2010 International Conference on Distributed Computing Systems

Duty-Cycle-Aware Minimum Latency Broadcast Scheduling in Multi-hop Wireless Networks

Xianlong Jiao*†, Wei Lou†, Junchao Ma†, Jiannong Cao†, Xiaodong Wang*, and 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: {csxljiao, csweilou, csjma, csjcao}@comp.polyu.edu.hk, {xdwang, xmzhou}@nudt.edu.cn

Abstract

Broadcast is an essential and widely-used operation in multi-hop wireless networks. Minimum latency broadcast scheduling (MLBS) aims to provide a collision-free scheduling for broadcast with the minimum latency. Previous work on MLBS mostly assumes that nodes are always active, and thus is not suitable for duty-cycle-aware scenarios. In this paper, we investigate the duty-cycle-aware minimum latency broadcast scheduling (DCA-MLBS) problem in multi-hop wireless networks. We prove both the one-to-all and the all-to-all DCA-MLBS problems to be NP-hard. We propose a novel approximation algorithm called OTAB for the one-to-all DCA-MLBS problem, and two approximation algorithms called UTB and UNB for the all-to-all DCA-MLBS problem under the unit-size and the unbounded-size message models respectively. The OTAB algorithm achieves a constant approximation ratio of $17|T|$*, where* $|T|$ *denotes the number of time-slots in a scheduling period. The UTB and UNB algorithms achieve the approximation ratios of* $17|T|+20$ *and* $(\Delta+22)|T|$ *respectively, where* Δ *denotes the maximum node degree of the network. Extensive simulations are conducted to evaluate the performance of our algorithms.*

1. Introduction

Multi-hop wireless networks consist of nodes with a limited transmission range. If a node wants to communicate with distant nodes, it requires some intermediate nodes to relay its message. Broadcast is one of the most fundamental communications, and it is widely used in multi-hop wireless networks for routing discovery, data collection, and code update, etc. The two most commonly used broadcast tasks are the one-to-all broadcast and the all-to-all broadcast (also called gossiping in [1]). The one-to-all broadcast aims to disseminate a message from one special node to all the other nodes, while the all-to-all broadcast aims to distribute the messages from all nodes to all the other nodes.

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

In wireless networks, while the packets transmitted by a node can be received by all the nodes within its communication range, two parallel transmissions to one common node can cause signal collision, and the common node will receive neither of the two messages. The objective of minimum latency broadcast scheduling (MLBS) is to minimize the broadcast latency while ensuring the transmissions are collision-free. The MLBS problem in multi-hop wireless networks has been proved to be NP-hard in [2], and several approximation algorithms [2]–[7], which follow the assumption that all the nodes are always active, have been proposed for this problem. However, it is well known that nodes in multi-hop wireless networks often switch between the active state and the sleep state to save the energy. Therefore, most of the previously proposed algorithms are not suitable in these scenarios.

In this paper, we investigate the Duty-Cycle-Aware MLBS (DCA-MLBS) problem. We first present one novel scheduling algorithm OTAB for the one-to-all DCA-MLBS problem. This algorithm provides a scheduling with a constant approximation ratio of $17|T|$, which greatly improves the previously known guarantee of $24|T|+1$ in [8], where $|T|$ denotes the number of time-slots in a scheduling period. We then propose two algorithms called UTB and UNB for the all-to-all DCA-MLBS problem under the *unit-size message model* and the *unbounded-size message model* respectively. Under the *unit-size message model*, the messages may be sent only one by one without combination. Under the *unbounded-size message model*, the messages may be combined as one message. To the best of our knowledge, this is the first work to investigate the all-to-all DCA-MLBS problem. These two algorithms have approximation ratios of $17|T|+20$ and $(\Delta+22)|T|$ respectively, where Δ denotes the maximum node degree of the network. We also prove the correctness of our algorithms. Furthermore, we show the efficiency of our algorithms by conducting extensive simulations under different network configurations.

2. Related Work

Since broadcast plays a very important role in multi-hop wireless networks, a lot of studies have been done on this

1063-6927/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDCS.2010.20

topic [9], [10]. The simplest implementation of broadcast is flooding, which may cause a large number of contention and collision [11]. Therefore, lots of efforts have been made to improve the efficiency of the broadcast [11], [12]. The multi-hop wireless network is often modeled as a unit disk graph (UDG) when the nodes have the same transmission radius. The MLBS problem has been proven to be NP-hard in both the general graphs [9] and the unit disk graphs [2].

Many algorithms have been presented for the one-to-all MLBS problem [2]–[6]. Gandhi et al. [2] presented the first collision-free broadcast scheduling algorithm for multi-hop wireless networks with a constant approximation ratio of more than 400. Huang et al. in [3] further improved this approximation ratio to 16. In [4], the authors presented an approximation algorithm with a smaller constant approximation ratio of 12. For multi-hop wireless networks where the interference range is α times as large as the transmission range, Chen et al. [5] gave a collision-free and interference-free MLBS algorithm with an approximation ratio of $O(\alpha^2)$. The ratio turns out to be 26 if α equals to 2. Mahjourian et al. [6] further considered the carrier sensing range when designing conflict-free broadcast scheduling algorithms.

Much work has also been done for the all-to-all MLBS problem [1], [2], [4], [7], [13]. Recently, Gandhi et al. [2] investigated the all-to-all MLBS problem under the unit-size message model and the UDG model, and proposed an approximation algorithm of a constant ratio. In [7], Huang et al. showed that this ratio is more than 1000, and give a 27-approximation algorithm. This ratio was further improved to 20 by Gandhi et al. in [4]. The all-to-all MLBS problem under the unbounded-size message model for radio networks was studied in [1], [13]. Gasieniec et al. [1] gave two gossiping algorithms which work in $O(n\sqrt{d}\log^2 n)$ and $O(d\Delta^{3/2}\log^3 n)$ time respectively, where d is the diameter of the network, and n denotes the network size. Gasieniec et al. [13] proposed an $O(d)$-time gossiping algorithm for known radio networks with a certain maximum node degree.

None of the work mentioned above, however, takes the active/sleep cycles into consideration. The duty-cycle-aware broadcast problems have been extensively studied in [14]–[16]. But to the best of our knowledge, the only work so far to study the DCA-MLBS problem has been done by Hong et al. [8]. Their algorithm, called ELAC, is proposed for the one-to-all DCA-MLBS problem, and achieves a ratio of $24|T| + 1$. In this paper, we further investigate the one-to-all DCA-MLBS problem and the all-to-all DCA-MLBS problem in the multi-hop wireless networks.

