Minimum-Latency Gossiping in Multi-hop Wireless Networks

Scott C.-H. Huang*
Computer Science Department,
City University of Hong Kong,
Tat Chee Ave,
Kowloon, Hong Kong SAR
shuang@cityu.edu.hk

Hongwei Du
Computer Science Department,
City University of Hong Kong,
Tat Chee Ave,
Kowloon, Hong Kong SAR
hongwei@cityu.edu.hk

E.-K. Park†
CSEE Department,
University of Missouri at Kansas City,
5100 Rockhill Rd, Kansas City,
MO 64110, USA
ekpark@umkc.edu

ABSTRACT

We studied the minimum-latency gossiping (all-to-all broadcast) problem in multi-hop wireless networks defined as follows. Each node in the network is initially given a message and the objective is to design a minimum-latency schedule such that each node distributes its message to all other nodes. We considered the unit-size message model, in which different messages cannot be combined as one message, and the unit disk graph model, in which a link exists between two nodes if and only if their Euclidean distance is less than 1. This problem is known to be NP-hard in such models. In this work we designed a gossiping scheme that significantly improved all current gossiping algorithms in terms of approximation ratio. Our work has approximation ratio 27, a great improvement of the current state-of-the-art algorithm (which has ratio 1000+).

Categories and Subject Descriptors

F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General

General Terms

Algorithms

Keywords

TDMA, broadcast, gossip

1. INTRODUCTION

Broadcast is a fundamental operation in wireless networks. Naïve flooding is not practical as it causes severe contention, collision, and congestion. Avoiding collision, reducing redundancy as well as increasing reliability in wireless networks are the main objectives of the broadcast storm problem [23]. Numerous network protocols are based on broadcasting: routing, information dissemination, and service/resource discovery. Since many systems have stringent end-to-end delay requirements, the design of low-latency broadcasting scheme is essential to many practical applications.

There are two basic tasks in network communication, broadcasting and gossiping. The broadcasting operation is about distributing a single message from one source node to all other nodes. The gossiping operation is about distributing a unique message from each node to all other nodes in the network. Essentially, the gossiping task can be viewed as *all-to-all broadcast*. There are three models regarding whether or not we can combine two or more messages as a single message: *unit-size*, *bounded-size*, and *unbounded-size* models.

In this paper, regarding message models we study the gossiping problem in the *unit-size* model, in which multiple messages CANNOT be combined as a single message. As for network models we considered the unit disk graph (UDG) model and designed the *Interleaved Gossiping Algorithm* that improved all previous gossiping algorithms in this model. Our algorithm is theoretically proven to be *constant approximation* with ratio 27. This is known to be the algorithm that has the lowest latency so far.

The rest of this paper is organized as follows. Related work is shown first in §2. We present the preliminaries in §3, and then we present a naïve gossiping algorithm in §4. A better algorithm, namely the interleaved gossiping algorithm, is presented in §5. A concrete example is shown in §6 to help understand our algorithm. In §7 we analyze our algorithm presented in §5 and establish theoretical bounds for it. We conclude this work in §8.

2. RELATED WORK

Lots of works have been done about the gossiping problem in unbounded-size model [4, 6, 7, 9, 15, 19, 24, 25]. Chrobak et al [6, 8] showed that deterministic gossiping can be performed in unknown directed ad hoc radio networks in $O(n^{3/2}\log^2 n)$time. A constructive version of their algorithm was recently proposed by Indyk [19]. This result was improved by Gasieniec and Lingas [15] for networks with diameter $D = n^\alpha$, $\alpha < 1$. They presented a gossiping algorithm that runs in $O(n\sqrt{D}\log^2 n)$ time. These results show that radio networks with a long diameter constitute a bottleneck in deterministic radio gossiping with messages of an arbitrary size. There is another approach for the radio gossiping algorithm with running time $O(Dd^2\log^3 n)$, where d stands for the maximum in-degree of the underlying graph of connections. Chrobak et al [7] proposed a randomized radio gossiping algorithm with expected running time $O(n\log^4 n)$. Studies on oblivious gossiping in

*Supported in part by the Research Grant Council of Hong Kong under the project CERG CityU 113807.

†This work was completed while the author was working at the National Science Foundation.

MobiHoc'08, May 26–30, 2008, Hong Kong SAR, China.

ad hoc radio networks can be found in [4]. Ravishankar and Singh studied this problem in two different models. They presented distributed gossiping algorithms for networks with nodes placed randomly on a line [24] and a ring[25].

The gossiping problem in bounded-size model was studied in the matching model [2, 3]. The results presented in [2] include the studies of the exact complexity of the gossiping problem in Hamiltonian graphs and k-ary trees, and optimal asymptotic bounds for general graphs (in the matching model with unit messages). It also contains a number of asymptotically optimal results in matching model with messages of arbitrarily bounded size. Flamini and Perennes [11] also studied the gossiping problem in bounded-size model. They focused on graphs with bounded degree. Bar-Yehuda et al [1] proposed a randomized gossiping algorithm in log-size model for unknown topology. Christersson et al [5] studied deterministic $b(n)$-gossiping algorithms in ad hoc radio networks meaning that each combined message contains at most $b(n)$ single messages or bits of auxiliary information, where b is an integer function and n is the number of nodes in the network. They derived theoretical upper bounds for many functions b.

Gossiping in the unit-size model has also been studied in the literature. Gasieniec and Potapov [16] studied this problem in general graph model. They proposed several optimal or close to optimal $O(n)$-time gossiping procedures for various standard network topologies, including lines, rings, stars and free trees. They also proved that there exists a radio network topology in which the gossiping (with unit size messages) requires $O(n \log n)$ time. Manne and Xin designed a randomized gossiping algorithm in general graphs that has latency $O(n \log n)$. Gandhi et al [14] studied this problem in the UDG model and designed a constant approximation algorithm. However, their approximation is too large to be practical. Although, the exact ratio was not given in [14], we estimated it to be at least 1000. Our interleaved gossiping algorithm is a significant improvement and has ratio 27.

