Distributed Multiple-Message Broadcast in Wireless Ad-Hoc Networks under the SINR Model

Dongxiao Yu[1], Qiang-Sheng Hua[2], Yuexuan Wang[2], Haisheng Tan[2], and Francis C.M. Lau[1]

[1] Department of Computer Science, The University of Hong Kong,
Pokfulam, Hong Kong, P.R. China

[2] Institute for Interdisciplinary Information Sciences,
Tsinghua University, Beijing, 100084, P.R. China

Abstract. In a multiple-message broadcast, an arbitrary number of messages originate at arbitrary nodes in the network at arbitrary times. The problem is to disseminate all these messages to the whole network. This paper gives the first randomized distributed multiple-message broadcast algorithm with worst-case performance guarantee in wireless ad-hoc networks employing the SINR interference model which takes interferences from all the nodes in the network into account. The network model used in this paper also considers the harsh characteristics of wireless ad-hoc networks: there is no prior structure, and nodes cannot perform collision detection and have little knowledge of the network topology. Under all these restrictions, our proposed randomized distributed multiple-message broadcast protocol can deliver any message m to all nodes in the network in $O(D + k + \log^2 n)$ timeslots with high probability, where D is the network diameter, k is the number of messages whose broadcasts overlap with m, and n is the number of nodes in the network. We also study the lower bound for randomized distributed multiple-message broadcast protocols. In particular, we prove that any uniform randomized algorithm needs $\Omega(D + k + \frac{\log^2 n}{\log\log\log n})$ timeslots to deliver k messages initially stored at k nodes to all nodes in the network.

1 Introduction

In wireless networks, how to achieve efficient communications is one of the most extensively studied problems. The main challenge is to deal with interferences. Hence, the modeling of wireless interferences will play a fundamental role in designing efficient network protocols. Previous work mostly adopted the graph based or the protocol interference model. In the graph based interference model, it is assumed that only nodes within d (a small constant) hops from a receiver can interfere with the transmission. The protocol model assumes that a transmission can be successful if and only if there is only one transmitter within a certain range centered at the receiver. A shortcoming of these two types of models is that they treat interference as a localized phenomenon, which is not likely

G. Even and M.M. Halldórsson (Eds.): SIROCCO 2012, LNCS 7355, pp. 111–122, 2012.

the case in practice. In real wireless networks, the interference is cumulative, being contributed to by all simultaneously transmitting nodes. Because of the lack of the ability to capture the cumulative property of interference, the protocols designed under the graph based or protocol model display a dramatically different performance from the expectation in practice. In this paper, we adopt the SINR model (also known as the physical interference model since it reflects the physical reality more accurately), which defines a global interference function and takes into account the cumulative property of interference. Besides interference, some other important aspects in wireless ad-hoc networks should also be considered when modeling the network. For instance, when the network begins operation, no built-in infrastructure or MAC layer is available to the nodes to facilitate communication between neighboring nodes, and the nodes are clueless about the network topology. Furthermore, the nodes may not be able to perform any type of collision detection because they may be just tiny sensors with limited capabilities and energy [15].

In the multiple-message broadcast problem, an arbitrary number of messages arrive at arbitrary nodes from the environment at arbitrary times. The problem is to deliver all these messages to all the nodes. A multiple-message broadcast protocol can be used as a building block for many applications, e.g., update of routing tables, topology learning of the underlying network, and aggregating functions in sensor networks.

Different from most previous work, in this work, we do not assume that all messages are initially stored at their nodes. In addition, we adopt the realistic global SINR interference model and assume the imposition of such rigorous restrictions as no prior structure, no collision detection and nodes have little knowledge about the network topology; all these add to the challenge of designing an efficient distributed protocol. Under all these rigorous but practical restrictions, we present a randomized distributed multiple-message broadcast algorithm for wireless ad-hoc networks, and show that, with high probability, any message m can be broadcast to all nodes in $O(D + k + \log^2 n)$ timeslots after its arrival, where D is the diameter of the communication graph defined by the maximum transmission range R_M (refer to Section 3), k is the number of messages whose broadcast overlap m (refer to Section 3) and n is the number of nodes in the network.[1] To the best of our knowledge, this work is the first one that studies time efficient distributed multiple-message broadcast algorithms in wireless ad-hoc networks under the SINR model. Our result significantly surpasses the best known results of $\max\{O(k \log n \log \Delta + (D + n/\log n) \log n \log \Delta), O((k\Delta \log n + D) \log \Delta)\}$ [1,8] under the graph based interference model, and breaks the expected $\Omega(k + D \log(n/D))$ lower bound [4,10] for randomized solutions under the graph-based radio network model. Note that the previous results are obtained