3. Preliminaries

3.1. Network Model

We model the 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 if their Euclidean distance is no larger than the transmission radius. Every node cannot send and receive the messages 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 call node s_c the *graph center* of the network if the hop distance of the path from node s_c to the farthest node in the network is the minimum. This hop distance denoted by R is called the *graph-theoretic radius* of the network, which can be achieved by using the Floyd-Warshall algorithm [17].

We assume that nodes determine the active/sleep time incoordinately in advance. The duty-cycle is defined as the ratio between the active time and 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 unchanged $|T|$ unit time-slots. Every node v chooses one active time-slot $A(v)$ in the scheduling period randomly and independently. A node can transmit the message at any time-slot, but is only allowed to receive the message at its active time-slot. Moreover, we assume that a node can correctly receive the message when no collision occurs at this node.

3.2. Problem Formulation

This paper studies the one-to-all broadcast problem and the all-to-all broadcast problem in duty-cycle-aware scenarios. In the one-to-all broadcast problem, one distinguished node disseminates its message to all the other nodes. The one-to-all broadcast completes when every node receives the message. In the all-to-all broadcast problem, every node has a message to send to all the other nodes. The broadcast task completes when every node receives the messages from all the other nodes. We model the broadcast scheduling as assigning the transmitting time-slots for every node, i.e., assigning a function $TTS : V \to 2^N$.

Unlike the general scenarios, the nodes in the duty-cycle-aware scenarios need to transmit more than one time to cover all their neighbors with different active time-slots. The objective of broadcast scheduling is to minimize the largest transmitting time-slots. Furthermore, the duty-cycle-aware broadcast scheduling requires taking the following constraints into account. First, only when the nodes wake up can the other nodes transmit messages to these nodes. Furthermore, the signal collision only occurs at the nodes which are active at the current time-slot.

If we choose node s as the source node and this node starts the broadcast operation at time-slot 0, for every edge $(u, v) \in E$, the latency $Lat(u, v)$ of this edge is determined as follows:

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

The shortest path tree rooted at node s can be achieved by applying Dijkstra's algorithm with this latency. The

broadcast tree is constructed based on the shortest path tree, and the broadcast 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.

Since the lower bounds of broadcast scheduling under two message models are different, the corresponding algorithms should be designed discriminatively. We have the following lemma about the hardness of the DCA-MLBS problem.

Lemma 1. *Both the one-to-all and the all-to-all DCA-MLBS problems are NP-hard.*

Proof: The NP-hardness of the one-to-all DCA-MLBS problem has been proved in [8]. We only need to prove that the all-to-all DCA-MLBS problem is NP-hard. Since the conventional all-to-all MLBS problem is NP-hard, we can use the similar proof used in [8] to prove the NP-hardness of the all-to-all DCA-MLBS problem. That is, if we set $T = \{0\}$, then all the nodes are always active, and the all-to-all DCA-MLBS problem reduces to the conventional all-to-all MLBS problem. □

3.3. Graph-Theoretic Definitions

We denote by $G[U]$ the subgraph of G induced by a subset U of V. We add edges between two nodes in $G[U]$ if their hop distance is 2, and denote by $G^2[U]$ the resulting graph. 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. U_1 is a *cover* of U_2, if, for any node in U_2, there is one node in U_1 which is adjacent to this node. A *minimal cover* of U is a cover of U in which the removal of any single node destroys the covering property.

A proper coloring of G is to assign natural numbers representing the colors to all the nodes, while ensuring adjacent nodes achieve varied numbers. The minimum node degree and the maximum node degree of G are respectively denoted by $\delta(G)$ and $\Delta(G)$. We denote by $\delta^*(G) = max_{U \subseteq V} \delta(G[U])$ the inductivity of G. According to [18], at most $1 + \delta^*(G)$ colors suffice to color the nodes in G by a proper smallest-degree-last node coloring. Since the nodes in an IS are not adjacent, one node can have at most five neighboring nodes in an IS. Furthermore, for any IS U of G, the inductivity of $G^2[U]$ is no larger than eleven [3].

4. Duty-Cycle-Aware Minimum Latency Broadcast Scheduling

We have shown that both the one-to-all and the all-to-all DCA-MLBS problems are NP-hard. In this section, we first present our approximation algorithm OTAB to solve the one-to-all DCA-MLBS problem. We then propose two approximation algorithms UTB and UNB for the all-to-all DCA-MLBS problem under two different message models.

4.1. Approximation Algorithm for the One-To-All DCA-MLBS Problem

We detail the approximation algorithm OTAB for the one-to-all DCA-MLBS problem in this subsection. The pseudocode of this algorithm is shown in Algorithm 1.

Algorithm 1 OTAB Algorithm

Input: $G = (V, E)$, s, A.
Output: Broadcast Scheduling $TTS : V \rightarrow 2^N$.

1: Apply Dijkstra's algorithm on the graph G to construct the shortest path tree T_{SPT} rooted at the source node s.
2: Assign $MaxLatency(T_{SPT})$ to D, and divide V into L_0, L_1, ..., L_D.
3: Invoke Algorithm 2 to construct the MIS'es and the broadcast tree, and color the parent nodes with two coloring methods f_{1j} and f_{2j}.
4: $t \leftarrow 0$
5: **for** $i \leftarrow 1$ to D **do**
6: **if** $L_i \neq \emptyset$ **then**
7: $j \leftarrow (i-1) \mod |T|$
8: **for** each node $u \in M_i$ **do**
9: $TTS(P(u)) \leftarrow TTS(P(u)) \bigcup \{t + f_{1j}(P(u))|T| + j\}$
10: **for** each node $w \in L_i \backslash M_i$ **do**
11: $TTS(P(w)) \leftarrow TTS(P(w)) \bigcup \{t + (f_{2j}(P(w)) + m_{1j})|T| + j\}$
12: $t \leftarrow t + (m_{1j} + m_{2j})|T|$

This algorithm starts with constructing the shortest path tree T_{SPT} rooted at the source node s. The construction of T_{SPT} can be implemented by assigning every edge the latency defined in Eq. 1 and then applying Dijkstra's algorithm on the graph G. We layer every node according to the latency of the path from node s to this node in the T_{SPT}. Then V can be divided into different layers from L_0 to L_D, where D is the maximum latency of the paths in the T_{SPT}.

Next, we construct the MIS'es layer by layer. The nodes in $V \backslash \{s\}$ are partitioned into different subsets $U_0, U_1, U_2, ..., U_{|T|-1}$ according to their active time-slots. Recall that every node v in $V \backslash \{s\}$ 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 $k = 0, 1, 2, ...$ 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 is $\{i | (i-1) \equiv j \mod |T|, 1 \leq i \leq D\}$. At each layer L_i, we find the independent set M_i of $G[L_i]$ such that $Q_j \bigcup M_i$ is an MIS of $G[\bigcup_{i' \in I'} L_{i'}]$, where I' is $\{i' | (i'-1) \equiv j \mod |T|, 1 \leq i' \leq i\}$, and j is $(i-1) \mod |T|$. Finally, we can find the MIS Q_j of $G[U_j]$. The pseudocode of this process is shown in Algorithm 2 Step I.

