gossiping in Wireless Sensor Networks

Min Kyung An, Nhat X. Lam, Dung T. Huynh and Trac N. Nguyen
Department of Computer Science
University of Texas at Dallas
Richardson, Texas 75080
Email: {mka081000, lxnhat, huynh, nguyentn}@utdallas.edu

Abstract—Gossiping is one of the most crucial applications in Wireless Sensor Networks (WSNs) which has been the focus of many researchers. A main issue of gossiping is how to assign timeslots to nodes for interference-free data transmission. There are three models concerning gossiping in WSNs: unit-sized, bounded-sized, or unbounded-sized messages. For these models, the problem of constructing minimum latency gossiping schedules has been widely studied in the literature although most of the existing studies are based on the graph model.

In this paper, we study the Minimum Latency Gossiping (MLG) problem with unbounded-sized messages in the graph model as well as the more realistic physical interference model known as Signal-to-Interference-Noise-Ratio (SINR) where there exist relatively few works [1] on the three gossiping models. In the SINR model, we prove the NP-hardness of the MLG problem with unbounded-sized messages. In both the graph model and SINR model, we propose a constant factor approximation algorithm that yields schedules whose latency is bounded by $O(\Delta + R)$, where $\Delta$ is the maximum node degree of a network and $R$ is its radius. We also study the performance of the algorithm through simulation.

Index Terms—Gossiping, All-to-All Broadcast, NP-hardness, Geometric Graph, Signal-to-Interference-Noise-Ratio, Approximation Algorithm

I. INTRODUCTION

The gossiping problem has been the focus of many researchers as it is one of the crucial applications in Wireless Sensor Networks (WSNs) along with the broadcast problem. In a WSN that consists of a number of sensor nodes, the gossiping problem, which is also known as all-to-all broadcast, is to distribute the message of each node to all the other nodes in the network, whereas the broadcast problem is to distribute a unique message from a source node to all the other nodes. As the small-sized sensors have limited energy resources, researchers have focused on reducing energy consumption while distributing data in a network so that the network life time is extended. An interesting approach is to assign timeslots to nodes to obtain a good (short) schedule thereby avoiding unnecessary transmissions. As data distribution may occur periodically, reducing the latency of the schedule, that is, constructing schedules with a minimum number of timeslots, has been a fundamental issue in such applications.

In the literature on the gossiping problem, there are three models: unit-sized, bounded-sized, or unbounded-sized messages. In the unit-sized-message model, a node can send a single unit-sized message, and therefore combining messages is not allowed. In the bounded-sized-message model, a node can combine messages that have been received so far (up to some limit), whereas in the unbounded-sized-message model, there is no limit on the length of the combined message.

The gossiping problem in the graph model has been investigated by many researchers over the last several years. In the collision-free graph model, for the unit-sized-message model, [2] introduced a 1974-approximation algorithm, and [3] proposed an optimal randomized schedule with a latency bounded by $O(n \log n)$, where $n$ is the number of sensor nodes in the network. Later, [4] introduced a 27-approximation algorithm which produces gossiping schedules with a latency bounded by $27(n + R - 1)$, where $R$ is the network radius. In [5], two approximation algorithms with constant factors of 20 and 34 have been studied. For the bounded-sized-message model, [6] studied the gossiping problem where messages can be combined into a single message whose size is bounded by $\log n$, and [7] gave an improvement over [6]. For the unbounded-sized-message model, [8] showed that their algorithm produces gossiping schedules with $O(g + \frac{\nu \log n}{\log \nu - \log \log n})$ timeslots, where $\nu = \Omega(\log n)$ and $g$ is the network diameter, and [9] introduced a constant factor approximation algorithm whose latency is bounded by $7\Delta + 258R$. Although there have been many studies in the collision-free graph model for the gossiping problem, surprisingly there exists no study in the collision-interference-free graph model, to the best of our knowledge.

The graph model which has been used in many studies, however, is not an adequate model since cumulative interference caused by all the other concurrently transmitting nodes is ignored. Thus, researchers have started investigating problems in WSNs in the more realistic physical interference model which is known as the Signal-to-Interference-Noise-Ratio (SINR) model since its introduction by Gupta et al. in [10]. For the SINR model, [1] introduced a constant factor approximation algorithm for the gossiping problem in the unit-sized-message model, and [11] proposed a $O(\log n)$-approximation algorithm for the unbounded-sized-message model. However, [11] considered only the concurrently sending nodes within some predefined interference area from a receiver, and therefore, some interference caused by senders located far away is ignored.

