

Interference-Aware Gossiping Scheduling in Uncoordinated Duty-Cycled Multi-hop Wireless Networks

Xianlong Jiao^{*†}, Wei Lou[†], Xiaodong Wang^{*}, Junchao Ma[†], Jiannong Cao[†], Xingming Zhou^{*}

^{*}School of Computer, National University of Defense and Technology, Changsha, China

[†]Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong

Email: {xljiao, xdwang, xmzhou}@nudt.edu.cn, {csweilou, csjma, csjcao}@comp.polyu.edu.hk

Abstract

Gossiping is to broadcast the message of every node to all the other nodes in multi-hop wireless networks (MWNs). This operation plays an important role and is widely used in MWNs. Interference-aware gossiping scheduling (IAGS) aims to provide an interference-free scheduling for gossiping with the minimum latency. Previous work on IAGS mostly assumes that nodes are always active, and thus is not suitable for duty-cycled scenarios. In this paper, we investigate the IAGS problem in uncoordinated duty-cycled multi-hop wireless networks (IAGS-UDC problem) under protocol interference model and unbounded-size message model. We prove that the IAGS-UDC problem is NP-hard. We propose a novel approximation algorithm called MILD for this problem. The MILD algorithm achieves an approximation ratio of $3\beta^2(\Delta + 6)|T|$, where β is $\lceil \frac{2}{3}(\alpha + 2) \rceil$, α denotes the ratio of the interference radius to the transmission radius, Δ denotes the maximum node degree of the network, and $|T|$ denotes the number of time-slots in a scheduling period. Moreover, the number of transmissions scheduled by the MILD algorithm is at most 3 times as large as the minimum number of transmissions.

Keywords: Gossiping scheduling; interference; duty cycle; multi-hop wireless networks.

I. INTRODUCTION

Uncoordinated duty-cycled multi-hop wireless networks (UDC-MWNs) consist of nodes with limited transmission ranges and different duty cycles. Nodes in UDC-MWNs are powered by batteries, so they often switch between the active state and the sleep state to conserve the energy, and this switching operation is uncoordinated. Broadcast is one of the most important communications in UDC-MWNs, and one kind of the broadcast communications is gossiping, which is to broadcast the message of every node to all the other nodes. Gossiping is widely used in UDC-MWNs for data collection and code update, etc. In many time-critical applications of UDC-MWNs, gossiping is required to be completed with low latency.

There are many interference models in UDC-MWNs, such as *graph-based interference model* and *protocol interference model*. Under the graph-based interference model, the interference is treated as the collision, and if two nodes send messages to their common neighboring node concurrently, the common neighboring node will receive neither of the two messages. Under the protocol interference model, if one node lies in the interference range of one transmitter node, it cannot receive the messages from other nodes when this transmitter node is transmitting messages. Two common message models in UDC-MWNs are *unit-size message model* and *unbounded-size message model*. Under the unit-size message model, one node cannot combine its received messages as one message. Under the unbounded-size message model, one node can combine its received messages as one message, and can broadcast the combined message in one time-slot.

Interference aware gossiping scheduling (IAGS) aims to provide an interference-free scheduling for gossiping with the minimum latency. The IAGS problem in conventional multi-hop wireless networks (MWNs) is known to be NP-hard no matter whether the networks are modeled as general graphs [1] or unit disk graphs [2]. Many efficient approximation algorithms [2]–[7], which follow the assumption that all nodes always keep active, have been presented for this problem. Unlike in conventional MWNs, one node in UDC-MWNs may require transmitting several times to inform all its neighboring nodes with different active time. Hence, these algorithms are not suitable for the IAGS problem in UDC-MWNs.

In this paper, we investigate the IAGS problem in UDC-MWNs (IAGS-UDC problem) under both protocol interference model and unbounded-size message model. To the best of our

knowledge, this is the first work to study this problem under these two models. Our main contributions include: 1) We prove that the IAGS-UDC problem is NP-hard; 2) We propose a novel approximation algorithm called MILD for this problem; 3) We show the correctness of the MILD algorithm, and prove that the approximation ratio of this algorithm is at most $3\beta^2(\Delta+6)|T|$, where β is $\lceil \frac{2}{3}(\alpha + 2) \rceil$, α denotes the ratio of the interference radius to the transmission radius, Δ denotes the maximum node degree of the network, and $|T|$ denotes the number of time-slots in a scheduling period; 4) We prove that the number of transmissions scheduled by the MILD algorithm is at most 3 times as large as the minimal number of transmissions.

II. RELATED WORK

Since gossiping plays a very important role in MWNs, a lot of studies have been done on this problem [1]–[9]. Gossiping is known as the all-to-all broadcast, and the simplest implementation of broadcast is flooding, which may cause a large number of contention and collision [8]. The multi-hop wireless network is often modeled as a unit disk graph (UDG) when the nodes have the same transmission radius. The IAGS problem, which aims to provide an interference-free gossiping scheduling with the minimum latency, is known to be NP-hard in both the general graphs [1] and the unit disk graphs [2].