Algorithm 2 Construct the MIS'es and the broadcast tree, and color the parent nodes

Step I: Construct the MIS'es

$V\backslash\{s\}$ is partitioned into subsets U_0, U_1, ..., $U_{|T|-1}$.
for $j \leftarrow 0$ to $|T|-1$ **do**
  $Q_j \leftarrow \emptyset$
for $i \leftarrow 1$ to D **do**
  $j \leftarrow (i-1) \mod |T|$
  $I' \leftarrow \{i' | (i'-1) \equiv j \mod |T|, 1 \leq i' \leq i\}$
  Construct an IS M_i of $G[L_i]$ such that $Q_j \bigcup M_i$ is an MIS of $G[\bigcup_{i' \in I'} L_{i'}]$.
  $Q_j \leftarrow Q_j \bigcup M_i$

Step II: Construct the broadcast tree

$T_B \leftarrow (V_B, E_B)$, $V_B \leftarrow V$, $E_B \leftarrow \emptyset$
for $i \leftarrow 1$ to D **do**
  $j \leftarrow (i-1) \mod |T|$
  $X_1 \leftarrow \{u | P(u) = NIL, u \in M_i\}$, $Z_1 \leftarrow Q_j$
  $Y_1 \leftarrow \{F(u) | P(u) = NIL, u \in M_i\}$
  $X_2 \leftarrow \{u | P(u) = NIL, u \in L_i \backslash M_i\}$, $Z_2 \leftarrow U_j \backslash Q_j$
  $I' \leftarrow \{i' | (i'-1) \equiv j \mod |T|, 1 \leq i' \leq i\}$
  $Y_2 \leftarrow \bigcup_{i' \in I'} M_{i'}$
  for $k \leftarrow 1$ to 2 **do**
    $W_{kj} \leftarrow \emptyset$
    while $X_k \neq \emptyset$ **do**
      for each node $v \in Y_k$ **do**
        $C_{kj}(v) \leftarrow \{u | P(u) = NIL, u \in Z_k \bigcap N_G(v)\}$
      Find a node v' with maximum $|C_{kj}(v')|$.
      $W_{kj} \leftarrow W_{kj} \bigcup \{v'\}$
      for each node $u \in C_{kj}(v')$ **do**
        $P(u) \leftarrow v'$
        $E_B \leftarrow E_B \bigcup \{(v', u)\}$
        $X_k \leftarrow X_k \backslash \{u\}$

Step III: Color the parent nodes

for $j \leftarrow 0$ to $|T|-1$ **do**
  Apply a proper D2-coloring method to color the nodes of W_{1j} in $G[W_{1j} \bigcup Q_j]$ by the front-to-back ordering. Use f_{1j} to denote this coloring method.
  Apply a proper D2-coloring method to color the nodes of W_{2j} in $G[W_{2j} \bigcup (U_j \backslash Q_j)]$ by the smallest-degree-last ordering. Use f_{2j} to denote this coloring method.

Once the MIS'es have been found, we start constructing the broadcast tree as shown in Algorithm 2 Step II. For nodes in M_i, we choose some of their father nodes in the T_{SPT} as their parent nodes in the broadcast tree. The choosing process also proceeds layer by layer. At each layer L_i, one of the father nodes of those nodes in M_i is picked as the parent node if this father node covers the maximum number of uncovered nodes in Q_j, where j is $(i-1) \mod |T|$. Its children nodes are set as these uncovered nodes. The edges from the father node to all its children nodes are added into the broadcast tree. This process continues until all the nodes in M_i have been assigned parent nodes. Since nodes in $L_i \backslash M_i$ must be covered by nodes in $\bigcup_{i' \in I'} M_{i'}$, where I' is $\{i' | (i'-1) \equiv j \mod |T|, 1 \leq i' \leq i\}$, we pick some nodes in this set as their parent nodes. The choosing process is similar to the previous one. Note that the node which covers the most uncovered nodes in $U_j \backslash Q_j$ will be first chosen. The parent nodes of nodes in Q_j and $U_j \backslash Q_j$ are collected in W_{1j} and W_{2j} respectively.

After the broadcast tree is constructed, we color the parent nodes. As shown in Algorithm 2 Step III, we first apply a proper D2-coloring method to color the nodes of W_{1j} in $G[W_{1j} \bigcup Q_j]$ by the front-to-back ordering (i.e., from the first node to the last node in W_{1j}), which is denoted as a function $f_{1j} : W_{1j} \rightarrow \{0, 1, ...\}$. Then we apply a proper D2-coloring method to color the nodes of W_{2j} in $G[W_{2j} \bigcup (U_j \backslash Q_j)]$ by the smallest-degree-last ordering, which is also denoted as a function $f_{2j} : W_{2j} \rightarrow \{0, 1, ...\}$. We denote by m_{1j} (and m_{2j}) the total number of colors required to color the nodes in W_{1j} (and W_{2j}).

Finally we schedule the transmissions from the parent nodes to their children nodes as shown in Algorithm 1. This schedule starts at time-slot 0, and works layer by layer. At each layer L_i containing nodes, since the nodes in M_i may be the parent nodes of nodes in $L_i \backslash M_i$, we first schedule the transmissions to the nodes in M_i. Afterward, the transmissions to nodes in $L_i \backslash M_i$ are scheduled. The current time-slot t increases by $(m_{1j} + m_{2j})|T|$ so that all the transmissions to nodes in L_i finish.

4.2. Approximation Algorithm for the All-To-All DCA-MLBS Problem under the Unit-Size Message Model

In this subsection we detail the UTB approximation algorithm for the all-to-all DCA-MLBS problem under the unit-size message model. It contains two phases. In the first phase, all the messages are gathered to one special node. Then these messages are broadcasted from this special node to all the other nodes. The pseudocode of the UTB algorithm is shown in Algorithm 3.

First, the UTB algorithm finds a special node s from V such that the maximum latency of the shortest path tree T_{SPT} rooted at this node is the minimum. It cannot be implemented by directly applying the Floyd-Warshall algorithm [17], because the latencies of the edges defined in Eq. 1 vary for different source nodes. We compute the maximum latency of the shortest path tree rooted at every node, and find the node s. The time complexity of this method is the same as that of the Floyd-Warshall algorithm, which is $O(n^3)$. Then we construct the maximal independent sets and the broadcast tree, and color the parent nodes by exploiting the similar method used in the OTAB

Algorithm 3 UTB Algorithm

Input: $G = (V, E)$, A.
Output: Broadcast Scheduling $TTS : V \rightarrow 2^{\mathcal{N}}$.

