This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2007 proceedings.

Minimum-Latency Broadcast Scheduling in Wireless Ad Hoc Networks

Scott C.-H. Huang*, Peng-Jun Wan*†, Xiaohua Jia*, Hongwei Du* and Weiping Shang‡
*Department of Computer Science, City University of Hong Kong.
Emails: shuang@cityu.edu.hk, {pwan,jia,hongwei}@cs.cityu.edu.hk
†Department of Computer Science, Illinois Institute of Technology.
Email: wan@cs.iit.edu.
‡Institute of Applied Mathematics, Chinese Academy of Science.
Email: shangwp@amss.ac.cn

***Abstract*— A wide range of applications for wireless ad hoc networks are time-critical and impose stringent requirement on the communication latency. This paper studies the problem Minimum-Latency Broadcast Scheduling (MLBS) in wireless ad hoc networks represented by unit-disk graphs. This problem is NP-hard. A trivial lower bound on the minimum broadcast latency is the radius R of the network with respect to the source of the broadcast, which is the maximum distance of all the nodes from the source of the broadcast. The previously best-known approximation algorithm for MLBS produces a broadcast schedule with latency at most $648R$. In this paper, we present three progressively improved approximation algorithms for MLBS. They produce broadcast schedules with latency at most $24R-23$, $16R-15$, and $R+O(\log R)$ respectively.**

I. INTRODUCTION

A wide range of applications for wireless ad hoc networks such as military surveillance, emergency disaster relief and environmental monitoring are time-critical. These applications impose stringent requirement on the communication latency. One major challenge in achieving fast communication is how to handle the intrinsic broadcasting nature of radio communications. From the perspective of communication latency, the broadcasting nature of radio transmission is a double-edged sword. On one hand, it may speed up the communications since it enables a message to reach all neighbors of its transmitter simultaneously in a single transmission. On the other hand, it may also slow down the communications since the transmission by a node may interfere and disable nearby communications. In order to achieve fast communication, one has to magnify the speed-up impact while diluting the slow-down impact of the broadcasting nature. In this paper, we study the problem **Minimum-Latency Broadcast Scheduling** (MLBS) in wireless ad hoc networks in which communication proceeds in synchronous time-slots. In its most general setting, an instance of MLBS consists of an undirected graph $G=(V,E)$ representing the communication topology and a distinguished node $s \in V$ as the source of the broadcast. For any subset U of V, denote by $Inf(U)$ the set of nodes in $V \setminus U$ each of which has exactly one neighbor in U. Then a broadcast schedule of latency ℓ is a sequence $\langle U_1, U_2, \cdots, U_\ell \rangle$ satisfying that (1) $U_1 = \{s\}$; (2) $U_i \subseteq \cup_{j=1}^{i-1} Inf(U_j)$ for each $2 \le i \le \ell$; and (3) $V \setminus \{s\} \subseteq \cup_{j=1}^{\ell} Inf(U_j)$. The problem MLBS seeks a broadcast schedule of the smallest latency.

MLBS in general (undirected) graphs has been extensively studied in the literature. Let n be the number of nodes in the input graph G. Chlamtac and Kutten [2] established the NP-hardness of MLBS in general graphs. Recently, Elkin and Kortsarz proved a logarithmic multiplicative inapproximability and a polylogarithmic additive inapproximability: Unless $NP \subseteq BPTIME\left(n^{O(\log\log n)}\right)$, there exist two universal positive constants c_1 and c_2 such that MLBS admits neither multiplicative $(c_1 \log n)$-approximation [6] nor additive $\left(c_2 \log^2 n\right)$-approximation [7]. A trivial lower bound on the minimum broadcast latency is the *radius* R of G with respect to s, defined as the maximum distance of the nodes in G from s. However, R is a very loose lower bound in general. Indeed, Alon et al. [1] proved the existence of a family of n-node graphs of radius 2, for which any broadcast schedule has latency $\Omega\left(\log^2 n\right)$. Approximation algorithms for MLBS in general graphs developed in the literature can be classified into two categories, multiplicative approximation algorithms [2], [3], [12], and additive approximation algorithms [4], [8], [10], [11], [13]. Table I summarizes the latency of the broadcast schedules constructed by these approximation algorithms.

Approx. Algorithm	Latency
Chlamtac and Kutten [2]	$O(R\Delta)$
Chlamtac and Weinstein [3]	$O\left(R\log^2(n/R)\right)$
Kowalski and Pelc [12]	$O\left(R\log n+\log^2 n\right)$
Gaber and Mansour [10]	$O\left(R+\log^6 n\right)$
Elkin and Kortsarz [8]	$R+O\left(\sqrt{R}\log^2 n\right)$
Gasieniec, Peleg, and Xin [11]	$R+O\left(\log^3 n\right)$
Cicalese, Manne, and Xin [4]	$R+O\left(\log^3 n/\log\log n\right)$
Kowalski and Pelc [13]	$O\left(R+\log^2 n\right)$

TABLE I
APPROXIMATION ALGORITHMS FOR MLPS IN GENERAL GRAPHS.