While these studies have been concerned with gossiping,
some other researchers have focused on related applications such as data aggregation and broadcast. Table I shows a summary of some related works. Although there have been several approximation algorithms for these related problems, there are surprisingly few studies regarding the complexity of the problems. For the problem of data aggregation, NP-hardness was proved for the collision-free model by [12], and [13] and [14] showed not only an $\Omega(\log n)$ approximation lower bound for the problem, but also the NP-hardness in the collision-interference-free graph model and the geometric SINR model, respectively. For the broadcast problem, [2] proved its NP-hardness in the collision-free model. However, to the best of our knowledge, the NP-hardness of the broadcast problem in both the collision-interference-free and SINR models as well as that of the gossiping problem in all network models remain open. (Note that the existing NP-hardness of the broadcast problem in [2] holds for the collision-free graph model, but not for the other models.)

<table>
<thead>
<tr>
<th>Models</th>
<th>Data Aggregation</th>
<th>Broadcast</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-Free</td>
<td>[12], [15], [16]</td>
<td>[2], [17], [18], [19]</td>
</tr>
<tr>
<td>C-I-Free</td>
<td>[13]</td>
<td>[20], [21], [22]</td>
</tr>
<tr>
<td>SINR</td>
<td>[23], [24], [14], [25], [26]</td>
<td>[21]</td>
</tr>
</tbody>
</table>

TABLE I: Summary of Works on Problems Related to Gossiping

In this paper, we continue the study of the gossiping problem with unbounded-sized messages in the graph model and the geometric SINR model. Extending the proof in [14], we show that this problem is NP-hard in the geometric SINR model. Additionally, for the uniform power model, we introduce a constant factor approximation algorithm yielding schedules whose latency is bounded by $O(\Delta + R)$ in both the graph model and the geometric SINR model. Our algorithm gives an improved approximation ratio of 44 over the existing ratio of 285 given by [9] in the collision-free graph model. In the collision-interference-free graph model and the SINR model, our approximation algorithm is the first one to the best of our knowledge. Moreover, regarding broadcast, the broadcast subroutine of our gossiping algorithm provides broadcast schedules with a better approximation ratio than existing ones given by [20], [21] for the collision-interference-free model.

This paper is organized as follows. Section II describes our network models and defines the Minimum Latency Gossiping (MLG) problem with unbounded-sized messages. In Section III, we show the NP-hardness of MLG in the geometric SINR model. Section IV introduces a constant factor approximation algorithm for the MLG problem and shows some simulation results. Section V contains some concluding remarks.

II. PRELIMINARIES

A. Network Models

In our paper, we model a wireless sensor network as $(V, D, p)$, where $V$ represents a set of $n$ nodes, and $D : V \times V \rightarrow R^+$ represents the distance function between nodes. Letting $p_{\text{max}} : V \rightarrow R^+$ be the maximum power level function, we define a power assignment function as $p : V \rightarrow R^+$, $p(u) \leq p_{\text{max}}(u)$, $u \in V$.

1) Graph Model: In the graph model, given a transmission power level $p(u)$ for each node $u$, let $R_{p(u)}^v = \{v \in V, D(u, v) \leq p(u)\}$ denote the set of all nodes that can be reached by $u$ with the power level $p(u)$. Two nodes $u$ and $v$ can communicate only if they are in the coverage area of each other, i.e., $u \in R_v^u$, and $v \in R_p^u$. However, we also need to consider the collision or interference that interferes with the communication. Given a power level $p(u)$ of $u$, the interference range of $u$ is defined as $\rho \cdot p(u)$, where $\rho \geq 1$ is the interference factor. Given $\rho \geq 1$, let $F_{p(u)}^v = \{v \in V, D(u, v) \leq \rho \cdot p(u)\}$ denote the set of all nodes in the interference range of $u$. Then, collision (or conflict) is said to occur at a receiver node $v$ if there exist other concurrently sending nodes $u$ and $v$ such that $v \in R_v^u \cap F_p^v$, where $\rho > 1$.

In the literature, the graph model concerning only collision (i.e., when $\rho = 1$) is called the collision-free model, whereas the graph model concerning both collision and interference (i.e., when $\rho > 1$) is called the collision-interference-free model. In the graph model, the communication graph can be modeled as a bidirectional graph $G(\mathit{V, E})$, where $E = \{(u, v)|u, v \in V, D(u, v) \leq p(u) \text{ and } D(v, u) \leq p(v)\}$.