3. PRELIMINARIES

We consider a network of n nodes. Each node is equipped with an RF transceiver that can be used to send or receive data. We consider omni-directional antennae only, and a node's transmission/reception range is roughly a disk centered at that node. This type of network can be represented as a *disk graph* as follows. Let G be a graph representing a network of n nodes. An arc (or directed edge) exists from u to v if and only if v lies in u's transmission area (which is a disk). For simplicity, we further assume that all nodes have the same transmission range. In such a case, we can normalize their radius to 1, and an edge exists between u, v if and only if the distance between them is less than 1. Note that in this case all edges are bi-directional and we can simply use an undirected graph to represent the network topology. This is called the UDG model. In this work, for simplicity, we only consider UDGs.

A node can either send or receive data at one time, and it can receive data correctly only if exactly one of its neighbors is transmitting at that moment. If two or more nodes are transmitting simultaneously and there is a node in their overlapped transmission area, then this node cannot receive the message clearly since both transmissions are interfering with each other. This type of situation is called *collision*. A node is equipped with some memory, so it can store messages received from neighbors and forward to other nodes. Note that a node cannot forward a message unless it has already received from the node having that message.

A TDMA schedule can be modeled as a function $\mathcal{W}$ from the set of natural numbers $\mathbb{N}$ (representing time) to V's power set (i.e. subsets of V) as follows. $\mathcal{W} : \mathbb{N} \to 2^V$, in which time $t \in \mathbb{N}$ is mapped to $\mathcal{W}(t) \subset V$, denoting that the nodes in $\mathcal{W}(t)$ are scheduled to transmit at time t. The *latency* of a TDMA schedule $\mathcal{W}$ is the last time slot such that there are still some node(s) transmitting. Formally, the latency of $\mathcal{W}$ can be defined as $lat(\mathcal{W}) = \min\{t | \mathcal{W}(t') = \emptyset, \forall t' > t\}$.

Minimum-Latency Gossiping Problem: Given a UDG $G = (V, E)$ such that each node has a message (depending on this node), find a TDMA schedule of minimum latency such that each node successfully distributes its own message to the entire network.

Note that, in order to make sense for this problem, we have to assume that the network is strongly connected. Essentially, the gossiping task can be viewed as *all-to-all broadcast*. Since both terms have been used extensively in the literature and they were studied in different message models in different context, we have to distinguish between the following three cases in order to clarify their problems and objectives.

1. *Messages of unit size*: In this model, all messages have the same size. If a node need to forward two messages, it has to do two separate transmissions. [14], [16], [22] belong to this category.

2. *Messages of bounded size*: In this model, nodes are allowed to send a combined message up to some limit (particularly up to $\log n$). [1], [5] belong to this category. In [1] nodes can combine up to $\log n$ messages.

3. *Messages of unbounded size*: In this model, nodes are allowed to send a combined message including all messages it has received so far. [26], [17] [15], [6], [8] belong to this category.

Although they all look alike, this little difference is vital in designing corresponding gossiping algorithms because each case has its own lower bound as well as advantages/disadvantages. The gossiping problem in these three cases should be treated as three fundamentally different problems. Note that our message model is (1), the unit-size model.

4. NAÏVE GOSSIPING ALGORITHM

We start off with a naïve gossiping algorithm, presented in Alg. 1. Its individual elements are presented separately in Algs. 2,3,4,5 later. There are three phases in the main algorithm: *Preprocessing*, *Data collection*, and *Naïve broadcast*.

Phase I: Preprocessing

In this phase, first we find the graph center defined as follows. A graph center s is a node in a graph G such that the length of the shortest path from s to the farthest node is minimized. Graph centers may not be unique (i.e. there can be more than one node satisfying this property). If more than one node has such a property, we simply pick any one. Graph centers can be obtained easily by applying the Floyd-Warshall algorithm [13]. Since it computes the lengths of shortest paths between every pair of nodes, we can simply look at the distance between every pair of nodes and find such a node. The details can be found in [10, 13]. Note that it's not a must to apply this algorithm. We can use any algorithm that finds the graph center to replace it. The Floyd-Warshall runs in $O(n^3)$ time, but this is the computation time complexity and it has nothing to to with our objective *latency*, which is the *transmission time complexity*.

Algorithm 1 Naïve Gossiping Algorithm

Input: $G = (V, E)$.
Output: Schedule $\mathcal{W} : \mathbb{N} \rightarrow 2^V$

▷ Phase I: Preprocessing

1: Find the graph center s by applying Floyd-Warshall algorithm.
2: Apply Alg. 2 to G, s to obtain the black nodes.
3: Apply Alg. 3 to G, s and $BLACK$ to obtain the blue nodes as well as constructing the broadcast tree T.

▷ Phase II: Data collection

4: Apply Alg. 4 to obtain a schedule $\mathcal{W}_1$ that collects data to s.

▷ Phase III: Naïve broadcast

5: Apply Alg. 7 to obtain a schedule $\mathcal{W}_2$ that broadcasts all data back to all nodes.

6: Combine $\mathcal{W}_1, \mathcal{W}_2$ to obtain the overall schedule $\mathcal{W}$.

Now we first construct a Breadth First Search (BFS) tree for the network and divide all nodes into layers (where the 0-th layer is s alone and the 1st layer is its neighbors). We then form a maximal independent set (referred to as the 'black' node set) $BLACK$ layer by layer (Alg. 2) as follows. Starting from the 0-th layer, we pick up a maximal independent set and mark these nodes black. We then move on to the 1st layer and pick up a maximal independent set and mark these nodes black again. Note that the black nodes of the 1st layer also need to be independent of those of the previous layer. We repeat this process until all layers have been worked on. Those who are not marked black are marked white at last. The pseudocode of layered MIS construction is given in Alg. 2.