Much work [3]–[5] has focused on the gossiping problem under graph-based interference model and unbounded-size message model. Chrobak et al. [3] have proposed a gossiping algorithm for unknown multi-hop radio networks, and shown that their algorithm can finish in $O(n^{3/2} \log^2 n)$ time. They have further presented a gossiping algorithm in [4], which is a randomized algorithm and can finish in $O(n \log^4 n)$ time. For networks with diameter $\delta = O(n^\epsilon)$ ($\epsilon > 1$), an efficient gossiping algorithm running in $O(\sqrt{\delta n} \log^2 n)$ is proposed in [5].

Many algorithms [2], [6], [7], [9] have been presented for the gossiping problem under unit-size message model. Recently, Gandhi et al. [2] have investigated the IAGS problem under graph-based interference model and unit-size message model, and proposed an approximation algorithm of a constant ratio. In [6], Huang et al. have shown that this ratio is more than 1000, and given a 27-approximation algorithm. This ratio has been further improved to 20 by Gandhi et al. in [7]. Wan et al. [9] have proposed a constant approximation algorithm to tackle the IAGS problem in multi-channel MWNs under protocol interference model and unit-size message model.

None of the work mentioned above, however, has taken the active/sleep cycles into consideration. The broadcast problems in duty-cycled scenarios have been extensively studied in [10]–[13]. The only work to study the IAGS-UDC problem is [13], which investigates this problem under the graph-based interference model and both two message models. To the best of our knowledge, none of previous work has focused on the IAGS-UDC problem under both protocol interference model and unbounded-size message model. In this paper, we will investigate this problem under these two models and give an efficient approximation algorithm for this problem.

III. PRELIMINARIES

A. Network Model

We model the uncoordinated duty-cycled multi-hop wireless network as a UDG $G = (V, E)$, where V contains all the nodes in the network, and E is the set of edges, which exist between any two nodes u and v if their Euclidean distance $d(u, v)$ is no larger than the transmission radius r . We consider protocol interference model as the interference model, and regard unbounded-size message model as the message model. We denote by r_f the interference radius, and by α the ratio of r_f to r . Every node cannot send and receive the message at the same time. We denote by n the number of nodes in the network and by $N_G(u)$ the set of neighboring nodes of node u .

We assume that nodes independently determine the active/sleep time in advance. The duty cycle is defined as the ratio of the active time to the whole scheduling time. The whole scheduling time is divided into multiple scheduling periods of the same length. One scheduling period T is further divided into fixed $|T|$ unit time-slots $\{0, 1, \dots, |T| - 1\}$. Every node v independently chooses one time-slot in T as its active time-slot $A(v)$. A node can transmit the message at any time-slot, but is only allowed to receive the message at its active time-slot.

B. Problem Formulation

This paper studies the gossiping problem in UDC-MWNs. In this IAGS-UDC problem, every node has a message to send to all the other nodes. The gossiping task completes when every node receives the messages from all the other nodes. We model the gossiping scheduling as assigning the transmitting time-slots for every node, i.e., assigning a function $TTS : V \rightarrow 2^N$. The objective of gossiping scheduling is to minimize the largest transmitting time-slot. It is easy to have the following result.

Lemma 1: The IAGS-UDC problem is NP-hard.

Proof: If we set $T = \{0\}$ and $\alpha = 1$, then all the nodes are active all the time, and the interference can be treated as the collision. In this case, the IAGS-UDC problem reduces to the conventional IAGS problem. The conventional IAGS problem has been proved to be NP-hard in [2], so the IAGS-UDC problem is NP-hard. ■

To schedule the transmissions efficiently, we construct a shortest path tree as follows. If we choose one node w as the source node and this node starts broadcasting its message at time-slot 0, the latency $Lat(u, v)$ of every edge $(u, v) \in E$ is:

$$Lat(u, v) = \begin{cases} A(v) + 1, & \text{if } u = w; \\ A(v) - A(u), & \text{if } u \neq w \text{ and } A(v) - A(u) > 0; \\ |A(v) - A(u) + |T||, & \text{else} \end{cases} \quad (1)$$

The shortest path tree rooted at node w can be achieved by applying Dijkstra's algorithm with this latency. The broadcast tree is constructed based on the shortest path tree, and the gossiping is scheduled according to the broadcast tree. To distinguish the parent nodes of node v in the shortest path tree and in the broadcast tree, we call the parent node of node v in the shortest path tree as the father node of node v , and denote it by $F(v)$; we denote by $P(v)$ the parent node of node v in the broadcast tree.

C. Graph-Theoretic Definitions

We denote by $G[U]$ the subgraph of G induced by a subset U of V . If there is no edge between any two nodes in $G[U]$, we call the subset U an Independent Set (IS) of G . A Maximal Independent Set (MIS) of G is not a subset of any other IS of G . It is known that a node can be adjacent to at most 5 nodes in an IS. A proper tessellation of hexagons in the whole plane is to partition the plane into hexagons of the same size. Coloring of these hexagons is to assign every hexagon one natural number representing the color of this hexagon. According to [14], a proper $3\beta^2$ coloring of half-open half-closed hexagons can make sure that the distance between two hexagons of the same color is larger than $3\beta - 2$ radii of the hexagon. If we set β as $\lceil \frac{2}{3}(\alpha + 2) \rceil$ and set the radius of the hexagon as $r/2$, the distance between two hexagons of the same color will be larger than $(\alpha + 1)r = r_f + r$.