Find a special node s such that the maximum latency of the shortest path tree T_{SPT} rooted at this node is the minimum.
Assign $MaxLatency(T_{SPT})$ to D, and divide V into L_0, L_1, ..., L_D.
Invoke Algorithm 2 to construct the MIS'es and the broadcast tree, and color the parent nodes with two coloring methods f_{1j} and f_{2j}.
Node s sends the message $l(1 \leq l \leq n)$ at time-slot t_l, where $t_l = t + (l-1)(m_1^* + m_2^*)|T|$, and t is the time-slot when the data gathering completes.
for $l \leftarrow 1$ to n **do**
  $t' \leftarrow t_l$
  for $i \leftarrow 1$ to D **do**
    if $L_i \neq \emptyset$ **then**
      $j \leftarrow (i-1) \mod |T|$
      for each node $u \in M_i$ **do**
        $TTS(P(u)) \leftarrow TTS(P(u)) \bigcup \{t' + f_{1j}(P(u))|T| + j\}$
      for each node $w \in L_i \backslash M_i$ **do**
        $TTS(P(w)) \leftarrow TTS(P(w)) \bigcup \{t' + (f_{2j}(P(w)) + m_1^*)|T| + j\}$
      $t' \leftarrow t' + (m_1^* + m_2^*)|T|$

Algorithm 4 Data Gathering

Construct the BFS tree T_{BFS} rooted at node s.
Assign $MaxDepth(T_{BFS})$ to β, and divide V into S_0, S_1, ..., S_β.
$t \leftarrow 0$
while node s has not received $n-1$ messages from other nodes **do**
  for $l \leftarrow 0$ to 2 **do**
    for all $k \equiv l \mod 3$ and $0 \leq k \leq \beta$ **do**
      Find a node $u \in S_k$, such that one of its neighboring nodes $v \in S_{k'}(k' > k)$ has a message to transmit or forward.
      $TTS(v) \leftarrow TTS(v) \bigcup \{t + A(u)\}$
    $t \leftarrow t + |T|$
return TTS, t

algorithm. We set $m_1^* = max\{m_{1j} | 0 \leq j \leq |T| - 1\}$ and $m_2^* = max\{m_{2j} | 0 \leq j \leq |T| - 1\}$.

Next, we modify the data gathering algorithm proposed in [7] to make it suitable for duty-cycle-aware scenarios. First, we construct the BFS tree T_{BFS} rooted at node s. The nodes in V are separated into several disjoint subsets $S_0, S_1, S_2, ..., S_\beta$ according to their depths in the T_{BFS}, where β is the maximum depth in the T_{BFS}. Then the transmissions are scheduled in an interleaving manner in the subsets with an interval depth of three. That is, in the first group of subsets S_0, S_3, S_6, ..., we find a node u from each subset such that its neighboring node v in the subset with a greater depth has a message to transmit or forward. Then the transmission from node v to node u is scheduled at time-slot $t + A(u)$, where t is the current time-slot. These transmissions can finish in $|T|$ time slots, so we advance the current time-slot by $|T|$ time-slots. Then the next group of subsets with an interval depth of exactly three (i.e., S_1, S_4, S_7, ...) are handled in a similar way, so do the final group of subsets (S_2, S_5, S_8, ...). This process repeats until node s receives $n-1$ messages from other nodes. The pseudocode of the data gathering process is shown in Algorithm 4.

Once the messages are gathered to node s, we start scheduling the broadcast of these messages. Assume that the messages are numbered by $1, 2, ..., n$, which is the order to be sent by node s. Node s starts to send one message every $(m_1^* + m_2^*)|T|$ time-slots. For example, node s will start to send the message l $(1 \leq l \leq n)$ at time-slot $t + (l-1)(m_1^* + m_2^*)|T|$. The schedule of the transmissions of each message is the similar to that used in the OTAB algorithm. This process is repeated n times for n messages.

4.3. Approximation Algorithm for the All-To-All DCA-MLBS Problem under the Unbounded-Size Message Model

Now we detail the approximation algorithm UNB for the all-to-all DCA-MLBS problem under the unbounded-size message model. Recall that in this scenario the nodes can combine all the messages they have already received as one message, and the combined message can be sent in one time-slot. The UNB algorithm also contains two phases: data gathering and broadcasting. Note that the node responsible for data gathering only needs to send one combined message after receiving all the other messages. The data gathering phase is different from that in the UTB algorithm. It can be thought as a data aggregation phase, which was extensively studied in [19], [20]. However, none of these algorithms take the duty-cycle-aware scenarios into consideration.

We can still use Algorithm 4 to gather the messages, and then use Algorithm 1 to distribute the combined message. However, the approximation ratio may be very large, since the lower bound of the all-to-all broadcast problem in this scenario is smaller than that of the problem under the unit-size message model. Therefore, we present the UNB algorithm, which applies Algorithm 6 to implement data gathering, and then exploits the similar method to Algorithm 1 to broadcast the combined message. The pseudocode of the UNB algorithm is shown in Algorithm 5.

Similar to UTB, the UNB algorithm starts with finding one special node s. The nodes in V are layered according to their latencies in the shortest path tree T_{SPT}. Algorithm 2 is applied to construct the maximal independent sets and

Algorithm 5 UNB Algorithm

Input: $G = (V, E)$, A.
Output: Broadcast Scheduling $TTS : V \to 2^{\mathcal{N}}$.

1: Find a special node s such that the maximum latency of the shortest path tree T_{SPT} rooted at this node is the minimum.
2: Assign $MaxLatency(T_{SPT})$ to D, and divide V into L_0, L_1, ..., L_D.
3: Invoke Algorithm 2 to construct the MIS'es and the broadcast tree, and color the parent nodes with two coloring methods f_{1j} and f_{2j}.
4: Invoke Algorithm 6 to aggregate the messages to node s.
5: Node s sends the combined message at time-slot t when the data aggregation completes.
6: **for** $i \leftarrow 1$ to D **do**
7: **if** $L_i \neq \emptyset$ **then**
8: $j \leftarrow (i-1) \mod |T|$
9: **for** each node $u \in M_i$ **do**
10: $TTS(P(u)) \leftarrow TTS(P(u)) \bigcup \{t + f_{1j}(P(u))|T| + j\}$
11: **for** each node $w \in L_i \backslash M_i$ **do**
12: $TTS(P(w)) \leftarrow TTS(P(w)) \bigcup \{t + (f_{2j}(P(w)) + m_{1j})|T| + j\}$
13: $t \leftarrow t + (m_{1j} + m_{2j})|T|$

the broadcast tree, and to color the parent nodes. Next, we start gathering the messages to node s. Unlike Algorithm 4, the data gathering process works from the bottom layer to the top layer. Before transmitting the messages, every node combines its received messages. At each layer L_i, messages of nodes in $L_i \backslash M_i$ are first transmitted to their parent nodes. To make the computation convenient, the current time advances to multiple times of $|T|$ after these transmissions. Next, we schedule the transmissions from nodes in M_i to their parent nodes. Again, the current time advances to multiple times of $|T|$ after these transmissions.