[1] We define the running time of a multiple-message broadcast algorithm as the maximum number of timeslots to disseminate a message to the whole network. If all messages are initially stored at their nodes, our defined performance measurement is equivalent to that in previous work [1], which is the number of timeslots needed to broadcast all messages to all nodes.

with knowledge of some network parameters, e.g., Δ and D; In contrast, our algorithm does not assume any prior information concerning such parameters. The trivial $\Omega(D+k)$ lower bound indicates that our algorithm is asymptotically optimal for networks with diameter $D \in \Omega(\log^2 n)$.

Besides the proposed algorithm, we also study the lower bound of the time needed by randomized distributed algorithms to accomplish multiple-message broadcast. Specifically, we show that if all the nodes use the same transmission power, any uniform randomized algorithm in which all awaken nodes transmit a message with the same probability (independent of the communication history) in every timeslot [3] needs $\Omega(D+k+\frac{\log^2 n}{\log\log\log n})$ timeslots to accomplish multiple-message broadcast even under the assumption that all messages are initially stored at their nodes.

2 Related Work

Although the SINR model (or the physical interference model) poses great challenges for designing efficient distributed algorithms due to its global interference, there have been some attempts in recent years. In [14], with the assumption that all nodes can perform physical carrier sensing, an $O(\log n)$ time randomized distributed algorithm for computing a constant approximate dominating set was presented. The local broadcasting problem was first considered in [5]. In this paper, based on whether each node knows the number of nodes in its proximity region or not, the authors gave two randomized distributed algorithms with approximation ratios $O(\log n)$ and $O(\log^3 n)$, respectively. The latter result was improved by some recent papers [19,16], the latter of which achieves an approximation ratio of $O(\log n)$. By assuming that nodes can perform physical carrier sensing, the authors of [19] also gave two distributed deterministic local broadcasting algorithms both having an approximation ratio of $O(\log n)$ for asynchronous wake-up and synchronous wake-up scenarios. The distributed $(\Delta+1)$-coloring (Δ is the maximum network degree) was studied in [18] and an $O(\Delta \log n + \log^2 n)$ time randomized distributed algorithm was given. There are also recent papers on finding efficient distributed algorithms for the minimum latency aggregation scheduling problem [12,13] and the wireless scheduling problem [11,6].

The multiple-message broadcast problem is also called the the Many-to-All communication problem [4]. All previous work assumes the standard graph-based radio network model. In this model, there is a link existing between any pair of nodes that can communicate with each other. A transmission is successful iff there is only one neighbor transmitting a message to the receiver. Additionally, except [8,9], all work assumes that all messages are stored at their nodes at the beginning of the algorithm. The authors of [1] first initiated the study of this problem. They designed a randomized algorithm accomplishing multiple-message broadcast in $O(k \log n \log \Delta + (D + n/\log n) \log n \log \Delta)$ rounds in expectation. Assuming nodes receive messages at arbitrary times from the environment, the authors of [8] proved that their modular approach can broadcast a message to all nodes in $O((k\varDelta \log n + D)\log \varDelta)$ rounds with high probability when there are at most k concurrent messages. How to use network coding techniques to accelerate the multiple-message broadcast has been studied in [7], in which the proposed randomized algorithm achieves a time complexity of $O(k \log \varDelta + (D + \log n)\log \varDelta \log n)$. All the above work assume that nodes know some or all network parameters, e.g., $\varDelta$ and D. The best known lower bound for randomized solutions under the graph-based radio network model is $\varOmega(k + D\log(n/D))$ in expectation [4,10]. In the paper [9], by introducing an abstract MAC layer providing reliable local broadcast communication, the authors gave a multiple-message broadcast protocol for regional networks and showed that the protocol can broadcast a message to all nodes in $O((D + k)F_{prog} + (k - 1)F_{ack})$ rounds, where F_{prog} and F_{ack} are progress and acknowledgement bounds respectively.