Algorithm 2 Construct an MIS layer by layer

Divide all nodes into layers $L_0, L_1, L_2, \dots, L_l$
$BLACK \leftarrow \emptyset$
for $i \leftarrow 0$ to l **do**
  Find an MIS $BLACK_i \subset L_i$ also independent of $BLACK$
  $BLACK \leftarrow BLACK \cup BLACK_i$
end for
return $BLACK$

To construct the broadcast tree, we pick some of the white nodes and color them blue to interconnect all black nodes as follows. To connect the two adjacent layers, we look at the lower layer's black nodes. Each black node must have a parent on the upper layer and this parent node must be white since black nodes are independent of each other. We add this white parent for each black node on the lower layer and also add an edge between them. These added white nodes are colored blue. Moreover, we know that this blue node must be dominated by a black node either on the same layer (which is the upper layer) or on the layer above (which is the layer above the upper layer). We then add an edge between this blue node and its dominator. We repeat this process layer by layer starting from the o-th layer and finally obtain the desired broadcast tree in this manner. Note that, in this tree, each black node has a blue parent at the upper layer and each blue node has a black parent *at the same layer or the layer right next to it above*. The pseudocode is given in Alg. 3.

Algorithm 3 Broadcast tree construction

$T = (V_T, E_T), V_T = V, E_T \leftarrow \emptyset$
Connect s too all nodes in L_1
for $i \leftarrow 1$ to $l - 1$ **do**
  ▷ /* Connect $BLACK_{i+1}$ to $BLUE_i$ */
  for all black nodes $v \in BLACK_{i+1}$ **do**
    Find its parent $p(v)$ in BFS tree
    Add $p(v)$ to $BLUE_i$
    Add an edge between $p(v), v$ to E_T
  end for
  ▷ /* Connect $BLUE_i$ to $p(BLUE_i)$ */
  for all blue nodes $w \in BLUE_i$ **do**
    Find w's dominator d_w in $p(BLUE_i)$
    Add an edge between d_w, w to E_T
  end for
end for
▷ /* Connect remaining white nodes */
for all remaining white nodes u **do**
  Find u's dominator d_u
  Add an edge between u, d_u to E_T
end for
return T

Phase II: Data collection

The algorithm in this phase is based on [12]. We modified their algorithm slightly to fit our scenario here. In this phase, each node has a message to transmit and all messages are relayed to s. We can use a simple interleaving algorithm as follows. Layers are grouped according to their layer number modulus 3. For example, $L_1, L_4, L_7, \dots$ are one group, $L_2, L_5, L_8, \dots$ are another group, and $L_3, L_6, L_9, \dots$ are the third group. In each time slot, exactly one of these three groups transmit or forward a message and these three groups take turns to be the transmitting group. In each time slot, we pick up a node from each layer in the transmitting group and schedule them to transmit concurrently. For example, each of $L_1, L_4, L_7, \dots$ has one node transmitting in time slot 0, each of $L_2, L_5, L_8, \dots$ has one node transmitting in time slot 1, each of $L_3, L_6, L_9, \dots$ has one node transmitting in time slot 2, and so on so forth. Later we will show that there will be no collision at all and this phase terminates after $3(n - 1)$ time slots. The pseudocode is presented in Alg. 4.

Phase III: Naïve broadcast

We simply apply the EBS algorithm in [18] repeatedly every 48 time slots until all packets have been released. The EBS schedule is basically a greedy first-fit algorithm.

In § 7, we will show that the naïve gossiping algorithm has latency $48(N + R - 2) + 1$ (Lemma 7.4) and approximation ratio 51 (Theorem 7.1). They can be further reduced in the next section.

5. INTERLEAVED GOSSIPING ALGORITHM

The naïve gossiping algorithm has approximation ratio 51. Although it's already an improvement of [14], we can further reduce the latency by interleaving those broadcasts to the fullest extent. We present our interleaved gossiping algorithm in Alg. 6, which simply replaces the naïve broadcast (step 5 in Alg. 1) by the interleaved broadcast (Alg. 7).

Interleaved broadcast

First we color all black nodes by applying the smallest-degree-last ordering in G^2 as described in [18]. We need at most 12 colors according to [18]. In this algorithm, we always use 12 colors even if we could use less. In some cases, even if we may use less than

Algorithm 4 Data Collection

Input: $G = (V, E)$, $s \in V$.
Output: Schedule $\mathcal{W} : \mathbb{N} \rightarrow 2^V$

$t \leftarrow 0$.
repeat
  for all $i \equiv 1 \mod 3$ **do** pick a node $x_i \in L_i$ that either needs to transmit or forward a message. Set $\mathcal{W}(t) \leftarrow \{x_i | i \equiv 1 \mod 3\}$.
  end for
  $t \leftarrow t + 1$
  for all $i \equiv 2 \mod 3$ **do** pick a node $x_i \in L_i$ that either needs to transmit or forward a message. Set $\mathcal{W}(t) \leftarrow \{x_i | i \equiv 2 \mod 3\}$.
  end for
  $t \leftarrow t + 1$
  for all $i \equiv 0 \mod 3$ **do** pick a node $x_i \in L_i$ that either needs to transmit or forward a message. Set $\mathcal{W}(t) \leftarrow \{x_i | i \equiv 0 \mod 3\}$.
  end for
  $t \leftarrow t + 1$
until all nodes have finished transmitting and forwarding
return $\mathcal{W}$

Algorithm 5 Naïve Broadcast

Input: $G = (V, E)$, $s \in V$.
Output: Schedule $\mathcal{W} : \mathbb{N} \rightarrow 2^V$

Starting from $t \leftarrow 0$, schedule s to release a new message every 48 time slots by applying the EBS algorithm in [18] until all n messages have been released. All relay transmissions of different packets released by s are executed in an interleaving manner.
return $\mathcal{W}$

Algorithm 6 Interleaved Gossiping Algorithm

Both Phase I and Phase II are exactly the same as Alg. 1. Replace Phase III (line 5 in Alg. 1) by the *Interleaved broadcast* described in Alg. 7.