To schedule all the transmissions, we modify the IMC algorithm proposed in [20] to make it suitable for this scenario. The first set X_k is covered by the second set Y_k, $k = 1$ or 2. First, we find a minimal cover $Z \subseteq Y_k$ of X_k. Note that every node z in Z must have one proprietary neighbor x in X_k, which is only adjacent to this node in the minimal cover Z. This can be easily achieved according to the property of the minimal cover. Clearly, the transmissions between all these pairs of node x and node z are collision-free, so we schedule them in the same scheduling period. For other nodes in X_k except the proprietary neighboring nodes, we find a minimal cover of the set including these nodes again. The transmissions from these nodes to the nodes in the minimal cover are scheduled by using the similar method. This process continues until all the nodes in X_k have been scheduled to transmit messages.

Algorithm 6 Data Aggregation

1: $t \leftarrow 0$
2: Every node combines its received messages before transmitting the messages.
3: **for** $i \leftarrow D$ down to 1 **do**
4: $X_1 \leftarrow L_i \backslash M_i$, $Y_1 \leftarrow \{P(w) | w \in L_i \backslash M_i\}$
5: $X_2 \leftarrow M_i$, $Y_2 \leftarrow \{P(u) | u \in M_i\}$
6: **for** $k \leftarrow 1$ to 2 **do**
7: $c \leftarrow 0$, $X' \leftarrow X_k$, $Y' \leftarrow Y_k$, $t' \leftarrow t$
8: **while** $X' \neq \emptyset$ **do**
9: Find a minimal cover $Z \subseteq Y'$ of X'.
10: **for** each node $z \in Z$ **do**
11: Find a proprietary neighbor $x \in X'$ of Z.
12: $TTS(x) \leftarrow TTS(x) \bigcup \{t + c|T| + A(z)\}$
13: **if** $t' < t + c|T| + A(z) + 1$ **then**
14: $t' \leftarrow t + c|T| + A(z) + 1$
15: $X' \leftarrow X' \backslash \{x\}$
16: $Y' \leftarrow Z$
17: $c \leftarrow c + 1$
18: $t \leftarrow \lceil t'/|T| \rceil |T|$
19: **return** TTS, t

5. Performance Analysis

In this section we analyze the performance of OTAB, UTB and UNB algorithms. First we prove that all these three algorithms provide correct and collision-free broadcast scheduling. We then give the approximation ratios of these algorithms.

5.1. Correctness Analysis

Theorem 1. *The OTAB algorithm provides a correct and collision-free broadcast scheduling.*

Proof: We first prove the correctness of the OTAB algorithm. Since the broadcast scheduling provided by this algorithm works layer by layer, we only need to prove that all the nodes in each layer can be informed. We prove the correctness by induction. In the first layer, $L_0 = \{s\}$. Since node s already owns the message, it is true for the first layer. Assume that all the nodes in the layers before $L_i (1 \leq i \leq D)$ have been informed. The nodes in L_i are partitioned into two subsets: M_i and $L_i \backslash M_i$. The nodes in M_i will be informed by their parent nodes. These parent nodes are chosen from the father nodes in the T_{SPT}. Clearly, these father nodes can cover the nodes in M_i, and are in the layers before L_i. So the nodes in M_i can be informed.

Consider each node w in $L_i \backslash M_i$. Its parent node $P(w)$ is picked from $\bigcup_{i' \in I'} M_{i'}$, where I' is $\{i' | (i' - 1) \equiv j \mod |T|, 1 \leq i' \leq i\}$. According to the construction method of M_i, node $P(w)$ must exist because otherwise node w can be added into M_i. Moreover, we consider two cases. In the first case, node $P(w) \in M_i$. According to Algorithm 1, node

$P(w)$ will be informed before node w. In the second case, node $P(w)$ belongs to the MIS in the previous layer. Based on the inductive assumption, node $P(w)$ can be informed before it transmits the message. Therefore, node w can be certainly informed by its parent node $P(w)$.

Next, we prove that the transmissions from the parent nodes to their children nodes are collision-free. Since the transmissions are scheduled strictly layer by layer, we only consider the transmissions at each layer. At each layer, the messages are first delivered to the nodes in M_i and then to the nodes in $L_i \backslash M_i$. So we distinguish between two cases.

Case 1: The messages are delivered to the nodes in M_i without collision. We prove it by contradiction. Suppose there are two nodes u_1 and u_2 in M_i. Signal collision occurs when their parent nodes $P(u_1)$ and $P(u_2)$ deliver messages to them. That is, these two parent nodes are scheduled to deliver messages to their respective children nodes at the same time-slot, and one child node is covered by both two parent nodes. Without loss of generality, we assume node u_1 is covered by both two nodes $P(u_1)$ and $P(u_2)$. According to the rule of choosing parent nodes and the coloring method shown in Algorithm 2, node $P(u_1)$ must be picked as the parent node earlier than node $P(u_2)$ (otherwise node u_1 will become the child node of node $P(u_2)$), and these two nodes must be colored with different chromatic numbers. Thus, the transmissions from these two nodes to nodes u_1 and u_2 are separated, which contradicts the assumption.

Case 2: The messages are delivered to the nodes in $L_i \backslash M_i$ without collision. Since both the rule of choosing parent nodes and the coloring method in this case are similar to those used in the previous case, we can prove it by using the similar method to that used in the previous case. □

Theorem 2. *The UTB algorithm provides a correct and collision-free broadcast scheduling.*

Proof: After finding the special node s, the messages will be gathered to this node. For each node u in the subset S_k, only if its neighboring node v in the subset with a greater depth has a message to transmit or forward, the transmission from v to u is scheduled. In addition, the transmissions always occur from the nodes far away from node s to the nodes close to node s or node s, so ultimately node s will receive all the messages. Afterward, node s broadcasts the messages to all the other nodes by applying the similar method to OTAB. The correctness of OTAB has been proved, so UTB is correct.

The data gathering algorithm works in an interleaving manner. Only the transmissions to nodes in the subsets whose depths have an exact interval of three are parallel. Moreover, only one transmission is scheduled for each subset. Clearly, these transmissions are collision-free.