2) SINR Model: In the physical interference model (SINR) [10], if a node $u$ transmits with its power level $p(u)$, then the power received at another node $v$ is $p(u) \cdot D(u, v)^{-\alpha}$, where $\alpha$, the path loss exponent, is commonly assumed to be in the interval $[2, 6]$. In order that node $v$ can receive and decode data sent by $u$, the ratio of the received power at $v$ to the interference caused by all the other concurrently transmitting nodes and background noise must be beyond an SINR threshold $\beta \geq 1$. Formally, node $v$ can receive data successfully via the link $(u, v)$ only if

$$\text{SINR}(u, v) = \frac{p(u) \cdot D(u, v)^{-\alpha}}{N + I_v} \geq \beta$$

where $N > 0$ is the background noise, and

$$I_v = \sum_{w \in X: w \neq (u, v)} p(w) \cdot D(w, v)^{-\alpha}$$

is the cumulative interference at $v$ caused by nodes in $X$ that is the set of other concurrently transmitting nodes. Observing that $u$ can send its data to the nodes within the distance $(D(u))^\frac{1}{\alpha}$, the communication graph can be modeled as a directed graph $G(\mathit{V, E})$, where $E = \{(u \rightarrow v)|u, v \in V, D(u, v) \leq (D(u))^\frac{1}{\alpha}\}$.

B. Problem Definition

In this paper, we are concerned with gossiping in the unbounded-sized-message model, i.e., we assume that multiple messages can be combined as a single message, and there is no limit on the length of a message that one node can transmit.
Let $I_P$ be an instance of Partition which consists of a set $S$ of $n$ distinct and positive integers $a_1, a_2, \ldots, a_n$. Without loss of generality, assume that $a_1 < a_2 < \ldots < a_n$. We construct in polynomial time an instance $I_M$ of the MLG problem as follows.

In the instance $I_M$, we have $2n + 3$ nodes including $2n$ nodes $s_i$ and $r_i$, $1 \leq i \leq n$, 2 nodes $s_{n+1}$ and $s_{n+2}$ and a center node $s$. These nodes are deployed on the plane at the following positions.

\[
\text{pos}(s) = (0, 0), \\
\text{pos}(s_{n+1}) = (0, -\left(\frac{24P}{N\beta(A^2+\beta+24)}\right)^{\frac{1}{2}}), \\
\text{pos}(s_{n+2}) = (-\left(\frac{24P}{N\beta(A^2+\beta+24)}\right)^{\frac{1}{2}}, 0)
\]

and, for all $1 \leq i \leq n$,

\[
\text{pos}(s_i) = \left(\left(\frac{P}{N\beta}\right)^{\frac{1}{2}}, 0\right), \\
\text{pos}(r_i) = \left(\left(\frac{P}{N\beta}\right)^{\frac{1}{2}}, d_0\right)
\]

where $P$ is the maximum power value to be defined below.

Let $\sigma = \sum_{i=1}^{n} a_i$, $A = (\frac{1}{a_n-1})^{\frac{1}{2}} - (\frac{1}{a_n})^{\frac{1}{2}}$, $b_i = \frac{\sigma}{12\pi}$ and $d_0 = \left(\frac{12\pi^2\beta\sigma}{24n+2\pi\beta+24}\right)^{\frac{1}{2}}$.

With $d(u, v)$ denoting the Euclidean distance between $u$ and $v$, we define the maximum power level for the nodes $s_i$ as follows:

\[
\text{p}_\text{max}(s_i) = \text{p}_\text{max}(s_{n+1}) = \text{p}_\text{max}(s_{n+2}) = P, \\
\text{p}_\text{max}(r_i) = N\beta d(r_i, r_{i+1})^{\alpha}, 1 \leq i \leq n, r_{n+1} \equiv s
\]

\[
\text{p}_\text{max}(s) = N\beta d(s, s_1)^{\alpha} = \frac{P}{b_i} = \frac{12\pi^2\beta}{a_nA^2} > P
\]

**Fact 1.** Let $T_i = \{s_j | 1 \leq j \leq n + 1 \land i \neq j\}$. It holds for all $1 \leq i \leq n$ that $\text{SINR}(s_i, r_i)$ exceeds $\beta$ when node $s_i$ is assigned to send data to $r_i$ at the same timeslot as the nodes in $T_i$.