3 Network Model and Problem Definitions

We assume there are n processors; they are the nodes of the network. During the protocol execution, the time is divided into slots. Processors have synchronized clocks and they have access to a global clock. We also assume that all processors wake up at the beginning of the protocol. We do not assume any placement distribution for nodes, i.e., nodes are arbitrarily placed on the plane. At the beginning, the network is completely unstructured. Nodes have very little information about the network topology. They have no knowledge about their neighbors, even the number of nodes in their proximity range. Only a polynomial estimate n of the number of nodes in the network is given to the nodes. Nodes have no collision detection mechanism. In other words, nodes can not distinguish between the occurrence of a collision and the case that there are no transmissions. We also assume that each node has a unique ID. The IDs need not be in the interval $[1, n]$, which are only used for a receiver to identify its sender. Furthermore, we assume that there is only one channel available and nodes operate in half-duplex mode, i.e., in a timeslot, a node can only carry out either one of the two operations: receive and transmit.

We adopt the SINR interference model. In this model, a message sent by node u to node v can be correctly received at v iff

$$\frac{\frac{P_u}{d(u,v)^\alpha}}{N + \sum_{w \in V \setminus \{u,v\}} \frac{P_w}{d(w,v)^\alpha}} \geq \beta, \tag{1}$$

where P_u (P_w) is the transmission power of node u (w); α is the path-loss exponent whose value is normally between 2 and 6; β is a hardware determined threshold which is greater than 1; N is the ambient noise; $d(u, v)$ denotes the Euclidean distance between u, v and $\sum_{w \in V \setminus \{u,v\}} \frac{P_w}{d(w,v)^\alpha}$ is the accumulated interference experienced by the receiver v caused by all other simultaneously transmitting nodes in the network.

Given transmission power P for v, the transmission range R_T of a node v is defined as the maximum distance at which a node u can receive a clear transmission from v ($SINR \geq \beta$) when there are no other simultaneous transmissions in the network. According to (1), $R_T \leq (\frac{P}{\beta \cdot N})^{1/\alpha}$. We further define $R_T = (P/cN\beta)^{1/\alpha}$, where $c > 1$ is a constant determined by the environment. Based on the transmission ranges of nodes, we define a communication graph $G = (V, E)$, where V is the set of nodes in the network, and a link $(u, v) \in E$ if and only if $d(u, v)$ is not larger than the transmission range of u. Furthermore, if all nodes have the same transmission range R_T, the obtained communication graph is denoted as G_{R_T}. Obviously, in this case, G_{R_T} can be seen as an undirected graph. We say a network is connected in terms of R if the communication graph G_R is connected. Let P_M and R_M be the maximum transmission power and the corresponding maximum transmission range of nodes respectively. Denote D as the diameter of the communication graph G_{R_M}.

Given a distance d, we say two nodes are independent if the distance between them is larger than d. An independent set I in terms of d is defined as a set of nodes such that any pair of nodes in I are independent. An independent set I is maximal in terms of d if for any node v in the network, either $v \in I$, or there is a node in I that is within distance d from v. A dominating set S in terms of d is defined as that for any node v, either $v \in S$, or there is a node in S that is within distance d from v. Denote G_d^S as the subgraph of G_d induced by S. A dominating set S is said to be connected in terms of d if G_d^S is connected. Note that a maximal independent set is a dominating set, but not a connected dominating set.

For a message m, denote $arrive(m)$ as the event that the message m arrives at the network, i.e., m is received by some node v. Denote $clear(m)$ as the event that the network has completed the broadcast of message m, i.e., all nodes in the network have received m. Then $K(m)$ is used to denote the set of messages whose processing overlaps with the interval between $arrive(m)$ and $clear(m)$. In other words, $K(m)$ is the set of messages $m^{'}$ such that an $arrive(m^{'})$ event precedes the $clear(m)$ event and the $clear(m^{'})$ event follows the $arrive(m)$ event. Let $k = |K(m)|$.

4 Algorithm

4.1 Algorithm Description

The proposed multiple-message broadcast algorithm adopts a clustering strategy, which encompasses four processes: leader election, leader coloring, local information collection and broadcast. The whole algorithm is divided into three stages. The first stage works as a pre-stage in which a CDS (Connected Dominating Set) in terms of distance R is computed, where $R = \min\{\frac{1}{2}, (\frac{48c\beta(2^{\alpha-1}+\frac{\alpha-1}{\alpha-2})}{c-1})^{-\frac{1}{\alpha}}\} \cdot R_M$. Nodes in the computed connected dominating set are called leaders. Other nodes are called non-leaders, each of which chooses the first leader that successfully transmits a dominating message to it as its leader. A cluster is composed of a leader and its dominated non-leaders. The parameter R is chosen to guarantee that if there is only one sender in each disk with radius R_M, each sender can successfully transmit a message to all nodes within distance R. The second stage is used to get a TDMA-like scheduling scheme for nodes to perform the local information collection process and the broadcast process in the next stage by performing a coloring. In the third stage, each leader first collects messages that are received from the environment by its dominated non-leaders. Then the messages are disseminated to the whole network through the backbone network composed by the leaders. During the execution of the protocol, each leader v is assigned a queuing set Q_v to store received messages. Next we describe the algorithm in more details.