Next, we prove that the broadcast of n messages is collision-free. Consider the most complex case where there are $D+1$ layers from L_0 to L_D. As shown in Figure 1, during

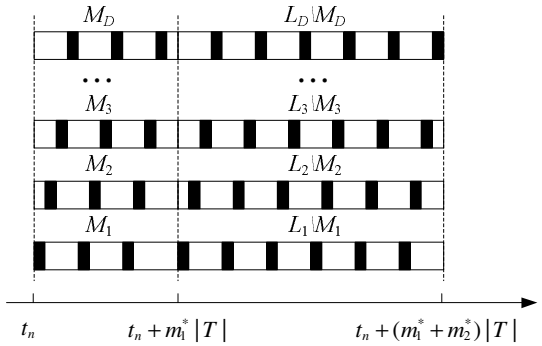


Figure 1. Parallel collision-free transmissions

the first $m_1^*|T|$ time-slots, the messages are delivered to nodes in M_1, M_2, ..., M_D. Consider two kinds of transmissions to nodes in M_i and $M_{i'}$, where $1 \le i \ne i' \le D$. If the nodes in M_i do not share the same active time-slots with the nodes in $M_{i'}$, these two kinds of transmissions are collision-free, because they must be scheduled at different time-slots. Otherwise, we have that $(i-1) \equiv (i'-1) \equiv j \mod |T|$. Therefore, both M_i and $M_{i'}$ belong to Q_j. The parent nodes of nodes in Q_j are colored by applying Algorithm 2. Using the similar proof exploited by previous theorem, we can prove that these two kinds of transmissions are collision-free. The last $m_2^*|T|$ time-slots are preserved for transmissions to nodes in $L_1 \backslash M_1$, $L_2 \backslash M_2$, ..., $L_D \backslash M_D$. Similarly, we can also prove that these transmissions are collision-free. □

Theorem 3. *The UNB algorithm provides a correct and collision-free broadcast scheduling.*

Proof: Consider the data gathering phase. The messages are gathered from the bottom layer to the top layer. At each layer L_i, nodes in $L_i \backslash M_i$ first deliver their messages to their parent nodes. Their parent nodes are in the upper layers or in M_i. Then the messages of nodes in M_i are delivered to their parent nodes. These parent nodes must be in the upper layers. Therefore, node s can certainly receive all the n messages. The broadcast phase is similar to that in OTAB, so UNB is correct.

It is clear that the transmissions in the broadcast phase of UNB are collision-free according to Theorem 1. We only need to prove that the data aggregation algorithm provides a collision-free scheduling. The data aggregation algorithm works layer by layer. In each iteration, the proprietary neighbors are scheduled to deliver messages to their respective parent nodes in the minimal cover. These transmissions are apparently collision-free. Therefore, the data aggregation algorithm provides a collision-free scheduling. □

5.2. Approximation Ratio Analysis

Lemma 2. *To color the parent nodes, the required numbers of colors $m_{1j} \le 5$ and $m_{2j} \le 12$, where $0 \le j \le |T| - 1$.*

Proof: First, consider the set W_{2j}, which is an IS of

G. According to [3], 12 colors are sufficient to color all the nodes in W_{2j} by the smallest-degree-last D2-coloring method, so it follows that $m_{2j} \leq 12$.

We prove $m_{1j} \leq 5$ by contradiction. If $m_{1j} > 5$, we assume that one node v in W_{1j} is colored with 5. This node must have one child node in Q_j. Based on the front-to-back coloring method, there must be five nodes v_1,v_2,v_3,v_4 and v_5 ordered before node v in W_{1j}. These nodes are colored with the numbers from 0 to 4, and each of them has at least one child node in Q_j which is adjacent to node v, because otherwise node v can be colored with one number in $\{0, 1, 2, 3, 4\}$. Since these children nodes are different, node v has more than six neighboring nodes in the IS Q_j which is contradictory. □

Theorem 4. *The approximation ratio of the OTAB algorithm is at most* $17|T|$.

Proof: The lower bound for the one-to-all DCA-MLBS problem has been prove to be D in [8]. We denote by D_{OTAB} the latency of the OTAB algorithm. Since the transmissions are scheduled strictly layer by layer, the broadcast finishes when all the nodes in the last layer receive the messages. In the worst case, there are $D+1$ layers in the T_{SPT}. According to Lemma 2, it follows that

$$D_{OTAB} = \sum_{1 \leq i \leq D}(m_{1j} + m_{2j})|T| \leq (5 + 12)|T|D = 17|T|D$$

□

Lemma 3. *$R + (n - 2)|T| + 1$ is the lower bound for the all-to-all DCA-MLBS problem under the unit-size message model, where R is the network graph-theoretic radius.*

Proof: First, consider the graph center s_c. The maximum depth of the BFS tree T_{BFS} rooted at node s_c is the minimum among the BFS trees rooted at all the nodes in the network. This maximum depth is the network graph-theoretic radius R. It is easy to find that each packet should be transmitted at least R times to reach all the nodes. To complete the all-to-all broadcast task, the total number of transmissions is at least nR. Therefore, we can find a node which transmits at least R times. Furthermore, this node requires to receive $n - 1$ messages from other nodes. Since a node cannot receive and transmit messages at the same time, the minimum latency for this node to complete these operations should be $R + (n - 2)|T| + 1$ time-slots, which provides a lower bound for the all-to-all DCA-MLBS problem under the unit-size message model. □

Lemma 4. *For the all-to-all DCA-MLBS problem under the unit-size message model, the maximum latency D of the T_{SPT} rooted at the special node s is no larger than* $(n - 2)|T|$.

Proof: Before we prove this lemma, we first claim that $D \leq R|T|$. Recall that R is the network graph-theoretic radius, and node s_c is the graph center. We construct the shortest path tree rooted at node s_c based on the latency defined in Eq. 1. Denote by D_{s_c} the maximum latency of this shortest path tree. Consider the T_{BFS} rooted at s_c. The maximum latency of the T_{BFS} is bounded by $R|T|$ if we do not take the collision into account. Since the shortest path tree provides the minimum latency between node s_c and any other nodes, we have that $D_{s_c} \leq R|T|$. Furthermore, according to the definition of D, it follows that $D \leq D_{s_c} \leq R|T|$.

Next, we prove that $R \leq n - 2$ by contradiction. Without loss of generality, we consider $n \geq 3$. Obviously R is smaller than n. If R is greater than $n-2$, R must be $n-1$. In this case, all the nodes must be connected one by one. If we choose an intermediate node, and construct the BFS tree rooted at this node, we can find that the newly constructed BFS tree has a maximum depth smaller than R, which contradicts the assumption. □

Theorem 5. *The approximation ratio of the UTB algorithm is at most* $17|T| + 20$.