**Fact 2.** For all $1 \leq i \leq n$, $s_i$ can send data only to $r_i$.

**Fact 3.** $s_{n+1}$ and $s_{n+2}$ can send data only to $s$.

**Fact 4.** $r_{i+1}$ can receive data from $r_i$ (where $r_{n+1} \equiv s$) if and only if there is no other node sending to $r_i$’s timeslot.

**Fact 5.** It holds for all $1 \leq i < n$ that $r_i$ can send data to $s$ through $r_{i+1}$ only.

**Fact 6.** $s_1$ can receive data from $s$ if and only if there is no other node sending in that timeslot.

**Lemma 1.** $I_P$ has a solution if and only if $I_M$ has an optimal gossiping schedule of length $n + 3$.

**Proof:** Omitted.

From Lemma 1 we obtain

**Theorem 2.** The MLG problem is NP-hard.

**IV. CONSTANT FACTOR APPROXIMATION ALGORITHM**

In this section, we introduce a constant factor approximation algorithm for the MLG problem in the graph model and the physical interference (SINR) model, assuming a uniform power level $P$, i.e., for all $v \in V$, $p(v) = P$. In those models, we make the following assumptions:

For the graph model, we set the maximum link length $r =$...
\( P \), and assume that the undirected graph \( G = (V, E) \), where \( E = \{(u, v)|d(u, v) \leq r\} \), is connected and \( \rho \geq 1 \).

For the SINR model, define \( r_{max} = \left( \frac{P}{N\delta^2} \right)^{\frac{1}{\delta}} \) and notice that if node \( u \) on link \( (u, v) \) of length \( r_{max} \) is transmitting, then none of the remaining nodes can transmit concurrently. Thus we are interested in links \( (u, v) \), where \( d(u, v) \leq \delta \left( \frac{P}{N\delta^2} \right)^{\frac{1}{\delta}} \) for some constant \( \delta \in (0, 1) \) as considered in [23]. Thus, in the SINR model, we let \( r \geq \delta \left( \frac{P}{N\delta^2} \right)^{\frac{1}{\delta}} \), and assume that the undirected graph \( G = (V, E) \), where \( E = \{(u, v)|d(u, v) \leq r\} \), is connected and \( \rho \geq 2 \) [10].

A. Algorithm

We first introduce some definitions and notations that are used subsequently:

- **Graph Center:** Given a communication graph \( G = (V, E) \), we call node \( c \) a center node if the distance from \( c \) to the farthest node from \( c \) is minimum.

- **Maximal Independent Set (MIS):** A subset \( V' \subseteq V \) of the graph \( G \) is said to be independent if for any vertices \( u, v \in V', (u, v) \notin E \). An independent set is said to be maximal if it is not a proper subset of another independent set.

- **Connected Dominating Set (CDS):** A dominating set (DS) is a subset \( V' \subseteq V \) such that every vertex \( v \) is either in \( V' \) or adjacent to a vertex in \( V' \). A DS is said to be connected if it induces a connected subgraph.

1) Gossip Tree Construction: Our algorithm assigns timeslots to nodes based on a gossip tree whose construction is based on that of a data aggregation tree in [23]. After a center node \( c \) is chosen, we construct a breadth-first-search (BFS) tree (cf. [30]) on \( G \) rooted at node \( c \) so that the latency can be bounded in terms of the network radius \( R \) rather than its diameter. [23] finds an MIS on \( G \) using an algorithm in [31] based on the BFS tree. We call nodes in the MIS dominators and the others dominatees. The constructed MIS satisfies the property that the distance between any pair of its complementary subset is exactly two hops [31]. Next, we connect the dominators using some connectors that were originally dominatees to obtain a CDS of \( G \). If there exist some remaining dominatees that are not connected to the CDS, then each of such dominatees is connected to its neighboring dominator that has the smallest hop distance to \( c \) in the BFS tree. We denote the newly formed tree by \( T \), and use it as the gossip tree in our algorithm.