IV. APPROXIMATION ALGORITHM

Since the IAGS-UDC problem is NP-hard, we propose and detail the MILD algorithm in this section. Recall that one node can combine its received messages as one message and send the combined message in one time-slot. Therefore, the MILD algorithm contains two processes. In the first process, the messages of all the nodes are gathered to a special node, which is called as a data aggregation process. In the second process, the special node combines all the messages as one message, and broadcasts this message to all the other nodes. The pseudocode of the MILD algorithm is shown in Algorithm 1.

The MILD algorithm starts with coloring all the nodes. We use a proper tessellation and $3\beta^2$ -coloring of hexagons with a radius of $r/2$ in the whole area to color these nodes, where $\beta = \lceil \frac{2}{3}(\alpha + 2) \rceil$. We use $f : V \rightarrow \{1, 2, \dots, 3\beta^2\}$ to denote this coloring method. After coloring all the nodes, we find a special node s . The maximum latency D of the shortest path tree T_{SPT} rooted at this node is the minimum. We can build the shortest path trees rooted at all the nodes based on the latency defined in Eq. 1, and find this special node. The tie can be broken randomly. Then all the nodes are divided into different layers L_0, L_1, \dots, L_D according to the latency of the shortest paths from node s to these nodes in T_{SPT} .

Next we construct the MIS'es layer by layer. The nodes in $V \setminus \{s\}$ are partitioned into different subsets $U_0, U_1, U_2, \dots, U_{|T|-1}$ according to their active time-slots. Recall that every node v in $V \setminus \{s\}$ can only receive the message at its active time-slot. The latency of the shortest path from node s to this node should be in the form of $k|T| + A(v) + 1$, where $k = 0, 1, 2, \dots$. So we can find that each subset U_j consists of nodes at several layers in the T_{SPT} , i.e., $U_j = \bigcup_{i \in I} L_i$, where $I = \{i | (i - 1) \equiv j \pmod{|T|}, 1 \leq i \leq D\}$. At each layer L_i , we find the independent set M_i of $G[L_i]$ such that $Q_j \cup M_i$ is an MIS of $G[\bigcup_{i' \in I'} L_{i'}]$, where $I' = \{i' | (i' - 1) \equiv j \pmod{|T|}, 1 \leq i' \leq i\}$, and $j = (i - 1) \pmod{|T|}$. Finally, we can find the MIS Q_j of $G[U_j]$. The pseudocode of this process is shown in Algorithm 2.

Once the MIS'es have been found, we start constructing the broadcast tree T_B rooted at node s as shown in Algorithm 3. For nodes in M_i , we choose some of their father nodes in T_{SPT} as their parent nodes in the broadcast tree. The choosing process also proceeds layer by layer. In each layer L_i , we pick one of the father nodes of those nodes in M_i as the parent node if this father node v covers the most unassigned nodes in Q_j , where $j = (i - 1) \pmod{|T|}$. These unassigned

Algorithm 1 MILD algorithm

Input: $G = (V, E)$, s , A , α , r , T .

Output: Gossiping Scheduling $TTS : V \rightarrow 2^N$.

- 1: Apply a proper tessellation and $3\beta^2$ -coloring of hexagons with a radius of $r/2$ in the whole area to color all the nodes. Use $f : V \rightarrow \{1, 2, \dots, 3\beta^2\}$ ($0 \leq j \leq |T| - 1$) to denote this coloring method, where $\beta = \lceil \frac{2}{3}(\alpha + 2) \rceil$.
 - 2: Find a special node s such that the maximum latency of the shortest path tree T_{SPT} rooted at this node is the minimum.
 - 3: Assign $MaxLatency(T_{SPT})$ to D , and divide V into L_0, L_1, \dots, L_D .
 - 4: Apply Algorithm 2 to construct the MIS'es $Q_0, Q_1, \dots, Q_{|T|-1}$ with different active time-slots, and to construct the IS'es M_1, M_2, \dots, M_D layer by layer.
 - 5: Apply Algorithm 3 to construct the broadcast tree T_B rooted at node s and to get the array P to maintain every node's parent node.
 - 6: Apply Algorithm 4 to achieve the scheduling of aggregating the messages to node s and broadcasting the combined message from node s to all the other nodes.
-