Proof: We first analyze the latency of the data gathering algorithm shown in Algorithm 4. This algorithm gathers the messages from the bottom layer to the top layer. It works in an interleaving manner, and node s receives one message every $3|T|$ time-slots. So it will take $3|T|(n - 1)$ time-slots for node s to receive all the $n - 1$ messages.

Next, we analyze the latency of the broadcast process of n messages from node s. Since node s sends one message every $(m_1^* + m_2^*)|T|$ time-slots, the last message will be transmitted at time-slot $3|T|(n - 1) + (m_1^* + m_2^*)|T|(n - 1)$. According to Theorem 4, the last message requires at most $17|T|D$ time-slots to reach all the other nodes. In addition, we consider the case $n \geq 3$, and clearly the network graph-theoretic radius $R \geq 1$. We denote by D_{UTB} the latency of UTB. According to Lemma 2, Lemma 3, and Lemma 4, it follows that

$$\begin{aligned}
D_{UTB} &\leq 3|T|(n-1) + (m_1^* + m_2^*)|T|(n-1) + 17|T|D \\
&\leq 3|T|(n-1) + 17|T|(n-1) + 17|T|D \\
&= 20|T|(n-1) + 17|T|D \\
&\leq 20|T|(n-2) + 20|T| + 17|T|(n-2)|T| \\
&= (17|T| + 20)(n-2)|T| + 20|T| \\
&< (17|T| + 20)(n-2)|T| + (17|T| + 20)(R+1) \\
&= (17|T| + 20)[R + (n-2)|T| + 1]
\end{aligned}$$

□

Lemma 5. *The latency of the data aggregation algorithm shown in Algorithm 6 is at most* $(\Delta + 5)|T|D$ *time-slots.*

Proof: The messages are gathered from the bottom layer to the top layer in the T_{SPT}. We analyze the latency of data aggregation at each layer. Consider two phases. In the first phase, the nodes in $L_i \backslash M_i$ are first scheduled to transmit messages to their parent nodes. According to Algorithm 6, one node will always exist in the minimal cover during the whole iterations, and its degree decreases by at least 1 after each iteration. Therefore, the number of iterations is limited by the maximum node degree Δ. In addition, the active time-

slot is bounded by $|T| - 1$. So the latency of this phase is at most $(\Delta - 1)|T| + (|T| - 1) + 1 = \Delta|T|$ time-slots. Since one node has at most five neighboring nodes in an MIS, the latency of the second phase is at most $5|T|$ time-slots. In the worst case, there are $D+1$ layers, so the latency of the data aggregation algorithm is bounded by $(\Delta + 5)|T|D$. □

Theorem 6. *The approximation ratio of the UNB algorithm is at most* $(\Delta + 22)|T|$.

Proof: First, we claim that D is a trivial lower bound for this problem. We analyze the latencies of two phases in UNB. In the first phase, the messages are gathered from the bottom layer to the top layer in the shortest path tree. According to Lemma 5, this process will take at most $(\Delta + 5)|T|D$ time-slots to finish. In the second phase, node s transmits the combined message to all the other nodes. The latency of this phase is at most $17|T|D$ according to Theorem 4. Combine these two kinds of latencies as $(\Delta + 5)|T|D + 17|T|D = (\Delta + 22)|T|D$. □

6. Performance Evaluation

In this section, we evaluate the performance of our proposed algorithms OTAB, UTB, and UNB by simulations. The performance of the OTAB algorithm is compared with that of the ELAC algorithm [8]. All the nodes are randomly deployed in a rectangle area of $200m \times 200m$. They have the same transmission radius. The metric we test is the broadcast latency. It is the total time-slots required by all the nodes to receive the broadcast messages. We study the effect of different network configurations including the network size, the transmission radius, and the duty cycle on the performance of the algorithms. The network size ranges from 200 to 1000 with an interval of 200. We vary the range of the transmission radius from $20m$ to $60m$. The duty cycle which equals to $1/|T|$ also varies between 0.1 and 0.02. We conduct the experiments with one parameter changed and the other two fixed. Every experiment is run on 20 randomly generated graph topologies. Moreover, we randomly choose 10 source nodes for the one-to-all broadcast scenarios and report the average performance of these experiments.

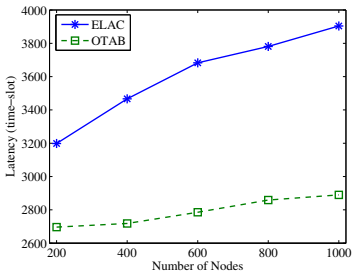
First, we evaluate the impact of the network size on the performance of the algorithms. The transmission radius is fixed to $30m$, and the duty cycle is set as 0.05. The performance curves of four algorithms are shown in Figure 2(a) and Figure 2(b). From Figure 2(a), we can see that the latency of our algorithm OTAB is significantly lower than that of ELAC especially when the network size grows. With the increase of the network size, more nodes must be covered so the performance curves of both the OTAB algorithm and the ELAC algorithm trend up. In Figure 2(b), we can see that the latency of UNB is much lower than that of UTB. This is because more messages have to be sent when the number of nodes increases, and after the data gathering finishes, node s has to send all the messages one by one under the unit-size message model. This latency varies linearly with the increase of the network size.

Next, we study the performance variation of the algorithms with different transmission radius. Figure 2(c) and Figure 2(d) give the results for the experiments with 400 nodes and the duty cycle of 0.05. We can see from Figure 2(c) that our OTAB algorithm performs better than the ELAC algorithm. With the increase of the transmission radius, a node can cover more nodes, so the latencies of both the ELAC algorithm and the OTAB algorithm decrease dramatically. Figure 2(d) shows that the latency of UTB increases slowly, and the performance of UNB does not change notably with the increase of the transmission radius. This is because there are more nodes in a layer of the shortest path tree when the transmission radius increases, which may increase the maximum chromatic number and ultimately increase the latency of broadcasting the messages one by one.

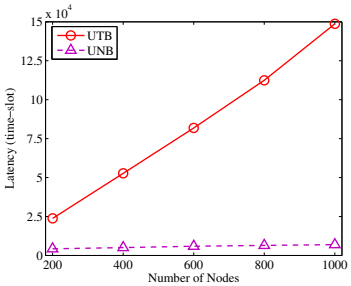
Finally, we evaluate the impact of the duty cycle on the performance of the algorithms. These experiments are run with the network size of 400 and the transmission radius of $20m$. The results are shown in Figure 2(e) and Figure 2(f). We can see from Figure 2(e) that, the OTAB algorithm outperforms the ELAC algorithm, and achieves up to 24.9% improvement with the decrease of the duty cycle. The performance curves of both two algorithms ascend quickly when the duty cycle decreases. This is because as the duty cycle decreases, the scheduling period contains more time-slots, and the maximum latency of the shortest path tree increases. Accordingly, as the number of layers increases, higher latency is required for the nodes in the last layer to receive the broadcast message. Figure 2(f) shows that the latency of UNB is much lower than that of UTB. Both the UTB algorithm and the UNB algorithm show an increasing trend in terms of the latency with the decrease of the duty cycle. This is because there are more layers in the shortest path tree, and higher latency is needed to complete the broadcast layer by layer when the duty cycle decreases.