2) Gossip Scheduling Algorithm: Our algorithm (Algorithm 1) starts by partitioning the network into square cells each of which has diagonal length \( r \), and labels each cell with the label \( CL(i, j) \) if its upper-left corner has coordinate \( (i, j) \). Once the gossip tree \( T \) is obtained, we gather data from each node to the center node \( c \) (Steps 4 – 5 in Algorithm 1). Node \( c \) then broadcasts the collected data combined with its own based on \( T \) (Steps 6 – 14 in Algorithm 1). Assigning timeslots for data gathering and broadcasting is based on a constant \( C \) which guarantees that two senders (receivers) can transmit (receive) data successfully if they are at least \( C \) cells apart from each other. The constant \( C \) is set as follows in the two different models:

- **Graph model:** \( C = \lceil \rho \cdot \sqrt{2} + 2 \rceil \)
- **SINR model:**
  \[ C = \left( \frac{P - 2\pi}{N(\delta^{-\frac{1}{2}}(\alpha-1))(\delta^{-\frac{1}{2}}) + 1} \right) \]

Algorithm 1 Gossiping

**Input:** A set \( V \) of nodes with a uniform power level \( P \)

**Output:** Length of schedule

1: Partition the network into square cells each of which has diagonal length \( r \).
2: Construct a gossip tree \( T \) using the algorithm in [23].
3: Set the first timeslot \( t \leftarrow 1 \).
4: // Data Gathering starts.
5: \( t \leftarrow \text{CCA}(V, T, t) \)
6: // Broadcasting start.
7: \( TS(c) \leftarrow TS(c) \cup \{t\} \)
8: \( t \leftarrow t + 1 \)
9: for \( i = 1 \) to \( R - 1 \) do
10: \( Let \ S_i \subseteq V \ be \ the \ set \ of \ connectors \ at \ level \ i \ in \ T. \)
11: \( if \ S_i \neq \emptyset \ then \ t \leftarrow \text{RBS}(S_i, t) \ end \) if
12: \( Let \ S_{i+1} \subseteq V \ be \ the \ set \ of \ dominators \ at \ level \ i + 1 \ in \ T. \)
13: \( if \ S_{i+1} \neq \emptyset \ then \ t \leftarrow \text{SBS}(S_{i+1}, t) \ end \) if
14: end for
15: return \( t \leftarrow 1 \)

Data Gathering. In order to gather data to \( c \), we use the data aggregation algorithm called Cell Coloring algorithm in [13], which is included in Algorithm 2. The Cell Coloring algorithm, which is based on an algorithm in [23], is originally built for the graph model assuming \( \rho \geq 1 \). We use the algorithm not only for the graph model, but also for SINR model where the constant \( C \) is defined accordingly.

Broadcast. Once all the data is gathered to \( c \), \( c \) broadcasts the collected data combined with its own to the whole network. We introduce a new broadcast scheduling algorithm which is also based on the data aggregation algorithm of [23]. The details of our broadcast algorithm are contained in Algorithm 1 (Steps 6 – 14). In Step 7, \( c \) (which is also a dominator) broadcasts the data to its neighbors (lower level connectors). In Steps 10 – 11, the connectors which have just received data from the upper level dominators relay the data to their lower level dominators. These steps are based on the receivers’ locations (Algorithm 4), i.e., the connectors whose receivers (lower level dominators) are \( C \) cells apart from each other are assigned to the same timeslot. Then, in Steps 12 – 13, the dominators relay the data to its dominatees and lower level connectors. These dominators are scheduled based on their (senders’) locations (Algorithm 3), i.e., the dominators which are \( C \) cells apart from each other are assigned to the same timeslot. These Steps 10 – 13 are repeated until the data is
disseminated to the whole network.

In Algorithms 1, 3 and 4, \( T(S(v)) \) denotes the set of timeslots at which node \( v \) is activated to send data.

Algorithm 2 Cell Coloring Algorithm (CCA) [13]

Input: A set \( V \) of nodes, a tree \( T \) and a starting timeslot \( t \)

Output: Timeslot \( t \)

1: Let \( S \subseteq V \) be the set of dominatees in \( T \).
2: if \( S \neq \emptyset \) then \( t \leftarrow SBS(S, t) \) end if
3: for \( i = R \) to 2 do
4: Let \( S_i \subseteq V \) be the set of dominators at level \( i \) in \( T \).
5: if \( S_i \neq \emptyset \) then \( t \leftarrow SBS(S_i, t) \) end if
6: Let \( S_{i-1} \subseteq V \) be the set of connectors at level \( i-1 \) in \( T \).
7: if \( S_{i-1} \neq \emptyset \) then \( t \leftarrow SBS(S_{i-1}, t) \) end if
8: end for
9: return \( t \)

Algorithm 3 AssignTimeSlot (SBS) [13]

Input: A set \( S \) of nodes and a starting timeslot \( t \)

Output: Timeslot \( t \)