MLBS in unit-disk graphs (UDGs) was only considered in [5], [9]. For a wireless ad hoc network in which all nodes

0743-166X/07/$25.00 ©2007 IEEE

lie in a plane and have transmission radii equal to one, its communication topology is a UDG $G = (V, E)$ in which there is an edge between two nodes if and only if their Euclidean distance is at most one. In [5] Dessmark and Pelc presented a broadcast schedule of latency at most $2400R$. This should be contrasted with the lower bound $\Omega\left(\log^2 n\right)$ from [1] valid for some graphs with constant R: the graphs constructed in [1] are "pathological," in particular they are not UDG. In [9] Gandhi, Parthasarathy and Mishra claimed the NP-hardness of MLBS in unit-disk graphs and constructed an improved broadcast schedule whose latency can be shown to be at most $648R$. We remark that their algorithm is incorrect but the bug can be fixed.

The main contribution of this paper consists of three progressively improved approximation algorithms for MLBS in UDGs, Basic Broadcast Schedule (**BBS**), Enhanced Broadcast Schedule (**EBS**), and Pipelined Broadcast Schedule (**PBS**). They produce broadcast schedules with latency at most $24R-23$, $16R-15$, and $R+O(\log R)$ respectively. The rest of this paper is organized as follows. In Section II, we introduce some terms, notations and and simple facts. In Section III, Section IV and Section V, we present the first approximation algorithm and the second approximation algorithm respectively. Finally, we conclude this paper in Section VI.

II. Preliminaries

In this section, we introduce some terms, notations and simple facts. Let $G = (V, E)$ be an undirected graph with $|V| = n$, and s be a fixed node in G. The subgraph of G induced by a subset U of V is denoted by $G[U]$. The minimum degree of G is denoted by $\delta(G)$. The inductivity of G is defined by $\delta^*(G) = \max_{U\subseteq V}\delta(G[U])$. For any positive integer k, the k-th power of G, denoted by G^k, is a graph over V in which there is an edge between two nodes u and v if and only if their distance in G is at most k. The *depth* of a node v is the distance between v and s, and the *radius* of G with respect to s, denoted by R, is maximum distance of all the nodes from s. They can be computed by conducting a standard breath-first-search (BFS) on G. For $0 \leq i \leq R$, the layer i of G consists of all nodes of depth i.

Let X and Y be two disjoint subsets of V. A (X, Y)-schedule of latency ℓ is a sequence $\langle U_1, U_2, \cdots, U_\ell\rangle$ satisfying that (1) $U_1 \subseteq X$; (2) $U_i \subseteq X \cup \left(\cup_{j=1}^{i-1} Inf(U_j)\right)$ for each $2 \leq i \leq \ell$; and (3) $Y \subseteq \cup_{j=1}^{\ell} Inf(U_j)$. X is a *cover* of Y if each node in Y is adjacent to some node in X, and a *minimal cover* (MC) of Y if X is a cover of Y but no proper subset of X is a cover of Y. Suppose that X is a cover of Y. Any ordering $x_1, x_2, \cdots, x_m$ of X induces a minimal cover $W \subseteq X$ of Y by the following sequential pruning method: Initially, $W = X$. For each $i = 1$ up to m, if $W \setminus \{x_i\}$ is a cover of Y, remove x_i from W. If X is a cover of $V \diagdown X$, then X is called a *dominating set* of G. If X is a dominating set and $G[X]$ is connected, then then X is called a *connected dominating set* of G.

A subset U of V is a k-*independent set* (k-IS) of G if the pairwise distances of the nodes in U are all greater than k, and a *maximal k-independent set* of G is U is a k-independent set of G but no proper subset of U is a k-independent set of G. Note that a set U is a (maximal) k-IS if and only if it is a (maximal) IS in G^k. Any node ordering $v_1, v_2, \cdots, v_n$ of V induces a maximal k-IS U in the following first-fit manner: Initially, $U = \{v_1\}$. For $i = 2$ up to n, add v_i to U if $dist_G(v_i, U) > k$. The parameter k is often omitted if $k = 1$. Clearly, any maximal IS of G is a dominating set of G, and for any 2-IS U the set $Inf(U)$ consists of all nodes adjacent to U. If G is a UDG, then a set U is an IS of G if and only if any pair of nodes in U are separated by an Euclidean distance greater than one. In addition, each node can be adjacent to at most five nodes in any IS, and any nodes at layer $0 < i < R$ can be adjacent to at most four nodes at the layer $i+1$ in any IS.

A *proper node coloring* of G is an assignment of colors, represented by natural numbers, to the nodes in V such that any pair of adjacent nodes receive different colors. It is equivalent to a partition of V into independent sets. Any node ordering $v_1, v_2, \cdots, v_n$ of V induces a proper node coloring of G in the following first-fit manner: For $i = 1$ to n, assign to v_i the least possible color which is not used by any neighbor v_j with $j < i$. A particular node ordering of interest is the smallest-degree-last ordering. A smallest-degree-last ordering $v_1, v_2, \cdots, v_n$ can be generated by the following simple algorithm: Initially, $U = V$. For $i = n$ down to 1, set v_i to be the node of smallest degree in $G[U]$ and then remove v_i from U. It's well-known that the node coloring of G induced by a smallest-degree-last ordering uses at most $1 + \delta^*(G)$ colors [14].