Stage 1 Leader Election: This stage is to compute a connected dominating set in terms of R, the nodes of which form a backbone network for performing the broadcast in the Stage 3. As shown in [2], for a graph G, if we find connectors such that any pair of MIS (Maximal Independent Set) nodes within three hops are connected by these connectors, the MIS nodes and the connectors constitute a CDS. In this stage, the nodes first execute the MIS algorithm in [18] to compute an MIS in terms of $R/3$. Any two nodes within three hops in the computed MIS have distance at most R, so they are connected in terms of R. This stage takes $O(\log^2 n)$ timeslots. By the end of this stage, a CDS in terms of R is correctly computed with high probability[2]. Furthermore, we will show that the computed CDS satisfies the property that in any disk with radius R_M, there are only a constant number of leaders. Denote χ as a constant upper bound for the number of leaders in a disk with radius R_M. The value of χ will be given later.

Stage 2 Leader Coloring: In this stage, we want to find a coloring for leaders in the computed connected dominating set, such that for any two leaders, if the distance between them are not larger than R_M, they get different colors. Since for any leader, there are at most $\chi - 1$ other leaders within distance R_M, χ colors are enough to color all the leaders. We use a greedy coloring algorithm to accomplish the coloring process. The MIS algorithm in [18] is iteratively executed to get an MIS in terms of R_M from leaders that have not been colored. In the i-th execution, an MIS in terms of R_M is obtained from the remaining uncolored leaders and its nodes are assigned the color i. Finally, each leader gets a color from $\{1, 2, \ldots, \chi\}$. After the coloring is computed, there are χ timeslots for the leaders to inform their dominated non-leaders of their colors. Since each execution of the MIS algorithm needs $O(\log^2 n)$ timeslots, this stage takes at most $O(\chi \log^2 n) \in O(\log^2 n)$ timeslots.

Stage 3 Local Information Collection and Broadcast: This stage is for leaders to collect the messages that arrive at their dominated non-leaders and then disseminate the received messages to the whole network. The stage is divided into iterative substages, each of which consists of χ phases. In each substage, the i-phase is for leaders with color i to accomplish information collection and broadcast. Each phase has three timeslots. This TDMA-like scheduling makes sure that in each phase, any two leaders that are active have distance larger than R_M, so that each leader can successfully transmit an acknowledgement message to its dominated non-leaders and send the received message to all nodes within distance R. During the execution of this stage, all nodes use the same transmission power

[2] We assume that the network is connected in terms of $R/3$, i.e., the communication graph $G_{R/3}$ is connected.

$P_B = cN\beta R^{\alpha}$, which leads to the same transmission range R (refer to Section 3). Next we describe the detailed operations in the i-th phase of a substage.

The i-th Phase: In this phase, leaders with color i and non-leaders in their clusters execute a three-timeslot scheme as described in Algorithm 1. The first two timeslots are used for local information collection and the third is for the leaders to broadcast the received messages. In particular, in the first timeslot, each non-leader that has received a message from the environment endeavors to transmit the message to its leader with a specified transmission probability. Here the transmission probability is initially set as a small value $\frac{\lambda}{n}$, where λ is a constant to be given later. After Stage 3 has started, every non-leader u that is performing the local information collection process updates its transmission probability at every $9\chi\lambda^{-1}4^{2(\lambda+1)}\log n$-th timeslot as shown in Algorithm 1. The updating principle is set to guarantee that on one hand, nodes can increase the transmission probability, by which they can finally get a large enough transmission probability ensuring a successful transmission; on the other hand, the sum of transmission probabilities of nodes in any local region will not exceed a constant which is the base of obtaining a sufficient condition for successful transmissions. Furthermore, while a non-leader transmits the message, it also adds its ID and its leader's ID into the transmitted packet such that its leader can distinguish whether the received message is for it or not. In the second timeslot, if a leader v receives a message from one of its dominated non-leaders u, it stores the received message into Q_v and transmits an $Ack_v(u)$ message to inform u that it has received the message. A leader v also adds the messages received from other leaders into Q_v. After receiving the $Ack_v(u)$ message, u will stop transmitting and quit the local information collection process. In the third timeslot, for each leader v with color i, if Q_v is not empty, it transmits the first message in Q_v to all nodes within distance R and deletes it from Q_v.