Algorithm 7 Interleaved Broadcast

Input: $G = (V, E)$, $s \in V$.
Output: Schedule $\mathcal{W} : \mathbb{N} \rightarrow 2^V$

Color all black nodes by applying the smallest-degree-last ordering in G^2 using 12 colors as described in [18].
Starting from $t \leftarrow 0$, schedule s to release a new message every 24 time slots until all n messages have been released.
▷ /* The following part is executed in a way that the transmissions of different messages are all interleaved. A node is scheduled to transmit only if it has received a message but hasn't yet finished relaying it. If a node has no message to relay, it simply idles.*/
repeat the following in an interleaving manner
  For all $1 \leq t' \leq 12$, $\mathcal{W}(t + t') \leftarrow \{x \in BLACK | color(x) = t'\}$
  For all $i \equiv 1 \mod 3$, apply IMC with $X = BLUE_i$, $Y = children(BLUE_i)$ and starting time $t + 13$.
  For all $i \equiv 2 \mod 3$, apply IMC with $X = BLUE_i$, $Y = children(BLUE_i)$ and starting time $t + 17$.
  For all $i \equiv 0 \mod 3$, apply IMC with $X = BLUE_i$, $Y = children(BLUE_i)$ and starting time $t + 21$.
  $t \leftarrow t + 24$
until every node has received n messages
return $\mathcal{W}$

Algorithm 8 Iterative Minimal Covering (IMC)

Input: Input graph $G = (V, E)$, vertex subsets $X, Y \subset V$ such that X is a cover of Y in G, and starting time t_S.
Output: Schedule $\mathcal{W} : \mathbb{N} \rightarrow 2^V$

$l \leftarrow 0, X_0 \leftarrow X, Z \leftarrow Y$
repeat
  $l \leftarrow l + 1$
  Find a minimal cover $X_l \subset X_{l-1}$ of Z.
  $W_{l-1} \leftarrow X_{l-1} - X_l$
  $Z \leftarrow Z \setminus Inf(X_l)$, where $Inf(X_l) = \{x \in V - X_l | x$ hasexactly one neighbor in $X_l\}$
until $Z = \emptyset$
Set $W_l \leftarrow X_l$
Set $\mathcal{W}(t_S + i - 1) \leftarrow W_i, \forall\, 1 \leq i \leq l$
return $\mathcal{W}$

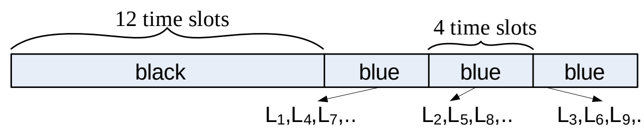

Figure 1: Group 24 time slots in which the first 12 slots are for black nodes' transmission and the remaining 12 slots are divided into three groups for blue nodes' transmission.

12 colors, we still apply a 12-coloring in which some colors never appear.

Now we group every 24 time slots as one unit as shown in Fig. 1. Black nodes only transmit in the first 12 slots, and blue nodes only transmit in the last 12 slots but in a 3-interleaving manner. (The actual transmission schedule within those 12 time slots will be defined later shortly.) That is to say, in every 24-slot unit, time slots 13 – 16 are scheduled only for those blue nodes at layer i such that $i \equiv 1 \mod 3$. Similarly time slots 17 – 20 are scheduled only for those blue nodes at layer i such that $i \equiv 2 \mod 3$, and time slots 21 – 24 are scheduled only for those blue nodes at layer i such that $i \equiv 0 \mod 3$, as shown in Fig. 2.

Within the given 12 time slots, a black node simply transmit according to its color. Since we applied a 12-coloring, we need exactly 12 time slots. For example, if a black node has color 7, its the actual transmission schedule is the 7th slot. For a blue node, we need to apply the *Iterative Minimal Covering* (IMC) algorithm to the broadcast tree T to determine their actual transmission schedule in those given 4 time slots. The pseudocode of IMC is given in Alg. 8. It is known that we need at most 4 time slots [18]. The details of IMC can also be found in [18].

The interleaved broadcast schedule is defined as follows. The source simply transmits a packet every 24 time slots. Any non-source nodes' relaying schedule is determined according to its color and layer as described in the 24-slot unit, as shown in Fig. 2. We simply follow this rule until all nodes have received n messages. Note that, in a 24-slot unit each black and blue node transmits exactly once.

6. AN EXAMPLE

Here in this section we present an example of the interleaved gossiping algorithm. Consider a network topology shown in Fig. 3(a). We divide all nodes into layers, as shown in Fig. 3(b). Then we construct the MIS layer by layer as shown in Fig. 3(c). In the first step, s is selected in the MIS and colored black. In the second step, since the source is black, all nodes at layer 1 must be white, otherwise it won't be independent of s. In the third step, we select an MIS b_2, d_2, e_2, f_2, h_2 at layer 2, which must also be independent of