III. Basic Broadcast Scheduling

In this section, we present a simple algorithm **BBS** for MLBS in UDG which produces a broadcast schedule with latency at most $24R-23$. This algorithm exploits the fact that any independent set U of a UDG G can be partitioned into at most twelve 2-independent sets of G in polynomial-time even if the positions of the nodes are not available. Indeed, we observe that a partition of U into 2-IS's is equivalent to a proper node coloring of $G^2[U]$ with each subset in the partition corresponding to a color class. Thus, we compute the coloring induced by the smallest-degree-last ordering and output the color classes as the partition of U. The number of colors, or equivalently the number of subsets in the out partition, is at most $1 + \delta^*\left(G^2[U]\right)$. The next lemma shows that $\delta^*\left(G^2[U]\right) \leq 11$, and consequently the output partition meets the requirement.

Lemma 1: For any IS U of a UDG G, $\delta^*\left(G^2[U]\right) \leq 11$.

Proof: We first show that $\delta\left(G^2[U]\right) \leq 11$. Let v be the bottom-most node in U. It is sufficient to show that the the degree of v in $G^2[U]$ is at most 11. By the selection of v, all neighbors of v in $G^2[U]$ lie in the top half-annulus centered at v with radii one and two. Consider a half-annulus of radii one and two centered at a point v. Separate this half-annulus into two by an intermediate circle with radius $2\cos(\pi/7) - \sqrt{4\cos^2(\pi/7) - 3}$, then partition the inner half-annulus into 4 equal-sized pieces and the outer half annulus into 7 equal-sized pieces as shown in Figure 1. A straightforward calculation yields that each of the 11 pieces has diameter at most one. So each piece can contain at most one node in U. Hence, the half-annulus of radii one and two centered at v contains at most 11 nodes in U, and thus the degree of v in $G^2[U]$ is at most 11.

Next, we prove $\delta^*\left(G^2[U]\right) \leq 11$. Note that for any subset U' of U, U' is itself an IS of G and the subgraph of $G^2[U]$ induced by U' is $G^2[U']$. Thus, $\delta\left(G^2[U']\right) \leq 11$. This implies that $\delta^*\left(G^2[U]\right) \leq 11$. ■

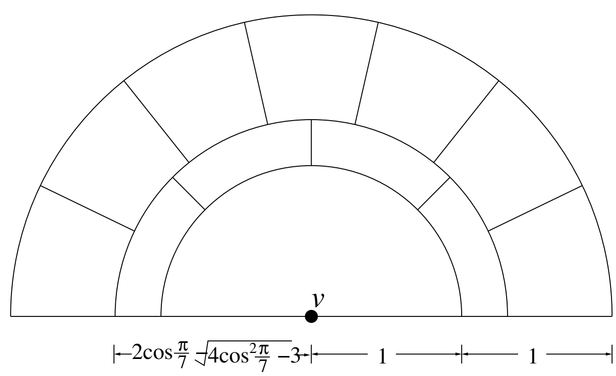

Fig. 1. Partition of the half annulus with radii 1 and 2 into 11 pieces of diameter at most one.

We remark that if the positions of the nodes are available, we can even partition U into at most 12 subsets, each of which consists of the nodes with pairwise distances greater than two and hence is 2-IS. Such partition can be obtained by the tiling approach presented in [16]. Specifically, we tile the plane into regular hexagons of side equal to $1/2$ (see Figure 2(a)). Each hexagon, or cell, is left-closed and right-open, with the top-most point included and the bottom-most point excluded (see Figure 2(b)). Clearly, each cell contains at most one node in U. Cells are further grouped into clusters of size 12 according to the pattern as shown in Figure 2(a). We then label the 12 cells in a cluster with the numbers 1 through 12 in an arbitrary pattern, as long as all clusters adopt the same pattern of labeling of the cells. For each $1 \leq i \leq 12$, let U_i be the set of nodes in U lying the cells with label i. Since the distance between any two points in two different (half-closed and half-open) cells with the same label is greater than two, all nodes

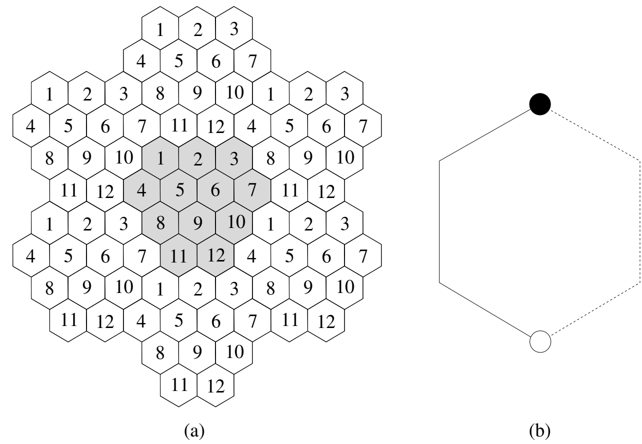

Fig. 2. Tiling of the plane into hexagons with 12 hexagons per cluster.

in U have pairwise distances greater than two. Hence, the sets U_i with $1 \leq i \leq 12$ form a desired partition of U.