In order to ensure that the above described multiple-message broadcast algorithm is correct with high probability, we set the parameters as follows: $\lambda = \frac{(1-1/c)}{192\beta} \cdot (2^{\alpha-1} + \frac{\alpha-1}{\alpha-2})^{-1}$, and $\chi = (\frac{6R_M}{R} + 1)^2$.

4.2 Analysis

In this section, we prove that with probability $1 - O(\frac{1}{n})$, for any message m, the proposed multiple-message broadcast algorithm can disseminate m to the whole network after $arrive(m)$ occurs for at most $O(D + k + \log^2 n)$ timeslots. Before starting the analysis, we first define some notations. We use T_v and I_v to denote the disks of radii R and R_M centered at node v, respectively. The notation E_v^d denotes the disk of radius d centered at v. Without confusion, we also use these notations to denote the nodes in the corresponding disks.

The following lemma is proved in [18], which states the correctness and efficiency of the MIS algorithm.

Lemma 1. *After executing the MIS algorithm for $O(\log^2 n)$ timeslots, a maximal independent set can be correctly computed with probability at least $1-O(n^{-1})$.*

In the following, we assume that the MIS is correctly computed, and the error probability will be summed up in the proof of the main theorem (Theorem 1).

Using a standard area argument, we can give a constant bound on the number of leaders in any disk I_i as shown in Lemma 2 which follows. For the detailed proof of the lemma, please refer to the full version [17].

Lemma 2. *In a disk I_i with radius R_M, the number of leaders is at most χ.*

Algorithm 1. 3-Timeslot Scheme

```
Initially, $p_u = \frac{\lambda}{2n}$; $Q_v = \emptyset$;
3-timeslot scheme for a leader $v$
listen
if $v$ received a message from a non-leader $u$ in its cluster then transmit $Ack_v(u)$
    end if
if $Q_v$ is not empty then transmit the first message in $Q_v$ and delete the message
    from $Q_v$ end if
Message Received
if $v$ received a message that has not been received then add the message into $Q_v$
    end if
3-timeslot scheme for a non-leader $u$
if $u$ has a message received from the environment then transmit the message with
    probability $p_u$
    end if
listen
if $u$ received $Ack_v(u)$ then stop transmitting and quit the local information collection process end if
Update $p_u$
while
    $t = i \cdot 9\chi\lambda^{-1}4^{2(\lambda+1)} \log n$ for some integer $i > 0$
    if $u$ has taken part in the local information collection process and received less
        than $12 \log n$ $Ack$ messages from its leader in the past $9\chi\lambda^{-1}4^{2(\lambda+1)} \log n$ timeslots
        then $p_u = 2p_u$
    else $p_u = p_u/2$
    end if
end while
```

Lemma 3. *In Stage 2, the coloring process only needs to execute the MIS algorithm for at most χ times, and each leader can get a color from $\{1, 2, \ldots, \chi\}$.*

Proof. By Lemma 2, for each leader v, there are at most $\chi - 1$ leaders within distance R_M. From the coloring process, we know that after executing the MIS algorithm once, for each leader v, either v joins the computed MIS and gets a color, or a leader in I_v joins the MIS and gets a color. Thus after executing the MIS algorithm for at most $\chi - 1$ times, either v has chosen its color, or all leaders in I_v have been colored. For the second case, v will join the MIS in the next execution of the MIS algorithm and will get a color from $\{1, 2, \ldots, \chi\}$. □

By Lemma 1 and Lemma 3, we have the following corollary which states the time needed for executing Stage 1 and Stage 2.

Corollary 1. *With probability $1 - O(n^{-1})$, the coloring computed in Stage 2 is correct, and the number of timeslots needed for Stage 1 and Stage 2 is $O(\log^2 n)$.*

Next we analyze the correctness of Stage 3 and the number of timeslots needed for implementing Stage 3. We first state in the following Lemma 4 that making use of the TDMA-like scheduling as shown in Stage 3 can guarantee successful local broadcast within distance R for a leader v. Due to the space limitation, the proof is given only in the full version [17].