7. Conclusion

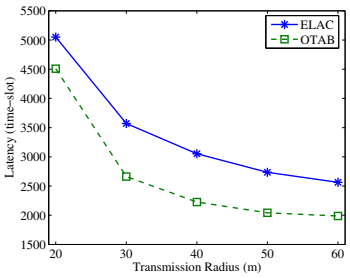
In this paper, we have studied the duty-cycle-aware minimum latency broadcast scheduling problem in multi-hop wireless networks. We show that both the one-to-all and the all-to-all DCA-MLBS problems are NP-hard since they reduce to conventional MLBS problems when the scheduling period contains only one time-slot. All of our three algorithms, OTAB, UTB and UNB, provide correct and collision-free broadcast scheduling. The approximation ratio of the OTAB algorithm is smaller than the previously known guarantee. If the scheduling period T contains a constant number of time-slots, the UTB algorithm achieves a constant approximation ratio of $17|T| + 20$, while the approximation ratio of UNB is at most $(\Delta + 22)|T|$. Extensive results show that the OTAB algorithm outperforms the previous



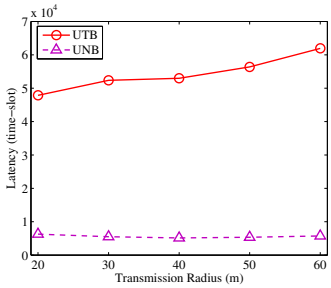
(a) Latency of ELAC and OTAB vs. Number of Nodes



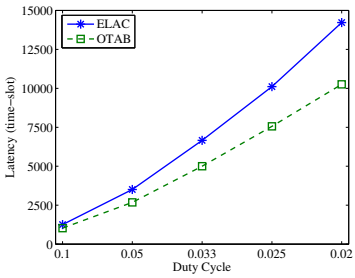
(b) Latency of UTB and UNB vs. Number of Nodes



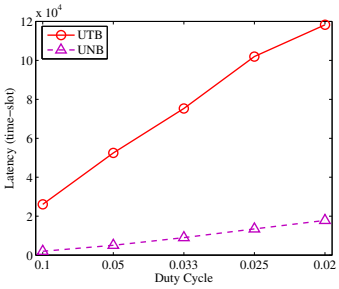
(c) Latency of ELAC and OTAB vs. Transmission Radius



(d) Latency of UTB and UNB vs. Transmission Radius



(e) Latency of ELAC and OTAB vs. Duty Cycle



(f) Latency of UTB and UNB vs. Duty Cycle

Figure 2. Performance Variation Under Different Network Configurations

algorithm under different network configurations and the performance of UTB is more sensitive to the change of network configurations than that of UNB.

References

[1] L. Gasieniec and A. Lingas, "On adaptive deterministic gossiping in ad hoc radio networks," *Information Processing Letters*, vol. 83, no. 2, pp. 89–93, 2002.

[2] R. Gandhi, S. Parthasarathy, and A. Mishra, "Minimizing broadcast latency and redundancy in ad hoc networks," in *Proc. of ACM MobiHoc*, 2003.

[3] S. C.-H. Huang, P.-J. Wan, X. Jia, H. Du, and W. Shang, "Minimum-latency broadcast scheduling in wireless ad hoc networks," in *Proc. of IEEE INFOCOM*, 2007.

[4] R. Gandhi, S. L. Y.-A. Kim, J. Ryu, and P.-J. Wan, "Approximation algorithms for data broadcast in wireless networks," in *Proc. of IEEE INFOCOM*, 2009.

[5] Z. Chen, C. Qiao, J. Xu, and T. Lee, "A constant approximation algorithm for interference aware broadcast in wireless networks," in *Proc. of IEEE INFOCOM*, 2007.

[6] R. Mahjourian, F. Chen, and R. Tiwari, "An approximation algorithm for conflict-aware broadcast scheduling in wireless ad hoc networks," in *Proc. of ACM MobiHoc*, 2008.

[7] S. C.-H. Huang, H. Du, and E.-K. Park, "Minimum-latency gossiping in multi-hop wireless networks," in *Proc. of ACM MobiHoc*, 2008.

[8] J. Hong, J. Cao, W. Li, S. Lu, and D. Chen, "Sleeping schedule-aware minimum latency broadcast in wireless ad hoc networks," in *Proc. of IEEE ICC*, 2009.

[9] I. Chlamtac and S. Kutten, "On broadcasting in radio networks - problem analysis and protocol design," *IEEE Transactions on Communications*, vol. 33, no. 12, pp. 1240–1246, 1985.

[10] I. Chlamtac and O. Weinstein, "The wave expansion approach to broadcasting in multihop radio networks," *IEEE Transactions on Communications*, vol. 39, no. 3, pp. 426–433, 1991.

[11] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc. of ACM MobiCom*, 1999.

[12] W. Lou and J. Wu, "On reducing broadcast redundancy in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 1, no. 2, pp. 111–123, 2002.

[13] L. Gasieniec, I. Potapov, and Q. Xin, "Time efficient gossiping in known radio networks," in *Proc. of the 11th Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2004.

[14] M. Miller, C. Sengul, and I. Gupta, "Exploring the energy-latency tradeoff for broadcasts in energy-saving sensor networks," in *Proc. of IEEE ICDCS*, 2005.

[15] F. Wang and J. Liu, "Duty-cycle-aware broadcast in wireless sensor networks," in *Proc. of IEEE INFOCOM*, 2009.

[16] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," in *Proc. of ACM MobiCom*, 2009.

[17] R. W. Floyd, "Algorithm 97: Shortest path," *Communications of the ACM*, vol. 5, no. 6, p. 345, 1962.

[18] D. W. Matula and L. L. Beck, "Smallest-last ordering and clustering and graph coloring algorithms," *Journal of the Association of Computing Machinery*, vol. 30, no. 3, pp. 417–427, 1983.

[19] B. Yu, J. Li, and Y. Li, "Distributed data aggregation scheduling in wireless sensor networks," in *Proc. of IEEE INFOCOM*, 2009.

[20] P.-J. Wan, S. C.-H. Huang, and L. Wang, "Minimum-latency aggregation scheduling in multihop wireless networks," in *Proc. of ACM MobiHoc*, 2009.