Now we are ready to describe the algorithm **BBS**. We first construct a BFS tree T of G rooted at s, and compute the depths of all nodes in T and the radius R of G. Then, we construct the MIS U of G induced by the increasing order of depth. The nodes in U are referred to as *dominators* as U is also a dominating set of G. The parents of the dominators in T are referred to as *connectors*, as they together with the dominators, form a connected dominating set. Only the dominators and connectors are the transmitting nodes. Their transmissions are scheduled layer by layer in the top-down manner. At each layer, transmissions by dominators precede the transmissions by connectors. Specifically, for each $0 \leq i \leq R$, denote by U_i the set of dominators with depth i. Note that $U_0 = \{s\}$ and $U_1 = \varnothing$. For each $2 \leq i \leq R$, compute a partition of U_i into c_i 2-IS's U_{ij} for $1 \leq j \leq c_i$ with $c_i \leq 12$. For each $1 \leq i \leq R-1$ and $1 \leq j \leq c_{i+1}$, denote W_{ij} to be the set of parents of nodes in $U_{i+1,j}$. Then, at layer 0, only the source s transmits as dominator in time-slot 0. At layer 1, no node is a dominator, and connectors transmit in the sequence $\langle W_{1j} : 1 \leq j \leq c_2\rangle$. At each layer i with $2 \leq i \leq R-1$, dominators transmit in the sequence $\langle U_{ij} : 1 \leq j \leq c_i\rangle$ and immediately afterwards, connectors transmit in the sequence $\langle W_{ij} : 1 \leq j \leq c_{i+1}\rangle$. At layer R , no node is a connector, and dominators transmit in the sequence $\langle U_{Rj} : 1 \leq j \leq c_R\rangle$.

The next theorem asserts the correctness of the algorithm **BBS** and establishes an upper bound on the latency of the broadcast schedule produced by **BBS**.

Theorem 2: Algorithm **BBS** is correct and it produces a broadcast schedule with latency at most $24R - 23$.

Proof: By the property of 2-IS, after a dominator transmits, all its neighbors in G are informed. By the selection of dominators, each connector is adjacent to some dominator in the previous or the same layer. Thus, all connectors in a layer must have been informed after the transmissions by dominators in the same layer. By the selection of the connectors and their transmission scheduling, the dominators at a layer must have been informed after all connectors at the previous layer have completed their transmissions. Finally, after the transmissions by all dominators, all other nodes are informed. Therefore, algorithm **BBS** is correct.

A straightforward calculation yields that the latency of the broadcast schedule is $1 + c_2 + \sum_{i=2}^{R-1}(c_i + c_{i+1}) + c_R = 1 + 2\sum_{i=2}^{R} c_i$. Since $c_i \leq 12$ for each $2 \leq i \leq R$, the latency is bounded by $1 + 2 \cdot 12 (R - 1) = 24R - 23$. ∎

Finally, we remark that while each dominator transmits exactly once, a connector may transmits at most four times. Precisely, the number of transmissions by a connector is equal to the number of dominator children in T, which is at most four.

IV. Enhanced Broadcast Scheduling

In this section, we present an algorithm **EBS** for MLBS in UDG which produces a broadcast schedule with latency at most $16R - 15$. The algorithm **EBS** is an enhancement from **BBS**. It differs from the algorithm **BBS** only in how the connectors are selected and scheduled for transmissions. Instead of choosing all parents of the dominators as connectors as in **BBS**, it selects a *minimal* subset of parents of the dominators as connectors. As a result, all the connectors at a layer would take at most 4 time-slots to transmit in **EBS**, a reduction from as many as 12 time-slots taken by the connectors at a layer to transmit in **BBS**. In addition, each of the dominators and connectors transmit exactly once in **EBS**, while a connector may transmits up to 4 times in **BBS**. Thus, **EBS** not only produces shorter broadcast schedule, but also eliminates the transmission redundancy completely.

The selection and transmission scheduling of the connectors in **EBS** are generated by an algorithm called *iterative minimal covering* (**IMC**). The algorithm **IMC** takes as input a graph G and a pair of disjoint vertex subsets (X, Y) satisfying that X is a cover of Y, and outputs a sequence $\langle W_i : 1 \leq i \leq l \rangle$ of disjoint subsets of X satisfying that (1) $W_1 \cup W_2 \cup \cdots \cup W_l$ is a minimal cover of Y, (2) $Y \subseteq Inf(W_1) \cup Inf(W_2) \cup \cdots \cup Inf(W_l)$, and (3) l is no more than the maximum number of nodes in Y adjacent to a node in X. It runs as follows. Initialize $l = 0$, $X_0 = X$, and $Z = Y$. Repeat the following iteration while $Z \neq \varnothing$: Increment l by 1, compute a minimal cover $X_l \subseteq X_{l-1}$ of Z, set $W_l = X_{l-1} \setminus X_l$, and remove $Inf(X_l)$ from Z. When $Z = \varnothing$, set $W_l = X_l$ and output the sequence $\langle W_i : 1 \leq i \leq l \rangle$. The next lemma shows that the output sequence meets the three desired properties.