Lemma 4. *In Stage 3, during executing the three-timeslot scheme, a leader v can successfully transmit a message to all its neighbors within distance R in the second and third timeslots.*

Before analyzing the timeslots needed for Stage 3, we first present the following property which gives a constant bound on the sum of transmission probabilities of non-leaders in any disk T_v centered at some leader v. The idea of the proof is to show in any disk T_v, when the sum of transmission probabilities of non-leaders is about to break the declared bound, with high probability, every non-leader in T_v must have received at least $12\log n$ *Ack* messages from the leader in the past $9\chi\lambda^{-1}4^{2(\lambda+1)}\log n$ timeslots. Then by the algorithm, these nodes will halve their transmission probabilities, which gurantees that the declared bound will not be broken during the execution of the algorithm with high probability. For the detailed proof, please refer to [17].

Property 1. In any timeslot during the execution of Stage 3, for any leader v, the sum of transmission probabilities of non-leaders in T_v is at most 2λ.

Based on the above property, we give an upper bound on the number of timeslots needed for the local information collection process in the following Lemma 5. Denote Δ_k^v as the number of messages in $K(m)$ that arrive at nodes within distance R from v. Let $\Delta_k = \max\{\Delta_k^v\}$ for all nodes v. Clearly, $\Delta_k \leq k$.

Lemma 5. *Assume that Property 1 holds. For a non-leader u, it can transmit its message to the leader after starting transmission for $O(\Delta_k + \log^2 n)$ timeslots with probability $1 - O(n^{-2})$.*

Proof. We first give a sufficient condition for a successful transmission from a non-leader u to its leader v in the following claim whose proof can be found in the full version [17].

Claim. If a non-leader u is the only transmitting node in T_v, v can successfully receive the message sent by u with probability at least $\frac{1}{2}$.

Next we bound the number of timeslots for u to successfully transmit its message to the leader. After Stage 3 has started, once a non-leader u receives a message from the environment, it transmits its message to the leader v in the corresponding phases by executing the three-timeslot scheme. As shown in Algorithm 1, after u starts executing the three-timeslot scheme, in every $9\chi\lambda^{-1}4^{2(\lambda+1)}\log n$ timeslots, either u receives at least $12\log n$ *Ack* messages from v and halves its transmission probability, or u doubles its transmission probability. Thus, after $6\chi\lambda^{-1}4^{2\lambda+1}\Delta_k + 9\chi\lambda^{-1}4^{2(\lambda+1)}\log^2 n$ timeslots, if u does not receive an $Ack_v(u)$ message, it must have a constant transmission probability $\lambda/2$, since u halves its transmission probability for at most $\frac{\Delta_k}{12\log n}$ times, and therefore it needs $\frac{\Delta_k}{12\log n} \cdot 9\chi\lambda^{-1}4^{2(\lambda+1)}\log n = 3\chi\lambda^{-1}4^{2\lambda+1}\Delta_k$ timeslots to increase the transmission probability to the initial value. Then based on the sufficient condition for a successful transmission in the above claim, we will show that the probability P_{no} that u can not send its message to v in the subsequent $6\chi\lambda^{-1}4^{2\lambda+1}\log n$ timeslots is at most $O(n^{-2})$. Let P_{only} denote the probability that u is the only transmitting node in T_v, and we have

$$\begin{aligned} P_{only} &\geq p_u \prod_{w \in T_v \setminus \{u\}} (1 - p_w) \geq p_u \cdot \left(\frac{1}{4}\right)^{\sum_{w \in T_v \setminus \{u\}} p_w} \\ &\geq \frac{\lambda}{2}\left(\frac{1}{4}\right)^{\sum_{w \in T_v} p_w} \geq \frac{\lambda}{2}\left(\frac{1}{4}\right)^{2\lambda}. \end{aligned} \tag{2}$$

The last inequality is derived by Property 1. So the probability that u can not send its message to v in the subsequent $2\lambda^{-1}4^{2\lambda+1}\log n$ transmissions is at most $P_{no} \leq (1 - \frac{1}{2} \cdot \frac{\lambda}{2}\left(\frac{1}{4}\right)^{2\lambda})^{2\lambda^{-1}4^{2\lambda+1}\log n} \leq e^{-\frac{\lambda}{4}\left(\frac{1}{4}\right)^{2\lambda} \cdot 2\lambda^{-1}4^{2\lambda+1}\log n} \leq n^{-2}$.

Therefore, after starting transmission for $6\chi\lambda^{-1}4^{2\lambda+1}\Delta_k + 9\chi\lambda^{-1}4^{2(\lambda+1)}\log^2 n + 6\chi\lambda^{-1}4^{2\lambda+1}\log n$ timeslots, with probability $1 - n^{-2}$, u must have successfully transmitted its message to v, since each non-leader transmits once in every 3χ timeslots. □

The following Lemma 6 is given in [9], which analyzes the pipelining effect of the multiple-message broadcast process. Let F_{prog} denote the maximum number of timeslots needed for a successful transmission. For a graph G, define $d_G(u, v)$ as the number of edges in the shortest path from u to v in G.