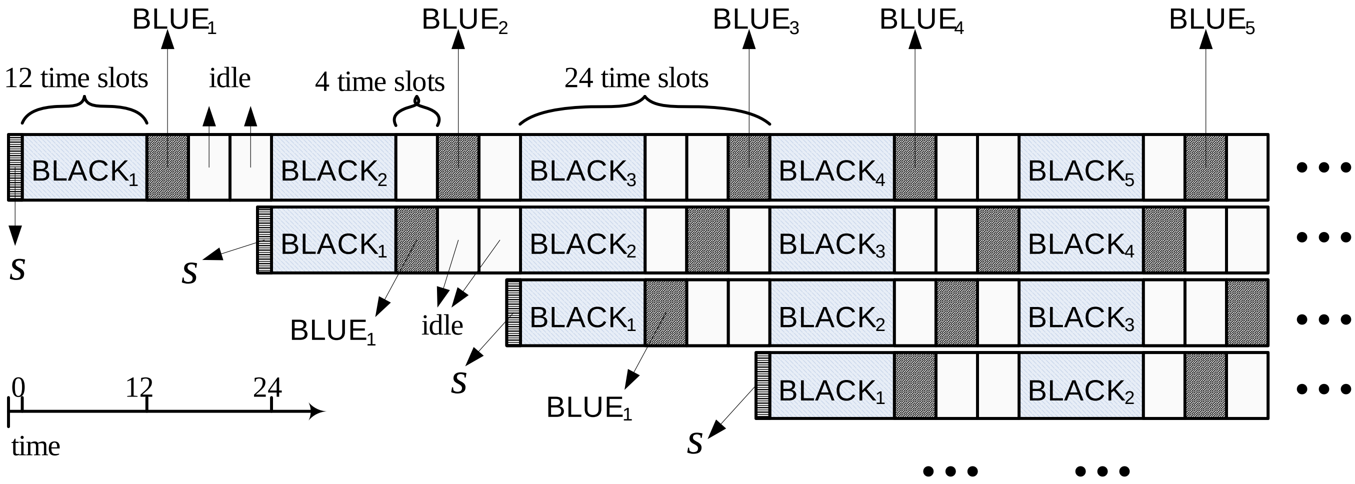

Figure 2: Interleaved broadcast scheduling. Blank blocks represent idling time slots. This figure clearly shows that s releases 4 packets in the first 96 time slots and these 4 broadcasts are interleaved.

the black nodes of the previous layer, $BLACK_1$, although there is no black node at layer 1 and this does not have any effect. Fig. 3(c) shows that b_2, d_2, e_2, f_2, h_2 are colored black at layer 2. We keep doing this and select black nodes until all layers have been worked on this way. The black node selection depends solely on G and there is nothing to do with the BFS tree. Not until all blue nodes have been selected do we need to consider the BFS tree, as shown in Fig. 4(a). In Alg. 3, we are trying to add appropriate blue nodes to interconnect all black ones. Since the source s does not have an upper layer and there are no black nodes at layer 1, we start from layer 2 directly. For each black node v at layer 2, we find its parent $p(v)$, color $p(v)$ blue, and connect $v, p(v)$ in the BFS tree, as shown in Fig. 4(b).

In Fig. 4(b) we see b_1, c_1, d_1, e_1 at layer 1 are colored blue and connected to some black nodes at layer 2. a_1, f_1 are not colored blue, so they remain white. We also connect $a_1, b_1, c_1, d_1, e_1, f_1$ to s since they are dominated by s. We keep working this way on layer 3. For simplicity, suppose we've already found the black nodes at layer 3 and their corresponding blue nodes at layer 2. Fig. 3(d) shows that there are 3 blue nodes (a_2, c_2, i_2) at layer 2 connected to their black children at layer 3, g_2 remains white. So far, we have

$$\begin{cases} L_2 = \{a_2, b_2, c_2, d_2, e_2, f_2, g_2, h_2, i_2\} \\ BLACK_2 = \{b_2, d_2, e_2, f_2, h_2\} \\ BLUE_2 = \{a_2, c_2, i_2\} \\ WHITE_2 = \{g_2\} \end{cases}$$

Now, for each blue or white node at layer 2, we know that it must be adjacent to at least one black node either at layer 2 or layer 1, since $BLACK_2$ it is a maximal independent set. Because of it maximality, each nodes of L_2 must be adjacent to at least one black node in $BLACK_1$ or $BLACK_2$. However, since $BLACK_1 = \emptyset$, each node of $BLUE_2$ or $WHITE_2$ must be adjacent to at least one node in $BLACK_2$, as shown in Fig. 4(b). We keep doing this for all layers and the broadcast tree will be constructed in this way.

An example of data collection is described here. Since Alg. 4 only needs layer numbers to do scheduling, whether a node is black, blue, or white makes no difference and we only need to consider the topology in Fig. 3(a), and layer information in in Fig. 3(b). Nodes are named in the following way. Note that, except for s, we use the subscript to represent the layer. For example, at the first layer we have $a_1, b_1, \ldots, f_1$. According to Alg. 4, we randomly pick up a node in $L_1 = \{a_1, b_1, \ldots, f_1\}$, say a_1, to transmit in time slot 0. We also randomly pick up a node in $L_4, L_7, \ldots$ (not shown in Fig. 4)(a) and schedule them to transmit concurrently. Then we randomly pick up a node, say a_2, in L_2 to transmit in time slot 1. We also pick up a node in $L_5, L_8, \ldots$ (not shown in Fig. 4(a)) and schedule them to transmit concurrently. We just follow this method until all nodes have finished transmitting and forwarding.

Finally we present an example of interleaved broadcast scheduling. First we consider G^2 and give a 12-coloring to all black nodes by applying the smallest-degree-last ordering. Fig. 4(c) shows (part of) G^2 with colors and degrees. Numbers in parentheses represent degrees, and numbers without parentheses represent colors. Since white nodes are not scheduled, we can ignore them. According to Alg. 7, s is scheduled to transmit in time slot 0 and all nodes in L_1 will receive the message collision-free. Then from time slot 1 to 12, all nodes will be idling since there is no black node in L_1. The transmission between $BLUE_1$ and $BLACK_2$ will be scheduled from 13 to 16, and 17 – 24 are idling slots according to Alg. 7. The actual transmission slots of during 13 – 16 will be determined according to Alg. 8 (IMC). Fig. 4(d) shows an example of how to determine those actual transmission slots. We run IMC with $X = BLUE_1$ and $Y = BLACK_2$ as follows. X_0 is set to X and Z is set to Y. Now the minimal cover of Z will be X_0 itself (in this particular case only!), so $X_1 = X_0$ and $W_0 = X_0 - X_1 = \emptyset$. However, $Inf(X_1) = \{b_2, e_2, f_2, h_2\}$ and Z is therefore reset to $\{d_2\}$. In the second iteration, the minimal covering of Z can either be $\{b_1\}$ or $\{c_1\}$, so we arbitrarily pick $X_2 = \{c_1\} \subset X_1$. Therefore, $W_1 = X_1 - X_2 = \{b_1, d_1, e_1\}$, $Inf(X_2) = \{d_2, e_2\}$, and $Z = \{d_2\} \setminus Inf(X_2) = \emptyset$. Now, $Z = \emptyset$ and $l = 2$, so we stop, set $W_2 = X_2$, and get the following schedule $\mathcal{W}$ such that $\mathcal{W}(13) = W_1$, $\mathcal{W}(14) = W_2$, and $\mathcal{W}(15) = \mathcal{W}(16) = \emptyset$. In other words, b_1, d_1, e_1 are schedule to transmit in slot 13, c_1 is scheduled to transmit in slot 14, and 15,16 are both idling slots.