Lemma 3: The algorithm IMC is correct.

Proof: Clearly, $X_i = W_i \cup W_{i+1} \cup \cdots \cup W_l$ for each $1 \leq i \leq l$. Thus (1) holds since X_1 is a minimal cover of Y. Next, we prove (2) holds. Let $Y_0 = Y$, and for each $1 \leq i \leq l$, let $Y_i = Y \setminus (Inf(X_1) \cup \cdots \cup Inf(X_i))$. Then, Y_i is the set Z at the end of the i-th iteration. Since Y_l is empty, $Y \subseteq Inf(X_1) \cup Inf(X_2) \cup \cdots \cup Inf(X_l)$. Consider an arbitrary node $y \in Y$. Then $y \in Inf(X_i)$ for some $1 \leq i \leq l$. Let x be the unique neighbor of y in X_i, and suppose that $x \in W_j$ for some $i \leq j \leq l$. Then, x is also the unique neighbor of y in W_j. So $y \in Inf(W_j)$, which implies that (2) holds. Finally, we prove (3). Let x be an arbitrary node in X_l. Then, x belongs to each X_i for $1 \leq i \leq l$. Since X_i is a minimal cover of Y_{i-1}, there is a node y_{i-1} in Y_{i-1} satisfying that y_{i-1} is a neighbor of x but is not a neighbor of any other node in X_i. Hence, $y_{i-1} \in Inf(X_i)$. This implies that $y_0, y_1, \cdots, y_{l-1}$ are distinct. Thus, x has at least l neighbors. In other words, l is no more than the neighbors in Y of x. Therefore, (3) holds as well. ∎

Lemma 3 implies that if additionally X is a subset of nodes at the layer i and Y is a subset of independent nodes at the layer $i+1$ for some $0 \leq i \leq R-1$, then the sequence out by the algorithm **IMC** consists of at most four sets.

The algorithm **EBS** construct a BFS tree T of G rooted at s, and compute the depths of all nodes in T and the radius R of G. Then we construct the MIS U of G induced by the increasing order of depth as the set of dominators. For each $0 \leq i \leq R$, denote by U_i the set of dominators with depth i. Note that $U_0 = \{s\}$ and $U_1 = \varnothing$. Compute a partition of U_i into c_i 2-IS's U_{ij} for $1 \leq j \leq c_i$ with $c_i \leq 12$. For each $1 \leq i \leq R-1$, let P_i be the set of parents of the nodes in U_{i+1}. Apply the algorithm **IMC** to G and the pair (P_i, U_{i+1}), and let $\langle W_{ij} : 1 \leq j \leq l_i \rangle$ be the output sequence of subsets of P_i. The nodes in $\cup_{i=1}^{R-1} \cup_{j=1}^{l_i} W_{ij}$ are the selected connectors, as they together with the dominators form a connected dominating set of G. Only the dominators and connectors are the transmitting nodes, and their transmissions are scheduled layer by layer starting from layer 0. For each layer, transmissions by dominators are scheduled before transmission by connectors. Specifically, layer 0 has only the source s as dominator, which transmits in time-slot 0. Layer 1 contains no dominators, and connectors in layer 1 transmit in the sequence $\langle W_{1j} : 1 \leq j \leq l_1 \rangle$. For each layer $2 \leq i \leq R-1$, dominators in layer i transmit in the sequence $\langle U_{ij} : 1 \leq j \leq c_i \rangle$ and immediately afterwards, connectors in layer i transmit in the sequence $\langle W_{ij} : 1 \leq j \leq l_i \rangle$. Layer R contains no connectors, and dominators in layer R scheduled in the sequence $\langle U_{Rj} : 1 \leq j \leq c_R \rangle$.

The next theorem asserts the correctness of the algorithm **EBS** and establishes an upper bound on the latency of the broadcast schedule produced by **EBS**.

Theorem 4: Algorithm **EBS** is correct and it produces a broadcast schedule with latency at most $16R - 15$.

Proof: The correctness of **EBS** follows from the same argument for the correctness of **BBS**. The latency of the broadcast schedule produced by **EBS** is $1 + l_1 + \sum_{i=2}^{R-1} (c_i + l_i) + c_R = 1 + \sum_{i=1}^{R-1} l_i + \sum_{i=2}^{R} c_i$. Since $l_i \leq 4$ for each $1 \leq i \leq R-1$ and $c_i \leq 12$ for each $2 \leq i \leq R$, the latency is bounded by $1 + 12(R-1) + 12(R-1) = 16R - 15$. ■