Lemma 6. *Assume that in timeslot t_0, a node u receives a new message m. Let v be a node at distance $d = d_G(u, v)$ from v. For integers $l \geq 1$, we define $t_{d,l} = t_0 + (d + 2l - 2)F_{prog}$. Then for all integers $l \geq 1$, at least one of the following two statements is true:*

(i) v received the message m by the time $t_{d,l}$;

(ii) there exists a set $M \subseteq K(m)$, $|M| = \min\{l, k\}$, such that for every $m' \in M$, v has received m' by the timeslot $t_{d,l}$.

Theorem 1. *With probability $1 - O(n^{-1})$, any message m can be broadcast to all nodes in the network after the event arrive(m) occurs for $O(D + k + \log^2 n)$ timeslots.*

Proof. By Corollary 1, Stage 1 and Stage 2 need $O(\log^2 n)$ timeslots. So if m arrives at the network before Stage 3 starts, it waits for $O(\log^2 n)$ timeslots before Stages 1 and 2 complete. Next we analyze the timeslots needed for m to be broadcast to the whole network in Stage 3.

As proved in Lemma 5, in Stage 3, if m arrives at a non-leader u, with probability $1 - O(n^{-2})$, u can send m to its leader in $O(\Delta_k + \log^2 n)$ timeslots. And after that, m will be broadcast by each leader that received m to all its neighbors within distance R in the corresponding phases. If m arrives at a leader, the broadcast process starts after its arrival. By Algorithm 1, each leader broadcasts one message for one timeslot in every χ phases. And by Lemma 4, a leader can successfully transmit the message to all its neighbors within distance R. So it only takes a constant number of timeslots for m to be propagated for one hop in the communication graph G_R. Note that R is constant fraction of R_M, the diameter of G_R is $O(D)$. Then by Lemma 6, after $O(D + 2k - 2)$ timeslots, either all nodes have received m, or all nodes have received k messages. By the definition of $K(m)$ and $k = |K(m)|$, m must have been received by all nodes. Combining all these, with probability $1 - O(n^{-2})$, m can be broadcast to the whole network after arriving at the network for $O(D + k + \log^2 n)$ timeslots. Since k is at most $O(n)$, this is true for any message with probability $1 - O(n^{-1})$.

Note that all the above discussions are based on Property 1 and the assumption that the connected dominating set and the coloring are correctly computed. Property 1 is shown to be correct with probability $1 - \frac{1}{n}$ (refer to [17]). Furthermore, by Lemma 1 and Corollary 1, we have known that in Stages 1 and 2, the connected dominating set and the coloring is correctly computed with probability $1-O(n^{-1})$. Thus with probability $1-O(n^{-1})$, any message m can be broadcast to all nodes after the event *arrive*(m) occurs for $O(D + k + \log^2 n)$ timeslots.

5 Lower Bound

In this section we give a lower bound on the time needed for a uniform randomized distributed algorithm to accomplish multiple-message broadcast. Recall that a randomized algorithm is called uniform if all awake nodes transmit a message with the same probability (independent of the communication history) in every timeslot. The proof of Theorem 2 can be found in [17].

Theorem 2. *Assume that all nodes use the same transmission power. Then, any uniform randomized multiple-message broadcast algorithm requires* $\Omega(D + k + \frac{\log^2 n}{\log \log \log n})$ *timeslots to disseminate all messages to the whole network with probability* $1 - \frac{1}{n}$.

6 Conclusion

In this paper, assuming a practical network model for wireless ad-hoc and sensor networks as well as the SINR interference model, we propose the first randomized distributed multiple-message broadcast algorithm for networks with arbitrary message arrivals. In particular, we show that the proposed algorithm can disseminate any message m to the whole network in $O(D + k + \log^2 n)$ timeslots if there are at most k overlapping messages. We also show that any uniform randomized algorithm needs at least $\Omega(D + k + \frac{\log^2 n}{\log \log \log n})$ timeslots to accomplish multiple-message broadcast. Our algorithm outperforms all previous best known results [1,8], and breaks the $\Omega(k + D \log(n/D))$ lower bound [4,10] under the graph based radio network model. An important feature of the proposed algorithm is that, in contrast with the previous work, it does not need any neighboring information, e.g., an estimate on Δ. Based on the trivial $\Omega(D + k)$ lower bound, our algorithm is optimal for networks with diameter $D \in \Omega(\log^2 n)$. Therefore an obvious research direction is to design a faster multiple-message broadcast protocol for networks with diameter $D = o(\log^2 n)$. Another direction is to consider deterministic distributed algorithms for the multiple-message broadcast problem under the SINR model.