1: while \( S \neq \emptyset \) do
2: Pick one node \( v_r \in S \) in each cell. Let \( S' \subseteq S \) be the set of such nodes.
3: for \( c_1 = 0, \ldots, C - 1 \) and \( c_2 = 0, \ldots, C - 1 \) do
4: \( X \leftarrow \emptyset \), \( X \leftarrow \{v_s | v_s \in S' \text{ with } CL(x, y) \text{ such that } c_1 = x \mod C \text{ and } c_2 = y \mod C\} \)
5: if \( X \neq \emptyset \) then
6: for each \( v_s \in X \) do
7: \( TS(v_s) \leftarrow TS(v_s) \cup \{t\} \)
8: end for
9: \( t \leftarrow t + 1, \ S \leftarrow S \setminus X \)
10: end if
11: end for
12: end while
13: return \( t \)

Algorithm 4 Receiver-based Scheduling (RBS)

Input: A set \( S \) of nodes and a starting timeslot \( t \)

Output: Timeslot \( t \)

1: for each \( v_s \in S \) do
2: \( Z \leftarrow Z \cup v_s \text{'s lower level dominators in } T \)
3: end for
4: for \( c_1 = 0, \ldots, C - 1 \) and \( c_2 = 0, \ldots, C - 1 \) do
5: \( X \leftarrow \emptyset \), \( X \leftarrow \{v_r | v_r \in Z \text{ with } CL(x, y) \text{ such that } c_1 = x \mod C \text{ and } c_2 = y \mod C\} \)
6: if \( X \neq \emptyset \) then
7: for each \( v_r \in X \) do
8: \( v_s \leftarrow v_r \text{'s upper level connector in } T \)
9: \( TS(v_s) \leftarrow TS(v_s) \cup \{t\} \)
10: end for
11: \( t \leftarrow t + 1 \)
12: end if
13: end for
14: return \( t \)

Then any two sender nodes that are at least \( C \) cells apart from each other can concurrently send (receive) data without interference.

Next, we prove that the latency of a gossip schedule found by Algorithm 1 is bounded by \( O(\Delta + R) \). We need the following lemmas.

Lemma 5. [23] The number of connectors in a cell is at most 12.

Lemma 6. In Algorithm 1, gathering data from all the other nodes to center node \( c \) takes at most \( \Delta \cdot C^2 + 6 \cdot R \) timeslots.

Proof: First consider the dominatees in each cell and their dominator \( v \) (Steps 1–2 in Algorithm 2). Obviously, there are at most \( \Delta \) dominatees in each cell, and one of those \( \Delta \) dominatees must be a connector to connect the dominator \( v \) to another dominator. Therefore, the number of dominatees is bounded by \( \Delta - 1 \), and gathering data from all the dominatees to the corresponding dominators takes at most \( (\Delta - 1)C^2 \) timeslots.

Next, consider the dominators at level \( i \) (Steps 4–5 in Algorithm 2). Since there is at most one dominator in each cell, gathering data from all the dominators at level \( i \) to the connectors at level \( i - 1 \) takes \( C^2 \) timeslots. As this process is repeated at most \( \frac{R}{2} \) times, gathering data from all dominators to upper level connectors takes at most \( \frac{R}{2} \cdot C^2 \) timeslots.

Now, let us consider only the connectors at level \( j \), where \( 1 < j < R \) (Steps 6–7 in Algorithm 2). In one cell, at most 11 of those connectors at level \( j \) have the role of sending the collected data to their dominators at level \( j - 1 \); one remaining connector in the cell must relay data from the dominator at level \( j - 1 \) to another dominator at level \( j - 3 \). Therefore, gathering data from the connectors at level \( j \) to the dominators at level \( j - 1 \) takes at most \( 11C^2 \) timeslots. As this process is repeated \( \frac{R-2}{2} \) times, it takes at most \( \frac{R-2}{2} \cdot 11C^2 \) timeslots. On the other hand, all the connectors at level 1 send data to
a sink, and this requires at most $12C^2$ timeslots.

Thus, the latency of data gathering is at most $(\Delta - 1)C^2 + R^2 \cdot C^2 + \frac{R^2}{2} \cdot 11 \cdot C^2 + 12C^2 = \Delta \cdot C^2 + 6R$.