We conclude this section with an algorithm *Inter-Layer Broadcast Scheduling* (**ILBS**), which outputs a (X, Y)-schedule of latency at most 16 for any pair of vertex subsets X and Y satisfying that X (resp., Y) is a subset of nodes in the layer i (resp., $i+1$) for some $0 \leq i \leq R-1$ and X is a cover of Y. The algorithm selects a maximal independent set U of $G[X \cup Y]$ induced by an ordering in which nodes in X are before the nodes in Y. Then, apply the algorithm IMC to G and the pair $(X, U \cap Y)$ to obtain a sequence $\langle W_i : 1 \leq i \leq l \rangle$ of disjoint subsets of X. Next, compute a partition of U into 2-IS's U_i for $1 \leq i \leq c$ with $c \leq 12$. Then, output transmission schedule is $\langle W_i : 1 \leq i \leq k \rangle, \langle U_i : 1 \leq i \leq c \rangle$. Note that if $i = 1$, then $X = \{s\}$ and the latency is trivially equal to one. If $i > 1$, then $l \leq 4$ and hence the latency is $k + c \leq 4 + 12 = 16$.

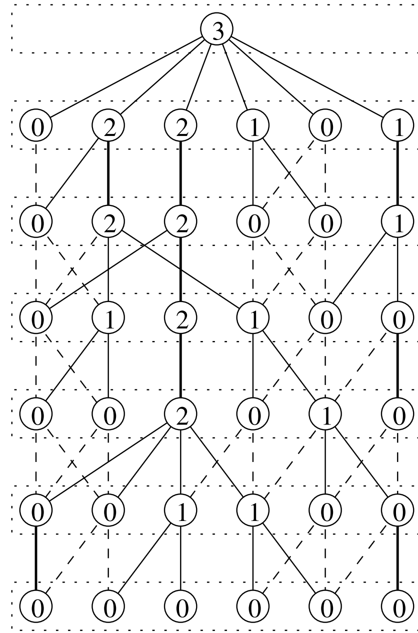

Fig. 3. The ranking of V and the canonical BFS tree consisting of solid edges.

V. Pipelined Broadcast Scheduling

In this section, we present an algorithm **PBS** for MLBS in UDG which produces a broadcast schedule with latency $R + O(\log R)$. Instead of scheduling the transmissions layer-by-layer in the top-down manner, **PBS** pipelines transmissions in more than one layers. This means that a node in a lower layer may receive and/or transmit the messages than a node in an upper layer. Such pipelining relies on a special BFS tree T referred to as *canonical BFS tree* and an associated ranking $rank$ of the nodes constructed layer-by-layer in the bottom-up manner as follows. Initially, T is empty and $rank(v) = 0$ for each node v at the layer R. The ranks and the children of all nodes at each other layer i are computed iteratively: Initialize U to be the set of nodes at layer i, and W to be the set of nodes at layer $i+1$. Repeat the following iteration while W is nonempty. Compute the maximum rank r of the nodes in W, and find a node $v \in U$ which is adjacent to the largest number of nodes in W with rank r. If v is adjacent to only one node in W with rank r, then $rank(v) = r$; otherwise, $rank(v) = r + 1$. Put all neighbors of v in W as the children of v in T. Remove v from U, and remove all neighbors of v from W. When W is empty, then for each node $v \in U$, $rank(v) = 0$. Figure 3 gives an example of the ranking and the canonical BFS tree constructed in this way.

The canonical BFS tree and the ranking have a number of interesting properties. Clearly, each node has rank no more than its parent in T. It's also easy to prove by induction in the bottom-up manner that for each node v, $rank(v) \leq \lfloor \log |T_v| \rfloor$, where T_v is the subtree of T induced by v and all its descendants. In particular, for each node v, $rank(v) \leq \lfloor \log n \rfloor$. Furthermore, if u_1 and u_2 are two nodes at the same layer, v_1 and v_2 are their child respectively at layer $i+1$, and all of them have the same rank, then neither u_1 and v_2 nor u_2 and v_1 are adjacent in G. Indeed, assume by symmetry that u_1 is ranked before u_2. Then, by the time of u_1 is picked for ranking, both v_1 and v_2 remain in W. Since u_1 and v_1 have the same rank, v_1 must be the only neighbor of u_1 in W. In particular, v_2 is not adjacent to u_1. Since u_1 is ranked before u_2, v_2 is also the only neighbor of u_2 in W and hence v_1 is not adjacent to u_2.

Now, we describe a weaker algorithm $\mathcal{A}$ for MLBS in UDG which produces a broadcast schedule with latency $R + 51r = R + O(\log n)$, and will later be used to develop the algorithm **PBS**. For each integer i and j, set $t_{ij} = i + 351(r - j)$. Compute the radius R, a canonical BFS tree T and the associated ranking. Let r be the rank of the source node s. For each $0 \leq i < R$ and $0 \leq j \leq r$, set V_{ij} to be the set of nodes in layer i with rank j, and V'_{ij} to be the set of their children. Let G_{ij} be the subgraph of G induced by $V_{ij} \cup V'_{ij}$. Each subgraph G_{ij} is a basic pipelined scheduling unit. Within G_{ij}, a session S_{ij} sends the message from V_{ij} to V'_{ij} as follows. Denote by W_0 the set of parents of nodes in V'_{ij} with rank j. Apply the algorithm **ILBS** to generate a $(V_{ij}, V'_{ij} \setminus Inf(W_0))$-schedule $\langle W_1, W_2, \cdots, W_l \rangle$. Then, for each $0 \leq k \leq l$, all nodes in W_k transmit in the time-slot $t_{ij} + 3k$.