Acknowledgements. This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00302, the National Natural Science Foundation of China Grant 61103186, 61073174, 61033001, 61061130540, the Hi-Tech research and Development Program of China Grant 2006AA10Z216, and Hong Kong RGC-GRF grants 714009E and 714311.

References

1. Bar-Yehuda, R., Israeli, A., Itai, A.: Multiple communication in multi-hop radio networks. SIAM Journal on Computing 22, 875–887 (1993)
2. Censor-Hillel, K., Gilbert, S., Kuhn, F., Lynch, N.A., Newport, C.C.: Structuring unreliable radio networks. In: Proc. PODC (2011)
3. Chlebus, B., Gasieniec, L., Lingas, A., Pagourtzis, A.T.: Oblivious gossiping in ad-hoc radio networks. In: Proc. DIALM (2001)
4. Chlebus, B., Kowalski, D., Radzik, T.: Many-to-Many Communication in Radio Networks. Algorithmica 54(1), 118–139 (2009)
5. Goussevskaia, O., Moscibroda, T., Wattenhofer, R.: Local broadcasting in the physical interference model. In: Proc. DialM-POMC (2008)
6. Halldórsson, M.M., Mitra, P.: Optimal Bounds for Distributed Wireless Scheduling in the SINR Model. In: Proc. ICALP (2011)
7. Khabbazian, M., Kowalski, D.R.: Time-efficient randomized multiple-message broadcast in radio networks. In: Proc. PODC (2011)
8. Khabbazian, M., Kuhn, F., Kowalski, D.R., Lynch, N.A.: Decomposing broadcast algorithms using abstract MAC layers. In: Proc. DialM-POMC (2010)
9. Kuhn, F., Lynch, N.A., Newport, C.C.: The abstract MAC layer. Distributed Computing 24(3-4), 187–206 (2011)
10. Kushilevitz, E., Mansour, Y.: An $\Omega(D \log(n/D))$ lower bound for broadcast in radio networks. SIAM J. Comput. 27(3), 702–712 (1998)
11. Kesselheim, T., Vöcking, B.: Distributed Contention Resolution in Wireless Networks. In: Lynch, N.A., Shvartsman, A.A. (eds.) DISC 2010. LNCS, vol. 6343, pp. 163–178. Springer, Heidelberg (2010)
12. Li, H., Hua, Q.-S., Wu, C., Lau, F.C.M.: Minimum-latency aggregation scheduling in wireless sensor networks under physical interference model. In: Proc. MSWiM (2010)
13. Li, H., Hua, Q.-S., Wu, C., Lau, F.C.M.: Latency-Minimizing Data Aggregation in Wireless Sensor Networks under Physical Interference Model. Ad Hoc Networks (to appear)
14. Scheideler, C., Richa, A., Santi, P.: An $O(\log n)$ dominating set protocol for wireless ad-hoc networks under the physical interference model. In: Proc. MOBIHOC (2008)
15. Woo, A., Culler, D.-E.: A transmission control scheme for media access in sensor networks. In: Proc. MOBICOM (2001)
16. Yu, D., Hua, Q.-S., Wang, Y., Lau, F.C.M.: An $O(\log n)$ Distributed Approximation Algorithm for Local Broadcasting in Unstructured Wireless Networks. In: Proc. DCOSS (2012)
17. Yu, D., Hua, Q.-S., Wang, Y., Tan, H., Lau, F.C.M.: Distributed Multiple-Message Broadcast in Wireless Ad-Hoc Networks under the SINR Model, `http://iiis.tsinghua.edu.cn/~qshua/sirocco12full.pdf`
18. Yu, D., Wang, Y., Hua, Q.-S., Lau, F.C.M.: Distributed $(\Delta + 1)$-Coloring in the Physical Model. In: Erlebach, T., Nikoletseas, S., Orponen, P. (eds.) ALGOSENSORS 2011. LNCS, vol. 7111, pp. 145–160. Springer, Heidelberg (2012)
19. Yu, D., Wang, Y., Hua, Q.-S., Lau, F.C.M.: Distributed local broadcasting algorithms in the physical interference model. In: Proc. DCOSS (2011)