**Lemma 7.** In Algorithm 1, broadcasting data takes at most $1 + C^2 \cdot (R - 1)$ timeslots.

**Proof:** First, note that in Steps 7–8 in Algorithm 1, the center node sends data to its neighbors in one timeslot. Next, consider the connectors at level $i$ (Steps 10–11). Any two of those connectors cannot share one dominator at level $i + 1$ on the gossip tree $T$; otherwise a cycle would result. This means that those connectors can be scheduled based on the receivers’ locations, i.e., the locations of the corresponding dominators at level $i + 1$. As there is at most one dominator in each cell, relaying data from the connectors at level $i$ to the dominators at level $i + 1$ takes at most $C^2$ timeslots. Since Steps 10–11 are repeated $\frac{R - 1}{2}$ times, at most $\frac{R - 1}{2} \cdot C^2$ timeslots are needed.

Now consider the dominators at level $j$ (Steps 12–13). Since at most one dominator can be located in a cell, sending data from the dominators to the connectors at level $j + 1$ takes at most $C^2$ timeslots. As the Steps 12–13 are repeated $\frac{R - 1}{2}$ times, it takes at most $\frac{R - 1}{2} \cdot C^2$ timeslots.

Thus, the latency of broadcasting is at most $1 + 2 \cdot C^2 \cdot (R - 1)$ timeslots.

**Lemma 8** (Lower bound). [9] If $\Delta$ is the maximum node degree in a network, then every gossip schedule with unbounded-size messages has at least $\Delta + R - 1$ timeslots.

**Theorem 9.** Algorithm 1 produces gossip schedules whose latency is bounded by $C^2 \cdot \Delta + (C^2 + 6)R + (1 - C^2) = O(\Delta + R)$, and it is a constant-factor approximation with the factor of $2(C^2 + 6)$.

**Proof:** By Lemma 6 and Lemma 7, the latency, denoted by $SOL$, of schedules produced by Algorithm 1 is bounded by $C^2 \cdot \Delta + (C^2 + 6)R + (1 - C^2)$. Next, without loss of generality, assume that $n \geq 2$, and therefore $\Delta \geq 1$ and $R \geq 1$. Then, denoting the lower bound by $OPT$, the approximation ratio is

$$\frac{SOL}{OPT} \leq \frac{C^2\Delta + (C^2 + 6)R + (1 - C^2)}{\Delta + R - 1} \leq \frac{(C^2 + 6)\Delta + (C^2 + 6)R}{\Delta + R - 1} \leq 2(C^2 + 6)$$

Note that this approximation ratio is 44 in the collision-free graph model, an improvement on the approximation ratio of 258 given by [9]. To the best of our knowledge, in the collision-interference-free graph model and the SINR model, our results are the first constant-factor approximation algorithms.

Also note that our broadcast scheduling algorithm yields an approximation ratio of $C^2$ by Lemma 7. In the collision-interference-free graph model, it is $C^2 = [\rho \sqrt{2} + 2]^2$ which improves on the approximation ratio of $6[\frac{2}{3}(\rho + 2)]^2$ given by [21]. In addition, [20] studies the broadcast algorithm with $\rho = 2$ giving an approximation ratio of 26, whereas our approximation ratio is 25.

**C. Simulation**

In our simulation, networks are generated randomly in the Euclidean plane where the number of nodes is 500. The nodes are randomly deployed on an area of size $m \times m$, where $m = 300, 400$ and 500. For each $m$, we generate 100 different networks, and average the latencies produced by the algorithm over the networks. For the simulation, we set the various parameters as follows:

1) **Graph model:**
   - Choice of $\rho$: We use $\rho = \{1, 2, \cdots, 10\}$
   - Initial power assignment: We first use Kruskal’s algorithm [30] to find the minimum spanning tree $T_{MST}$ using edge weights defined as the distance between any two nodes. Then, we set the initial power $P = r$, where $r$ is the length of the longest edge in $T_{MST}$. Given the initial power assignment, we obtain the initial graph $G = (V, E)$, where $E = \{(u, v)|d(u, v) \leq r\}$.

2) **SINR model:**
   - Choices of SINR parameters: We use $\alpha = 5$, $N = 1$ and $\beta = 1$.
   - Choice of $\delta$: We use $\delta = \{0.1, 0.2, \cdots, 0.9\}$.
   - Initial power assignment: Given a uniform power assignment, if $\delta$ is too small, the graph may not be connected. In order to make the initial graph connected even with the smallest $\delta = 0.1$ in our simulation, we set the initial power $P = \beta N \left(\frac{d}{0.9}\right)\frac{1}{2}$, and obtain the initial graph $G = (V, E)$, where $E = \{(u \rightarrow v)|d(u, v) \leq \delta \left(\frac{P}{N\beta}\right)^{\frac{1}{2}}\}$.