Algorithm 2 Construct the MIS'es

- 1: Divide $V \setminus \{s\}$ into subsets $U_0, U_1, \dots, U_{|T|-1}$.
 - 2: **for** $j \leftarrow 0$ to $|T| - 1$ **do**
 - 3: $Q_j \leftarrow \emptyset$
 - 4: **for** $i \leftarrow 1$ to D **do**
 - 5: $j \leftarrow (i - 1) \bmod |T|$, $I' \leftarrow \{i' | (i' - 1) \equiv j \bmod |T|, 1 \leq i' \leq i\}$
 - 6: Construct an IS M_i of $G[L_i]$ such that $Q_j \cup M_i$ is an MIS of $G[\cup_{i' \in I'} L_{i'}]$.
 - 7: $Q_j \leftarrow Q_j \cup M_i$
 - 8: **return** $Q_0, Q_1, \dots, Q_{|T|-1}$ and M_1, M_2, \dots, M_D
-

nodes are set as its children nodes and collected in $C_{1j}(v)$. This process continues until all the nodes in M_i have been assigned parent nodes. Since nodes in $L_i \setminus M_i$ must be adjacent to some nodes in $\bigcup_{i' \in I'} M_{i'}$, where $I' = \{i' | (i' - 1) \equiv j \pmod{|T|}, 1 \leq i' \leq i\}$, we pick some nodes in this set as their parent nodes. The choosing process is similar to the previous one. Note that the node v which covers the most unassigned nodes in $U_j \setminus Q_j$ will be first chosen. These unassigned nodes are set as the children nodes of node v and collected in $C_{2j}(v)$.

Algorithm 3 Construct the broadcast tree

```

1:  $T_B \leftarrow (V_B, E_B)$ ,  $V_B \leftarrow V$ ,  $E_B \leftarrow \emptyset$ 
2: for  $i \leftarrow 1$  to  $D$  do
3:    $j \leftarrow (i - 1) \pmod{|T|}$ 
4:    $X_1 \leftarrow \{u | P(u) = NIL, u \in M_i\}$ ,  $Y_1 \leftarrow \{F(u) | P(u) = NIL, u \in M_i\}$ ,  $Z_1 \leftarrow Q_j$ 
5:    $I' \leftarrow \{i' | (i' - 1) \equiv j \pmod{|T|}, 1 \leq i' \leq i\}$ 
6:    $X_2 \leftarrow \{u | P(u) = NIL, u \in L_i \setminus M_i\}$ ,  $Y_2 \leftarrow \bigcup_{i' \in I'} M_{i'}$ ,  $Z_2 \leftarrow U_j \setminus Q_j$ 
7:   for  $k \leftarrow 1$  to  $2$  do
8:     while  $X_k \neq \emptyset$  do
9:       for each node  $v \in Y_k$  do
10:         $C_{kj}(v) \leftarrow \{u | P(u) = NIL, u \in Z_k \cap N_G(v)\}$ 
11:        Find a node  $v'$  with the maximum  $|C_{kj}(v')|$ .
12:        for each node  $u \in C_{kj}(v')$  do
13:           $P(u) \leftarrow v'$ ,  $E_B \leftarrow E_B \cup \{(v', u)\}$ ,  $X_k \leftarrow X_k \setminus \{u\}$ 
14: return  $T_B$  and  $P$ 

```

Next, we aggregate the messages of all the nodes to the special node s . This process works from the bottom layer to the top layer. In each layer L_i , the scheduling contains two phases: the nodes in $L_i \setminus M_i$ first send their messages to their parent nodes, and then do the nodes in M_i . We collect the children nodes into the set X , and collect the parent nodes into the set Y . The transmissions are scheduled iteratively. In each iteration, every parent node y in Y chooses one of its children nodes x in X , and receives the message from this child node. Note that, to avoid the interference of the transmissions, we schedule these transmissions based on the chromatic numbers of nodes in Q_j , i.e., $f(u)$ of node $u \in Q_j$, where $j = (i - 1) \pmod{|T|}$. If the children

nodes are in $L_i \setminus M_i$, the scheduling is based on the chromatic numbers of their parent nodes, which belong to Q_j according to Algorithm 3. If the children nodes are in M_i , the scheduling is based on the chromatic numbers of these children nodes, which is also in Q_j . After each iteration, the current time-slot t advances to multiple times of $|T|$ when all the transmissions in this iteration can finish. The pseudocode of this process is shown in Algorithm 4 Step I.

Finally, after the messages are aggregated to node s , node s combines all these messages as one message, and broadcast this combined message to all the other nodes as shown in Algorithm 4 Step II. The scheduling works from the top layer to the bottom layer. In each layer L_i , the message is first delivered to nodes in M_i , and then nodes in M_i broadcast the message to its children nodes. Like the scheduling in the data aggregation process, to avoid the interference, we schedule the transmissions from parent nodes to their children nodes based on the chromatic numbers of nodes in Q_j , where $j = (i - 1) \bmod |T|$. We denote by m_{1i} the maximum chromatic number of nodes in M_i , and by m_{2i} the maximum chromatic number of nodes in M_i with children nodes in $U_j \setminus Q_j$. Recall that the children nodes in $U_j \setminus Q_j$ of node v are collected in $C_{2j}(v)$. The message is first delivered to each node u in M_i at $t + (f(u) - 1)|T| + j$. The current time-slot t increases by $m_{1i}|T|$ such that all these transmissions can finish. Each node v in M_i then broadcasts the message to its children nodes in $C_{2j}(v)$ at $t + (f(v) - 1)|T| + j$. Similarly, the current time-slot t increases by $m_{2i}|T|$ such that all these transmissions can finish.