Now, we schedule the black nodes from slot 25 to slot 36 according to their color. As shown in Fig. 4(b), $BLACK_2 = \{b_2, d_2, e_2, f_2, h_2\}$ where *color*$(b_2) = \#4$, *color*$(d_2) = \#6$, *color*$(e_2) = \#7$, *color*$(f_2) = \#10$, and *color*$(h_2) = \#12$. Therefore, b_2, d_2,

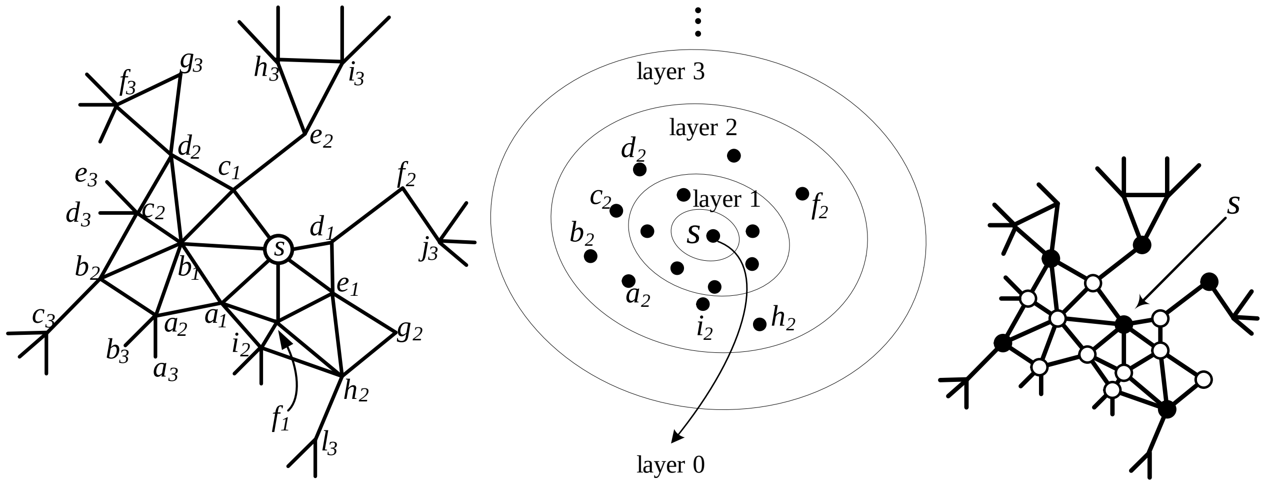

Figure 3: (a) G's topology (b) Layers of G (c) Layered MIS

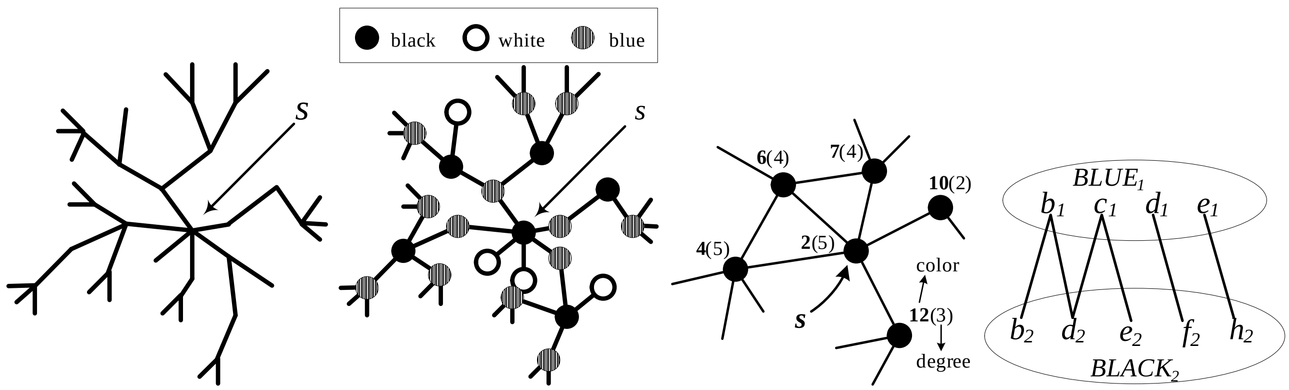

Figure 4: (a) BFS Tree (b) Broadcast Tree (c) Smallest-degree-last coloring in G^2 (d) IMC

Table 1: Interleaved broadcast schedule within time slots 0-36

slot(s)	0	1-12	13	14	15,16	17-20	21-24
node(s)	s	-	b_1, d_1, e_1	c_1	-	-	-

slot(s)	25-27	28	29	30	31	32,33	34	35	36
node(s)	-	b_2	-	d_2	e_2	-	f_2	-	h_2

e_2, f_2, h_2 are scheduled to transmit in time slots 28,30,31,34,36, respectively, and thus 25,26,27,29,32,33,35 are idling slots. To summarize this example, the overall interleaved broadcast schedule from time slot 0 to 36 is shown in Table 1. In this table, '-' represents idling time slots.

