Speedup of Information Exchange using Multiple Channels in Wireless Ad Hoc Networks

Dongxiao Yu*, Yuexuan Wang†*, Yu Yan‡, Jiguo Yu§ and Francis C.M. Lau*

*Department of Computer Science, The University of Hong Kong, Hong Kong, P.R. China
Email: {dxyu,amywang,fcmlau}@cs.hku.hk
†College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, P.R. China
‡Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, 100084, P.R. China
Email: yanyu02@gmail.com
§School of Information Science and Engineering, Qufu Normal University, Rizhao, 276826, P.R. China
Email: jiguoyu@sina.com

***Abstract*—This paper initiates the study of distributed information exchange in multi-channel wireless ad hoc networks. Information exchange is a basic operation in which each node of the network sends an information packet to other nodes within a specific distance R. Our study is motivated by the increasing presence and popularity of wireless networks and devices that operate on multiple channels. Consequently, there is a need for a better understanding of how and by how much multiple channels can improve communication. Based on the SINR interference model, we propose a multi-channel network model which incorporates certain features commonly seen in wireless ad hoc networks, including asynchrony, little non-local knowledge, limited message size, and limited power control. We then present a randomized algorithm that can accomplish information exchange in $O((\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log n}{\mathcal{P}}) \log n + \log \Delta \log n)$ timeslots with high probability, where n is the number of nodes in the network, Δ is the maximum number of nodes within the range R, $\mathcal{F}$ is the number of available channels and $\mathcal{P}$ is the maximum number of packets that can fit in a message. Our algorithm significantly surpasses the best known results in single-channel networks, achieving a $\Theta(\mathcal{F})$ times speedup if Δ and $\mathcal{P}$ are sufficiently large. We conducted empirical studies that confirmed the performance of the proposed algorithm as derived in the analysis.**

I. Introduction

Information exchange is one of the most fundamental operations that is frequently called for in the smooth running of a network. More precisely, in a network with n nodes (wireless devices), each node intends to send an information packet to all nodes within a specific range R, and the goal is to schedule all these requests in the minimum time. Such an operation is needed in many applications: for example, when a network begins operation, each node can use the algorithm to acquire the IDs of its neighbors. An efficient information exchange algorithm can also be used as a building block in many upper-layer applications, such as information broadcast, routing, network topology learning, and service/resource discovery.

Information exchange is obviously an age-old problem, but with the emergence of new capabilities in the wireless landscape, the problem presents new challenges that were not present before. In this work, we study in particular the effects of *multiple channels* on information exchange. We are motivated by the increasing availability of wireless services and devices that operate on multiple channels (e.g., Wi-Fi [13] and Bluetooth [1]). Although distributed information exchange has been extensively studied in single-channel networks [2], [7], [9], [12], [19], [22], it is largely unexplored in the multiple-channel domain, and the big question is *whether using multiple channels can effectively speed up information exchange.* This paper proposes an answer to this question, affirmatively. We consider the case of ad hoc networks (many wireless sensor networks are of this kind) where the nodes are simple devices with only one (tunable) transceiver that can operate on only one channel at a time. This is a common setting seen in the literature, which however could demand a rather non-trivial algorithmic design in making effective use of multiple channels [3]. In fact, it has been shown that for some problems, such as multi-hop wake-up [5] (waking up sleeping nodes by passing them messages), the utilization of multiple channels does not yield faster algorithms.

All wireless algorithms need to face the harsh reality of interference. We assume the now-popular SINR interference model [10]. The SINR model uses global fading interference from all the simultaneous transmissions to determine the success of a signal reception. Compared with traditional graph-based models, where interference is trivialized as being local and binary, the SINR model represents the physical interference in real networks more precisely. To the best of our knowledge, this is the first study on the impacts of multiple channels on distributed information exchange under the SINR model.

To gear our theoretical pursuit more closely in line with the reality, we add to our model several restrictions commonly seen in wireless ad hoc networks, including limited knowledge on the network topology in the nodes, limited message size and limited power control. Furthermore, as it is unreasonable to assume that all nodes would start the information exchange at the same time, we assume an asynchronous setting—that is, nodes may join the algorithm execution at any moment in time, and there is no global clock to rely on for any sort of synchronization among the nodes.

978-1-4799-8381-0/15/$31.00 ©2015 IEEE

A. Network Model and Problem Definitions

The network consists of n nodes in a set V. Nodes are arbitrarily placed in a two-dimensional Euclidean space. Each node has a unique ID. For two nodes u and v, denote by $d(u,v)$ the distance between them.