The next theorem asserts the correctness of the algorithm **EBS** and establishes an upper bound on the latency of the broadcast schedule produced by **EBS**.

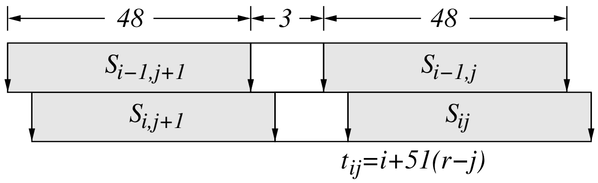

Fig. 4. The transmission scheduling of S_{ij}'s.

Theorem 5: The algorithm $\mathcal{A}$ is correct and it produces a broadcast schedule of latency at most $t_{R,0} = R + 51r$.

Proof: Each S_{ij} starts at the time-slot t_{ij} and ends no later than the time-slot $t_{ij} + 48$ by the property of ILBS (see Figure 4). If $j = 0$, then S_{ij} either has no transmissions or has all the transmissions only on the time-slot t_{ij}. We claim that that each S_{ij} ends before the time-slot $t_{R,0}$. Note that t_{ij} strictly increases with i and decreases with j. If $j > 0$, then S_{ij} ends no later than the time-slot $t_{ij} + 48 = t_{i,j-1} - 3 \leq t_{R-1,0} - 3 < t_{R,0}$. If $j = 0$, then S_{ij} ends no later than the time-slot $t_{ij} \leq t_{R-1,0} < t_{R,0}$. Thus, our claim is true, and consequently the latency is at most $t_{R,0} = R + 51r$.

Now, we show that if $(i,j) \neq (i',j')$, then S_{ij} and $S_{i'j'}$ do not interfere with each other. If $|i - i'| > 2$, the claim holds due to far separation. If $0 < |i - i'| \leq 2$, then the claim holds due to the interleaving of the transmissions. If $i = i'$, then $j \neq j'$. By symmetry, assume that $j < j'$. Then $t_{ij} = t_{i,j+1} + 51 \geq t_{i,j'} + 51 = (t_{i,j'} + 48) + 3$. This implies that the session in $S_{ij'}$ ends at least 3 slots before S_{ij} starts . Next, we prove by induction on that by all nodes in V_{ij} are already informed before the time-slot t_{ij}. This is true if $i = 0$. Assume that $i > 1$ and consider a node $v \in V_{ij}$. Let u be its parent in T and j' be the rank of u. Then, $j' \geq j$. If $j' = j$, then v is informed in $G_{i-1,j}$ in the time-slot $t_{i-1,j} = t_{ij} - 1$ by the property of T. If $j' > j$, then v is informed in $S_{i-1,j'}$, which ends before by S_{ij} starts. Therefore, the algorithm $\mathcal{A}$ is correct. ∎

Finally, we are ready to describe the algorithm **PBS**. We first construct a BFS tree T of G rooted at s, and compute the depths of all nodes in T and the radius R of G. Then we construct the MIS U of G induced by the increasing order of depth as the set of dominators. Compute the shortest-path tree T' from s to all other dominators. In other words, T' is the minimal subtree of T spanning the dominators. Let V' be the set of nodes in T', and G' be the subgraph of G induced by V'. The broadcast schedule consists of two phases. The first phase is a broadcast schedule in G' from s produced by the algorithm $\mathcal{A}$. The second phase schedules the transmissions by the dominators in the following way: Compute a partition of U into 2-IS's U_i for $1 \leq i \leq c$ with $c \leq 12$. Then, the dominators transmit in the sequence $\langle U_1, U_2, \cdots, U_c \rangle$.

Theorem 6: The algorithm **PBS** is correct and it produces a broadcast schedule of latency $R + O(\log R)$.

Proof: The correctness of **PBS** is trivial. Next, we show that the broadcast schedule produced by **PBS** has latency $R + O(\log R)$. Since the second phase of the broadcast schedule takes at most 12 time-slots, it is sufficient to show that the second phase of the broadcast schedule has latency $R + O(\log R)$. Denote by R' the radius of G' with respect to s. Then R' is equal to either R or $R - 1$. By the folklore area argument, $|U| = O\left(R'^2\right)$. Since the shortest-path between s and each dominator contains at most R' nodes other than s, $|V'| \leq |U| \cdot R' + 1 = O\left(R'^3\right)$. By Theorem 5, the broadcast schedule in G' from s produced by $\mathcal{A}$ has latency $R' + O\left(\log O\left(R'^3\right)\right) = R + O(\log R)$. ∎

VI. Discussions

In this paper, we present three progressively improved approximation algorithms **BBS**, **EBS**, and **PBS** for MLBS in UDGs. They produce broadcast schedules with latency at most $24R - 23$, $16R - 15$, and $R + O(\log R)$ respectively. In the broadcast schedules output by **BBS** and **EBS**, the number of transmitting nodes is no more than eight times the size of a minimum connected dominated set [15]. In addition, **EBS** eliminates the transmission redundancy in **BBS**. While the algorithm **PBS** may produce shorter broadcast schedule, it cannot guarantee that the number of transmitting nodes is within a constant factor of the minimum. If we the subgraph of G induced by the dominators and connectors constructed in **EBS** as the graph G' used in **PBS**, then such modified **PBS** outputs a broadcast schedule of latency $2R + O(\log R)$ and ensures that the number of transmitting nodes is within the constant factor of the minimum.