Example 1. We take an example to illustrate the MILD algorithm. The network consists of ten nodes. The network topology of G is shown in Fig. 1(a). The scheduling period T contains ten time-slots from 0 to 9. The active time-slots of ten nodes are $\{7, 6, 7, 4, 6, 7, 9, 5, 8, 2\}$. According to Algorithm 1, we first color all the nodes by a proper tessellation and 27-coloring ($\alpha = 2$) of hexagons as shown in Fig. 1(a), e.g., the color of node 9 is 26. We then find that node 4 is the special node and construct the shortest path tree rooted at node 4 as shown in Fig. 1(b). All the nodes from node 0 to node 9 are divided into several layers $L_8, L_7, L_6, L_{15}, L_0, \dots, L_{13}$ according to the latency of the shortest path from node 4 to every node. Afterward, we construct the MIS'es layer by layer, and construct the broadcast tree T_B as shown in Fig. 1(c).

Next, we aggregate the messages to node 4 from the bottom layer to the top layer according to Algorithm 4 Step I. In this example, the bottom layer L_{15} only contains node 3, and this node will transmit its message to its parent node 2 in T_B at time-slot $t + (f(3) - 1)|T| + A(2) = 0 + (21 - 1) * 10 + 7 = 207$. Then the current time-slot advances to time-slot $\lceil (207 + 1) / 10 \rceil * 10 = 210$,

Algorithm 4 Data aggregation and broadcast the combined message

Step I: Data aggregation

```

1:  $t \leftarrow 0$ 
2: for  $i \leftarrow D$  down to 1 do
3:   if  $L_i \neq \emptyset$  then
4:      $j \leftarrow (i - 1) \bmod |T|$ ,  $X_1 \leftarrow L_i \setminus M_i$ ,  $X_2 \leftarrow M_i$ 
5:     for  $k \leftarrow 1$  to 2 do
6:        $X \leftarrow X_k$ 
7:       while  $X \neq \emptyset$  do
8:          $Y \leftarrow \{P(x) | x \in X\}$ ,  $t' \leftarrow t$ 
9:         for each node  $y \in Y$  do
10:          Find one of its children nodes  $x$  in  $X$ .
11:           $z \leftarrow (k - 1)x + (2 - k)y$ 
12:           $TTS(x) \leftarrow TTS(x) \cup \{t + (f(z) - 1)|T| + A(y)\}$ 
13:          if  $t' < t + (f(z) - 1)|T| + A(y) + 1$  then
14:             $t' \leftarrow t + (f(z) - 1)|T| + A(y) + 1$ 
15:             $X \leftarrow X \setminus \{x\}$ 
16:             $t \leftarrow \lceil t' / |T| \rceil |T|$ 

```

Step II: Broadcast the combined message

```

1: for  $i \leftarrow 1$  to  $D$  do
2:   if  $L_i \neq \emptyset$  then
3:      $j \leftarrow (i - 1) \bmod |T|$ 
4:      $m_{1i} \leftarrow \max\{f(u) | u \in M_i\}$ ,  $m_{2i} \leftarrow \max\{f(v) | v \in M_i \text{ and } |C_{2j}(v)| \neq \emptyset\}$ 
5:     for each node  $u \in M_i$  do
6:        $TTS(P(u)) \leftarrow TTS(P(u)) \cup \{t + (f(u) - 1)|T| + j\}$ 
7:        $t \leftarrow t + m_{1i}|T|$ 
8:     for each node  $v \in M_i$  and  $|C_{2j}(v)| \neq \emptyset$  do
9:        $TTS(v) \leftarrow TTS(v) \cup \{t + (f(v) - 1)|T| + j\}$ 
10:     $t \leftarrow t + m_{2i}|T|$ 
11: return  $TTS$ 

```

and node 9 in the upper layer L_{13} is scheduled to transmit its message to its parent node 2 in T_B at time-slot $t + (f(9) - 1)|T| + A(2) = 210 + (26 - 1) * 10 + 7 = 467$. Node 4 will ultimately receive all the messages at time-slot 1260. Next, node 4 broadcasts the combined message from the top layer to the bottom layer. According to Algorithm 4 Step II, node 4 first sends the message to node 7 in L_6 at time-slot $t + (f(7) - 1)|T| + (6 - 1) \bmod |T| = 1260 + (21 - 1) * 10 + 5 = 1465$, and then the current time-slot advances to time-slot $1260 + 21 * 10 = 1470$. The schedule then proceeds to the next layer containing nodes, and so on. Node 3, the unique node in the bottom layer L_{15} , will receive the message from its parent node 2 at time-slot $2300 + (f(3) - 1)|T| + (15 - 1) \bmod |T| = 2504$. Finally, the total broadcast latency is $2300 + 21 * 10 = 2510$ time-slots.