7. ANALYSIS

PROPERTY 7.1. *In the broadcast tree T, the parent of a black node must be blue, and the parent of a blue node must be black. The parent of a node in $BLACK_i$ must be in $BLUE_{i-1}$, but the parent of a node in $BLUE_i$ may be in $BLACK_{i-1}$ or $BLACK_i$. (Note that i denotes the layer of the BFS tree, not of the broadcast tree.)*

LEMMA 7.1. *Collision will not happen in the data collection schedule define in Alg. 4.*

(*Proof.*) If two nodes $u \in L_i$ and $v \in L_j$ satisfy $|i-j| > 2$, then there will be no collision between u, v for the following reason. If collision happens, then there exists a node w adjacent to both u and v, which contradicts to $|i-j| > 2$. Alg. 4 is designed in the 3-interleaving fashion such that in any time slot, within any three consecutive layers, there can be at most one node transmitting. For this reason, collision will never happen. □

LEMMA 7.2. *The data collection schedule define in Alg. 4 has latency* $3(N-1)$.

(*Proof.*) According to Alg. 4, s receives a message every three time slots. Moreover, there are $N-1$ nodes that has a message to be sent to s (excluding s itself). Therefore, after $3(N-1)$ all messages will be received by s and the data collection schedule terminates. □

LEMMA 7.3. *Collision will not happen in the naïve broadcast schedule define in Alg. 5.*

(*Proof.*) The naïve broadcast algorithm applied the EBS algorithm in [18], so a single execution of EBS will not cause any collision according to [18]. The naïve broadcast algorithm calls EBS as a subroutine every 48 time slots, which means each call of EBS will be separated by at least 3 layers according to the properties of EBS. Therefore, collision will not happen. □

LEMMA 7.4. *The naïve broadcast schedule define in Alg. 5 has latency* $48(N + R - 2) + 1$.

(*Proof.*) The broadcast of first message will be completed in time slot $1 + 48(R - 1)$. Since the broadcast of all messages are interleaved, and each message is released every 48 time slots, the following $N - 1$ messages will arrive every 48 time slots. Therefore the last message will arrive after $48(N - 1)$ time slots and the latency of Alg. 5 becomes $48(N + R - 2) + 1$. □

LEMMA 7.5. $N + R - 1$ *is a lower bound for the gossiping problem.*

(*Proof.*) First we claim that at least one node should transmit R times for the following reason. There are N messages. The broadcasting of each message requires at least R transmissions, so the total number of transmissions is at least $N * R$. Hence, at least one node transmit R times.

Each node has to receive $N - 1$ times. For a node that transmits at least R times, it needs at least $N + R - 1$ time slots. Therefore, we found $N + R - 1$ is a lower bound for the gossiping problem. □

THEOREM 7.1. *The approximation ratio of the naïve gossiping algorithm (Alg. 1) is at most 51.*

(*Proof.*) According to Lemmas 7.2 and 7.4, we get a combined latency of $3(N - 1) + 48(N + R - 2) + 1 < 51 * (N + R - 1)$. According to Lemma 7.5, $N + R - 1$ is a lower bound. Therefore, the approximation ratio is clearly at most 51. □

LEMMA 7.6. *Collision will not happen in the interleaved broadcast schedule define in Alg. 7.*

(*Proof.*)

1. Black nodes do not cause collision to any blue nodes. This is because the first 12 time slots within a 24-slot round are reserved for black nodes only. No blue nodes are schedule to transmit during these 12 slots and no collision will happen.

2. Black nodes do not cause collision to each other. This can be proved according to the 12-coloring property. A black node is scheduled to transmit according to its color. Therefore, any two concurrently transmitting black nodes must have the same color, and according to the geometrical property of the 12-coloring the distance between any such pair of nodes must be at least 2. If collision happens, concurrently transmitting black nodes must have a common neighbor, which is a contradiction. More details regarding the 12-coloring can be found in [18].

3. Blue nodes do not cause collision to any black nodes. This is because the last 12 time slots within a 24-slot round are reserved for blue nodes only. No black nodes are schedule to transmit during these 12 slots and no collision will happen.

4. Blue nodes do not cause collision to each other. Consider two blue nodes $u \in BLUE_i$ and $v \in BLUE_j$. If $|i - j| > 2$ there will be no collision because u, v cannot have a common neighbor. If $|i - j| \leq 2$ and $i \neq j$, then there will be no collision because transmissions are all interleaved for different layers and u, v will be scheduled to transmit in different time slots according to Alg. 7. Finally, if $i = j$ there will be no collision either because blue nodes at the same layer are scheduled according to IMC (Alg. 8). The detailed proof of this part (that IMC does not cause any collision) can be found in [18]. □

LEMMA 7.7. *The interleaved broadcast schedule define in Alg. 7 has latency* $24(N + R - 2) + 1$.

(*Proof.*) The broadcast of first message will be completed in time slot $1 + 24(R - 1)$. Since the broadcast of all messages are interleaved, and each message is released every 24 time slots, the following $N - 1$ messages will arrive every 24 time slots. Therefore the last message will arrive after $24(N - 1)$ time slots and the latency of Alg. 7 becomes $24(N + R - 2) + 1$. □

THEOREM 7.2. *The approximation ratio of our interleaved gossiping algorithm (Alg. 6) is at most 27.*

(*Proof.*) According to Lemmas 7.2 and 7.7, we get a combined latency of $3(N - 1) + 24(N + R - 2) + 1 < 27 * (N + R - 1)$. According to Lemma 7.5, $N + R - 1$ is a lower bound. Therefore, the approximation ratio is clearly at most 27. □

8. CONCLUSION AND FUTURE WORK

We studied the minimum-latency gossiping problem in multi-hop wireless networks. We first presented a naïve gossiping scheme that achieved approximation ratio 51, and then we improved it by introducing the interleaved gossiping algorithm that has ratio 27. In both algorithms, we took great advantages of UDG's geometrical properties. However, the UDG model does not completely reflect the reality, and the gossiping problem can be re-investigated in more practical models. For example, we can consider the 2-disk model, in which the transmission range is distinguished from the interference range (usually the interference range is some 2-5 times larger). In this model, we still assume both transmission and interference ranges are disks and use two radii to represent them. We believe that our algorithms can be extended to this model by relaxing the approximation ratios as follows. We still find MIS and construct broadcast trees according to the transmission radius. However, in all interleaved transmissions, instead of separating each transmission by three hops, we need to enlarge this hop distance correspondingly according to the interference radius. This way we can still get reasonably good gossiping algorithm with a slightly large approximation ratio. This ratio may not be a constant anymore as compared to Alg. 6 with ratio 27. This ratio may depend on the interference radius, since obviously large interference radius may increase this ratio.