Table II shows the performance of Algorithm 1 in the graph model. For fixed $\rho$, as the network becomes denser (i.e., the network node degree becomes larger and the network radius smaller) the latency decreases, whereas as the network becomes sparser (i.e., the network node degree becomes smaller and the network radius larger) the latency increases. For fixed network density, as $\rho$ becomes larger (i.e., the interference range becomes wider), the latency increases.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>300 × 300</th>
<th>400 × 400</th>
<th>500 × 500</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>313.13</td>
<td>321.52</td>
<td>321.59</td>
</tr>
<tr>
<td>2</td>
<td>357.85</td>
<td>368.71</td>
<td>369.98</td>
</tr>
<tr>
<td>3</td>
<td>430.32</td>
<td>440.18</td>
<td>440.23</td>
</tr>
<tr>
<td>4</td>
<td>461.79</td>
<td>469.39</td>
<td>469.41</td>
</tr>
<tr>
<td>5</td>
<td>513.79</td>
<td>522.64</td>
<td>523.09</td>
</tr>
<tr>
<td>6</td>
<td>536.89</td>
<td>542.38</td>
<td>544.81</td>
</tr>
<tr>
<td>7</td>
<td>557.04</td>
<td>562.19</td>
<td>564.57</td>
</tr>
<tr>
<td>8</td>
<td>596.11</td>
<td>599.17</td>
<td>601.54</td>
</tr>
<tr>
<td>9</td>
<td>613.90</td>
<td>617.61</td>
<td>619.30</td>
</tr>
<tr>
<td>10</td>
<td>642.97</td>
<td>645.58</td>
<td>647.34</td>
</tr>
</tbody>
</table>

**TABLE II:** Latencies of Algorithm 1 in Graph Model

Table III shows the performance of Algorithm 1 in the SINR model. For fixed $\delta$, as the network becomes denser (i.e., the network node degree becomes larger, but the network radius becomes smaller) the latency decreases, whereas as
the network becomes sparser (i.e., the network node degree becomes smaller, but the network radius becomes larger) the latency increases. For fixed density, as \( \delta \) becomes smaller (i.e., the network node degree becomes smaller, but the network radius becomes larger), the latency increases, whereas as \( \delta \) becomes larger (i.e., the network node degree becomes larger, but the network radius becomes smaller), the latency decreases.

<table>
<thead>
<tr>
<th>( \delta )</th>
<th>300 ( \times ) 300</th>
<th>400 ( \times ) 400</th>
<th>500 ( \times ) 500</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>654.91</td>
<td>662.98</td>
<td>663.09</td>
</tr>
<tr>
<td>0.2</td>
<td>651.21</td>
<td>660.08</td>
<td>662.20</td>
</tr>
<tr>
<td>0.3</td>
<td>646.19</td>
<td>648.97</td>
<td>651.39</td>
</tr>
<tr>
<td>0.4</td>
<td>605.93</td>
<td>615.83</td>
<td>615.86</td>
</tr>
<tr>
<td>0.5</td>
<td>512.63</td>
<td>523.91</td>
<td>524.17</td>
</tr>
<tr>
<td>0.6</td>
<td>401.22</td>
<td>414.80</td>
<td>414.91</td>
</tr>
<tr>
<td>0.7</td>
<td>341.18</td>
<td>354.35</td>
<td>355.08</td>
</tr>
<tr>
<td>0.8</td>
<td>295.06</td>
<td>305.52</td>
<td>307.22</td>
</tr>
<tr>
<td>0.9</td>
<td>264.31</td>
<td>274.54</td>
<td>275.66</td>
</tr>
</tbody>
</table>

TABLE III: Latencies of Algorithm 1 in SINR Model

From these tables, we can observe that having smaller network radius rather than having smaller network node degree gives better results. This is because the variation of network node degree affects only the latencies of data aggregation schedules, whereas the variation of the network radius affects the latencies of the schedules for data aggregation as well as broadcast.

V. CONCLUSION

In this paper, we have studied the Minimum Latency Gossiping (MLG) problem with unbounded-sized messages in the graph model and SINR model. We have proved the NP-hardness of the problem in the SINR model, and proposed a constant factor approximation algorithm whose latency is bounded by \( O(\Delta + R) \) in both graph and SINR models assuming a uniform power level. We have also studied the performance of the algorithm through simulation. As to future work, we plan to study the gossiping problem with unit-sized and bounded-sized messages.

REFERENCES