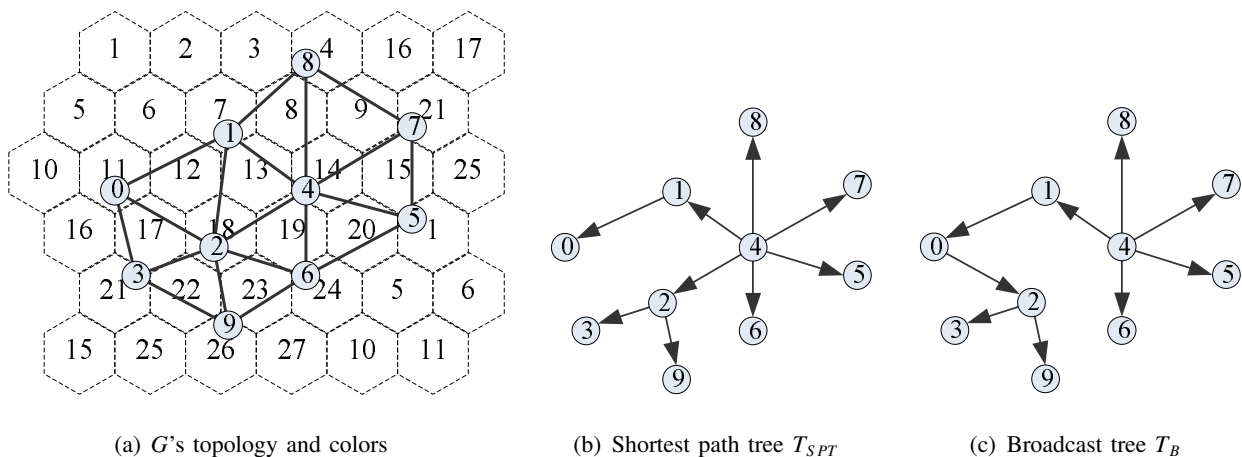


Fig. 1. An example to explain the MILD algorithm

V. PERFORMANCE ANALYSIS

In this section, we show the correctness of the MILD algorithm, and then give the approximation ratio of our algorithm. We also prove that the number of transmissions of our algorithm is within a constant factor of the minimum number of transmissions. Finally, we give the time complexity of our algorithm.

Theorem 1: The MILD algorithm provides a correct and interference-free gossiping scheduling.

Proof: First, we prove the correctness of the data aggregation process. This process works from the bottom layer to the top layer. In each layer L_i , the messages are first gathered from

nodes in $L_i \setminus M_i$ to their parent nodes. According to Algorithm 3, these parent nodes belong to an IS in upper layers or M_i . The messages of nodes in M_i are then aggregated to their parent nodes. These parent nodes are chosen from the father nodes in the shortest path tree, which must be in the upper layers. So node s in the top layer will ultimately receive the messages aggregated from all the other layers.

Next, we prove the correctness of the broadcast process. This process works from the top layer to the bottom layer, and we prove that all the nodes in each layer can be informed by induction. The top layer only contains node s , which already owns the message. So it is true for the top layer. We assume that all the nodes in $L_{i'}$ ($1 \leq i' < i \leq D$) have been informed, and prove that nodes in L_i can be informed. The nodes in L_i are partitioned into two subsets: M_i and $L_i \setminus M_i$. The parent nodes of the nodes in M_i are their father nodes in the upper layers of the shortest path tree. According to the inductive assumption, these parent nodes have been informed, and hence the nodes in M_i can be informed by their parent nodes. If the parent nodes of the nodes in $L_i \setminus M_i$ belong to an IS in upper layers, our claim is true according to the inductive assumption. If these parent nodes belong to M_i , since nodes in M_i will be informed before nodes in $L_i \setminus M_i$ according to the scheduling algorithm, our claim is also true. So the MILD algorithm is true.

Finally, we prove the transmissions scheduled by the MILD algorithm are interference-free. During the data aggregation process, nodes in $L_i \setminus M_i$ are first scheduled to transmit their messages to their parent nodes. We prove these transmissions are interference-free by contradiction. Suppose two nodes u_1 and u_2 in $L_i \setminus M_i$ transmit their messages to their parent nodes $P(u_1)$ and $P(u_2)$ at the same time-slot. If $P(u_1) = P(u_2)$, according to Algorithm 4 Step I, these two transmissions are scheduled separately in different iterations, which contradicts the assumption. Otherwise, without loss of generality, we assume that node $P(u_1)$ fails to receive the message from node u_1 due to the interference of node u_2 . Recall that $d(u_1, u_2)$ denotes the distance between two nodes u_1 and u_2 . It follows that $d(P(u_1), u_2) \leq r_f$ and $d(u_2, P(u_2)) \leq r$. According to the triangle rule, $d(P(u_1), P(u_2)) \leq d(P(u_1), u_2) + d(u_2, P(u_2)) = r_f + r$. According to the construction of the broadcast tree, both two nodes $P(u_1)$ and $P(u_2)$ belong to an IS, and therefore these two nodes are in different hexagons. As what is claimed in Section III-C, the chromatic numbers $f(P(u_1))$ and $f(P(u_2))$ should be different.