Another network model we can consider for a possible future extension is the signal-to-noise ratio model (SNR). This model may not be an easy extension anymore, as it's not a deterministic model and randomness is involved. In this case, we believe that we can still take some advantage of its geometrical properties as we did in this work, but we need to add some randomness to it.

Finally, as another future work, we want to design distributed version of this gossiping algorithm. This work is basically centralized, as we assume the full topology is known and both the broadcast tree and scheduling algorithms needs topology information. However, this assumption may be relaxed, since in all algorithms of this work we only need topology information of neighboring layers. That means, we only need local topology information. For this reason, we believe our gossiping algorithm can be designed in a distributed fashion.

9. REFERENCES

[1] R. Bar-Yehuda, A. Israeli, and A. Itai. Multiple communication in multihop radio networks. *SIAM Journal on Computing*, 22(4):875–887, 1993.

[2] J. C. Bermond, L. Gargano, A. A. Rescigno, and U. Vacarro. Fast gossiping by short messages. In *ICALP'95: the 22th International Colloquium on Automata, Languages and Programming*, pages 135–146, 1995.

[3] J.-C. Bermond, L. Gargano, A. A. Rescigno, and U. Vaccaro. Fast gossiping by short messages. *SIAM Journal on Computing*, 27(4):917–941, 1998.

[4] B. S. Chlebus, L. Gasieniec, A. Lingas, and A. T. Pagourtzis. Oblivious gossiping in ad-hoc radio networks. In *DIALM'01: the 5th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 44–51, 2001.

[5] M. Christersson, L. Gasieniec, and A. Lingas. Gossiping with bounded size messages in ad hoc radio networks. In *ICALP'02: the 29th International Colloquium on Automata, Languages and Programming*, pages 377–389, London, UK, 2002. Springer-Verlag.

[6] M. Chrobak, L. Gasiniec, and W. Rytter. Fast broadcasting and gossiping in radio networks. In *FOCS'00: the 41st IEEE Symposium on Foundation of Computer Science*, pages 575–581, 2000.

[7] M. Chrobak, L. Gasiniec, and W. Rytter. A randomized algorith for gossiping in radio networks. In *COCOON'01: the 7th Annual International Computing and Combinatorics Conference*, pages 483–492, 2001.

[8] M. Chrobak, L. Gasiniec, and W. Rytter. ast broadcasting and gossiping in radio networks. *Journal of Algorithms*, 43(2):177–189, 2002.

[9] A. E. F. Clementi, A. Monti, and R. Silvestri. Selective families, superimposed codes, and broadcasting in unknown radio networks. In *SODA'01: the 12th ACM-SIAM Symposium on Discrete Algorithms*, pages 709–718, 2001.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*, chapter 25. McGraw-Hill, 2nd edition, 2001.

[11] M. Flamini and S. Perennes. Lower bounds on the broadcasting and gossiping time of restricted protocols, 1999. Technical Report, INRIA.

[12] C. Florens, M. Franceschetti, and R. J. McEliece. Lower bounds on data collection time in sensory networks. *IEEE Journal on Selected Areas in Communications*, 22(6).

[13] R. W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6).

[14] R. Gandhi, S. Parthasarathy, and A. Mishra. Minimizing broadcast latency and redundancy in ad hoc networks. In *MobiHoc'03: the ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 222–232, 2003.

[15] L. Gasieniec and A. Lingas. On adaptive deterministic gossiping in ad hoc radio networks. *Information Processing Letters*, 83:89–93, 2002.

[16] L. Gasieniec and I. Potapov. Gossiping with unit messages in known radio networks. In *TCS'02: the IFIP 17th World Computer Congress - TC1 Stream / 2nd IFIP International Conference on Theoretical Computer Science*, pages 193–205, Deventer, The Netherlands, The Netherlands, 2002. Kluwer, B.V.

[17] L. Gasieniec, I. Potapov, and Q. Xin. Time efficient gossiping in known radio networks. In *SIROCCO'04: the 11th Colloquium on Structural Information and Communication Complexity*, pages 173–184, 2004.

[18] S. C.-H. Huang, P.-J. Wan, X. Jia, H. Du, and W. Shang. Minimum-latency broadcast scheduling in wireless ad hoc networks. In *INFOCOM'07: the 26th Annual IEEE Conference on Computer Communications*, pages 733–739, 2007.

[19] P. Indyk. Explicit constructions of selectors and related combinatorial structures, with applications. In *SODA'02: the 13th ACM-SIAM Symposium on Discrete Algorithms*, pages 697–704, 2002.

[20] X. Jia. A distributed algorithm of delay bounded multicast routing for multimedia applications in wide area networks. *IEEE/ACM Trans. on Networking*, 6(6).

[21] D. Li, X. Jia, and H. Liu. Energy efficient broadcast routing in ad hoc wireless networks. *IEEE Trans on Mobile Computing*, 3(2).

[22] F. Manne and Q. Xin. Optimal gossiping with unit size messages in known topology radio networks. In *CAAN'06: the 3rd Workshop of Combinatorial and Algorithmic Aspects of Networking*, pages 125–134, Berlin/Heidelberg, 2006. Springer.

[23] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom'99: the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162, New York, NY, USA, 1999. ACM Press.

[24] K. Ravishankar and S. Singh. Asymptotically optimal gossiping in radio networks. *Discrete Applied Mathematics*, 61:61–85, 1995.

[25] K. Ravishankar and S. Singh. Gossiping on a ring with radios. *ParallelProcessing Letters*, 6(1):115–126, 1996.

[26] Y. Xu. An $O(n^{1.5})$ deterministic gossiping algorithm for radio networks. *Algorithmica*, 36(1).