A generalization to the problem MLBS is **Minimum-Latency Multi-Source Multicast Scheduling** which seeks a shortest (X, Y)-schedule for any be two disjoint subsets of X and Y of a UDG G. All the three approximation algorithms can be extended in the straightforward manner for approximating this general problem with similar approximation factors.

Acknowledgments

The authors would like to thank Prof. Frances F. Yao for her insightful comments and suggestions on the early version of this work. Scott C.-H. Huang is supported in part by Research Grants Council of Hong Kong under Project No. CityU 1165/04E. Peng-Jun Wan and Weiping Shang are supported in part by Research Grants Council of Hong Kong under Project Number CityU 122105. Peng-Jun Wan is also supported in part by the US NSF Grant No. 557904. Hongwei Du and Xiaohua Jia are supported in part by Research Grants Council of Hong Kong [Project No. CityU 114006] and NSF China [Project No. 60633020].

REFERENCES

[1] N. Alon, A. Bar-Noy, N. Linial, D. Peleg: A lower bound for radio broad-cast. *Journal of Computer and System Sciences*, 43:290-298, 1991.

[2] I. Chlamtac, S. Kutten: On broadcasting in radio networks - problem analysis and protocol design. *IEEE Transactions on Communications* 33 (1985) 1240–1246.

[3] I. Chlamtac, O. Weinstein. The wave expansion approach to broadcasting in multihop radio networks. *IEEE Trans. on Communications*, 39:426-433, 1991. A conference version appeared in *IEEE INFOCOM* 1987, pp. 874-881.

[4] F. Cicalese, F. Manne, and Q. Xin: Faster Centralized Communication in Radio Networks, *Proceedings of the 17th International Symposium on Algorithms and Computation* (ISAAC'2006), Springer LNCS 4288, pp.339-348.

[5] A. Dessmark and A. Pelc, Tradeoffs between knowledge and time of communication in geometric radio networks, in *Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA 2001)*, pp. 59–66, Crete, Greece, July 2001.

[6] M. Elkin, G. Kortsarz: A logarithmic lower bound for radio broadcast. *Journal of Algorithms* 52 (2004) 8–25.

[7] M. Elkin, G. Kortsarz: Polylogarithmic additive inapproximability of the radio broadcast problem. P*roc. 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'2004)*, pages 105–116. 2004.

[8] M. Elkin, G. Kortsarz: Improved broadcast schedule for radio networks. *Symposium on Discrete Algorithms (SODA)*, 2005, 222-231.

[9] R. Gandhi, S. Parthasarathy, A. Mishra: Minimizing broadcast latency and redundancy in ad hoc networks, in *Proceedings of the 4th ACM international symposium on Mobile Ad hoc networking and computing (MobiHoc 2003)*, 2003, pp. 222-232.

[10] I. Gaber, Y. Mansour: Centralized broadcast in multihop radio networks. *Journal of Algorithms* 46 (1) (2003) 1–20. A conference version of this paper appeared in *Proc. of 6th AnnualACM-SIAM Symposium on Discrete Algorithms (SODA'1995)*, 577-585, 1995.

[11] L.Gasieniec, D.Peleg, and Q.Xin: Faster communication in known topology radio networks, *Proceedings of The 24th Annual ACM Symposium on Principles of Distributed Computing* (PODC'2005), pp. 129-137.

[12] D. Kowalski, A. Pelc: Centralized deterministic broadcasting in undirected multi-hop radio network. In *Random-Approx 2004*, pages 171–182, 2004.

[13] D. Kowalski, and A. Pelc, Optimal deterministic broadcasting in known topology radio networks, *Distributed Computing* 19(3): 185-195 (2007).

[14] D. W. Matula and L. L. Beck. Smallest-last ordering and clustering and graph coloring algorithms, *Journal of the Association of Computing Machinery*, 30(3):417-427, 1983.

[15] P.-J. Wan, K.M. Alzoubi, and O.Frieder: Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks, *ACM/Springer Mobile Networks and Applications*, Vol. 9, No. 2, pp.141-149, 2004. A preliminary version of this paper appeared in *IEEE INFOCOM*, 2002.

[16] P.-J. Wan, C.-W. Yi, X. Jia, D. Kim: Approximation Algorithms for Conflict-Free Channel Assignment in Wireless Ad Hoc Networks, *Wiley Journal on Wireless Communications and Mobile Computing*, Volume 6, Issue 2 (March 2006), pp. 201-211.