If these two transmissions are scheduled in different iterations, the transmitting time-slots should be different, which contradicts the assumption. If these two transmissions are scheduled

in the same iteration, the transmitting time-slots are $t + (f(P(u_1)) - 1)|T| + A(P(u_1))$ and $t + (f(P(u_2)) - 1)|T| + A(P(u_2))$ respectively, which are unequal and contradicts the assumption. Hence we prove these transmissions in this phase are interference-free. The transmissions from nodes in M_i to their parent nodes, and the transmissions in the broadcast process are also based on the chromatic numbers of nodes in an IS. We can use the similar proof to the previous one to prove that these transmissions are interference-free. ■

Lemma 2: The latency of the data aggregation process is at most $3\beta^2(\Delta + 4)|T|D$.

Proof: During the data aggregation process, the messages are aggregated to node s layer by layer. In each layer, nodes in $L_i \setminus M_i$ first aggregate their messages to their parent nodes iteratively. Since each parent node can receive the message of only one of its children nodes during one iteration, the parent node with the most children nodes will always exist in the set Y during all the iterations, and the number of its children nodes in the set X will decrease by one after each iteration. Moreover, this parent node belongs to an IS, which does not include node s , and therefore it should have one parent node in the broadcast tree. So this parent node has at most $\Delta - 1$ children nodes, where Δ is the maximum node degree of the network. The total number of iterations is bounded by $\Delta - 1$. $f(z)$ is no larger than $3\beta^2$, and $A(y)$ is at most $|T| - 1$. The latency of transmissions in one iteration is at most $(3\beta^2 - 1)|T| + (|T| - 1) + 1 = 3\beta^2|T|$. So the latency of data aggregation from nodes in $L_i \setminus M_i$ to their parent nodes is at most $3\beta^2(\Delta - 1)|T|$.

Nodes in M_i then aggregate their messages to their parent nodes. Since one parent node has at most 5 children nodes in M_i , the total number of iterations is bounded by 5. The latency of data aggregation from nodes in M_i to their parent nodes is at most $5 * 3\beta^2|T|$. Combine two kinds of latency as $3\beta^2(\Delta - 1)|T| + 5 * 3\beta^2|T| = 3\beta^2(\Delta + 4)|T|$, which is the latency of data aggregation in each layer. Since there are at most $D + 1$ layers, the latency of the entire data aggregation process is at most $3\beta^2(\Delta + 4)|T|D$. ■

Theorem 2: The approximation ratio of the MILD algorithm is at most $3\beta^2(\Delta + 6)|T|$.

Proof: We first claim that D is a trivial bound for the IAGS-UDC problem. The MILD algorithm contains two processes. The first process is the data aggregation process, and we have prove that the latency of this process is at most $3\beta^2(\Delta + 4)|T|D$ in Lemma 2. The second process is the broadcast process, which works layer by layer. In each layer, the schedule contains two phases. In the first phase, the message is broadcasted to nodes in M_i . Since the maximum chromatic number m_{1_i} is at most $3\beta^2$, the latency of this phase is at most $3\beta^2|T|$. In the second

phase, the message is broadcasted from nodes in M_i to their children nodes. Similar to previous phase, the latency of this phase is at most $3\beta^2|T|$ because m_{2i} is bounded by $3\beta^2$. Combine these two kinds of latency as $3\beta^2|T| + 3\beta^2|T| = 6\beta^2|T|$. In the worst case, there are $D + 1$ layers, so the latency of the broadcast process is at most $6\beta^2|T|D$. Combine the latency of two processes as $3\beta^2(\Delta + 4)|T|D + 6\beta^2|T|D = 3\beta^2(\Delta + 6)|T|D$. ■

Theorem 3: The number of transmissions scheduled by the MILD algorithm is at most 3 times as large as the minimum number of transmissions.

Proof: Since each node needs to transmit at least once to broadcast its message to others, the minimum number of transmissions is n . During the data aggregation process, every node combines its received messages as one message, and is only scheduled once to transmit the combined message to its parent node. In this process, only node s does not transmit, so the total number of transmissions is $n - 1$. During the broadcast process, in each layer L_i , the message is first broadcasted to each node u in M_i . Only one transmission is required to cover node u , so the number of transmissions in this phase is bounded by $|M_i|$. The message is then broadcasted from nodes in M_i to their children nodes with the same active time-slots to them. Note that, each node in M_i transmits only once to cover its such children nodes, and therefore the number of transmissions in this phase is bounded by $|M_i|$. Combine these two numbers of transmissions as $|M_i| + |M_i| = 2|M_i|$. The total number of transmissions during the broadcast process is bounded by $\sum_{1 \leq i \leq D} 2|M_i|$, which is at most $2(n - 1)$. So the number of transmissions scheduled by the MILD algorithm is at most $n - 1 + 2(n - 1) = 3(n - 1)$, which is at most 3 times as large as the minimum number of transmissions n . ■

Theorem 4: The time complexity of the MILD algorithm is $O(n^2|T|^2 + n^3)$.

Proof: The first step in the MILD algorithm is to apply a proper tessellation and $3\beta^2$ -coloring of hexgons to color the nodes. It takes $O(\text{Area}/r^2)$ time to tessellate and color the hexagons, where Area denotes the area size of the whole area, and takes $O(n)$ time to color all the nodes. We can regard $O(\text{Area}/r^2)$ as $O(1)$ if n is large. The next step is to find the special node s . It takes $O(n^2)$ time to construct the shortest path tree rooted at one node, and hence it takes $O(n^3)$ time to construct the shortest path tree rooted at all the nodes and to find node s . It takes $O(n)$ time to divide all the nodes into different layers. In the worst case, the nodes are connected one by one, and the latency of each edge is $|T|$, and the maximum latency of the shortest path tree rooted at the beginning node is $(n - 1)|T|$, so D is no larger than $(n - 1)|T|$. Hence, the running

time of constructing the MIS'es shown in Algorithm 2 is $O(n^2|T|^2)$. Moreover, we can get that the latency of constructing the broadcast tree is $O(n^2|T|^2 + n^3)$. The running time of the data aggregation process and the broadcast process is $O(n^3)$ and $O(n^2)$. Combine all the latency as $O(n^2|T|^2 + n^3)$, which is the time complexity of the MILD algorithm. ■

VI. CONCLUSION

In this paper, we investigate the IAGS-UDC problem. We prove this problem is NP-hard, and propose an approximation algorithm MILD. This algorithm provides a correct and interference-free gossiping scheduling, and achieves a ratio of $3\beta^2(\Delta + 6)|T|$. The number of transmissions scheduled by this algorithm is at most 3 times as large as the minimum number of transmissions. We also show that this algorithm is a polynomial time algorithm.

VII. ACKNOWLEDGEMENT

This work is supported in part by PolyU 5236/06E, PolyU 5232/07E, PolyU 5243/08E, PolyU 5102/08E, PolyU ICRG grant 1-ZV5N, National Basic Research Program of China grant 2006CB30300, and Hunan Provincial NSFC grant No.09ZZ4034.

REFERENCES

- [1] Chlamtac, I., Kutten, S.: On broadcasting in radio networks - problem analysis and protocol design. *IEEE Transactions on Communications* **33** (1985) 1240–1246
- [2] Gandhi, R., Parthasarathy, S., Mishra, A.: Minimizing broadcast latency and redundancy in ad hoc networks. In: *Proc. of ACM MobiHoc*. (2003)
- [3] Chrobak, M., Gasieniec, L., Rytter, W.: Fast broadcasting and gossiping in radio networks. *Journal of Algorithms* **43** (2002) 177–189
- [4] Chrobak, M., Gasieniec, L., Rytter, W.: A randomized algorithm for gossiping in radio networks. In: *Proc. of COCOON*. (2001) 483–492
- [5] Gasieniec, L., Lingas, A.: On adaptive deterministic gossiping in ad hoc radio networks. *Information Processing Letters* **83** (2002) 89–93
- [6] Huang, S.C.H., Du, H., Park, E.K.: Minimum-latency gossiping in multi-hop wireless networks. In: *Proc. of ACM MobiHoc*. (2008)
- [7] Gandhi, R., Y.-A. Kim, S.L., Ryu, J., Wan, P.J.: Approximation algorithms for data broadcast in wireless networks. In: *Proc. of IEEE INFOCOM*. (2009)
- [8] Ni, S.Y., Tseng, Y.C., Chen, Y.S., Sheu, J.P.: The broadcast storm problem in a mobile ad hoc network. In: *Proc. of ACM MobiCom*. (1999)

- [9] Wan, P.J., Wang, Z., Wan, Z., Huang, S.C.H., Liu, H.: Minimum-latency schedulings for group communications in multi-channel multihop wireless networks. In: Proc. of WASA. (2009)
- [10] Wang, F., Liu, J.: Duty-cycle-aware broadcast in wireless sensor networks. In: Proc. of IEEE INFOCOM. (2009)
- [11] Guo, S., Gu, Y., Jiang, B., He, T.: Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. In: Proc. of ACM MobiCom. (2009)
- [12] Hong, J., Cao, J., Li, W., Lu, S., Chen, D.: Sleeping schedule-aware minimum latency broadcast in wireless ad hoc networks. In: Proc. of IEEE ICC. (2009)
- [13] Jiao, X., Lou, W., Ma, J., Cao, J., Wang, X., Zhou, X.: Duty-cycle-aware minimum latency broadcast scheduling in multi-hop wireless networks. In: Proc. of IEEE ICDCS. (2010)
- [14] Huang, S.C.H., Wan, P.J., Deng, J., Han, Y.S.: Broadcast scheduling in interference environment. *IEEE Transactions on Mobile Computing* **7** (2008) 1338–1348