Multiple Communication Channels and Asynchrony. Nodes communicate through a shared medium which is divided into $\mathcal{F}$ channels $\{1, 2, \cdots, \mathcal{F}\}$. There is no global clock and nodes operate with their own local clocks. Nodes may wake up asynchronously to join the operation at any time. For the convenience of the analysis, we divide the time into timeslots. Note that our algorithm really does not rely on synchronization in any way, and the proposed algorithm would work correctly when timestepping is replaced by local clock values in real implementation. Also, by the standard argument introduced in [15] for slotted vs. unslotted ALOHA, the realistic unslotted case and the idealized slotted case differ only by a constant factor. Each node is equipped with a single half-duplex radio transceiver, such that in a timeslot, each node can select only one of the $\mathcal{F}$ channels to listen to or transmit on, but not both. This is common in many real devices. A node operating on a channel learns nothing about events on the other channels.

SINR model. Simultaneous transmissions on the same channel interfere with each other. We adopt the SINR interference model here. A message sent by node u to node v can be correctly received by v iff (i) u and v operate on the same channel and v does not transmit, and (ii) the following SINR ratio $SINR(u,v)$ is above a threshold $\beta \geq 1$ defined by the hardware.

$$SINR(u,v) = \frac{P_u/d(u,v)^\alpha}{N + \sum_{w \in S \setminus \{u,v\}} P_w/d(w,v)^\alpha} \geq \beta, \quad (1)$$

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, N is the ambient noise, and $\sum_{w \in S \setminus \{u,v\}} \frac{P_w}{d(w,v)^\alpha}$ is the interference experienced by the receiver v caused by nodes transmitting simultaneously on the same channel.

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 on the same channel. For a given power level P, from the SINR condition (1), $R_T \leq (\frac{P}{\beta \cdot N})^{1/\alpha}$. We define $R_T = (P/cN\beta)^{1/\alpha}$, where $c > 1$ is a constant determined by the environment. This assumption is a common one in related studies (cf. [14]) for dealing with weak capability nodes. It means that a received message can be successfully decoded only if the received power is a bit larger than the calculated ideal level.

Problem and Complexity Measure. In the information exchange problem, each node tries to send an information packet to all participating nodes within a given range R. We define a node v's running time as the length of the interval from the timeslot when v starts executing the algorithm to the timeslot when v's packet has been disseminated to all nodes within distance R, and the time complexity of the algorithm is the maximum of such running times.

TABLE I
NOTATIONS

n	#nodes	$\mathbb{C}_i$	competing state
$\mathcal{F}$	#channels	$\mathbb{B}_i$	broadcaster state
$\mathcal{P}$	message size(#packets)	$\mathbb{L}$	listening state
$d(u,v)$	dist. of u and v	$\mathbb{T}$	transmitting state
R	information exchange range	Δ	the max #nodes within dist. R from any node

Power Control. Nodes can only adjust their transmission power by up to a constant factor. This assumption considers in reality, arbitrary power control is very hard to implement given the wide variety of chipsets. In our algorithm, we only use two transmission power levels which differ by a factor of 2^α, where α is the path-loss exponent in the SINR model.

Message Size. Each message can carry a payload of at most $\mathcal{P}$ packets, where $\mathcal{P}$ is an arbitrary given integer. Here we do not impose any restriction on $\mathcal{P}$, and leave it open as to whether some special values of $\mathcal{P}$ may lead to better algorithms.

Capability and Knowledge of Nodes. Nodes have limited knowledge about the network topology and have only the simplest capabilities. In particular, each node only knows its own ID, the number of nodes n in the network, and the maximum number of nodes Δ within distance R from any node.[1] Nodes know nothing about the actual network topology, their neighbors (nodes within distance R), or their locations. They are not equipped to perform collision detection, nor physical carrier sensing.

Notation. Two nodes u, v, $d(u,v) \leq r$, are r-neighbors. Denote by Δ_v the number of R-neighbors of v. We write $\Delta = \max_{v \in V} \Delta_v$. Given the transmission power of each node, we can obtain a communication graph $G = (V, E)$, which consists of all nodes and edges (u,v) for all pairs of nodes u and v within the transmission range of each other. Then a set of nodes S is an independent set in graph G if there is no edge between any pair of nodes in S. An independent set S is maximal if for each node $w \notin S$, there is a node $v \in S$ such that w and v are connected by an edge. Given that all nodes have the same transmission range r, the (maximal) independent set in the obtained communication graph is called an r-(maximal) independent set. We summarize frequently used notations in Table I for the reader's convenience.

B. Our Contributions

This paper gives the first distributed algorithm for information exchange in multi-channel networks under the SINR model. Our algorithm adopts an *indirect* approach to accomplish information exchange. In contrast, in previous works, each node *directly* transmits its packet to its R-neighbors, which results in a trivial $\Omega(\Delta)$ lower bound, even when unbounded-size messages and multiple channels are assumed.

[1] Strictly speaking, our algorithm does not rely on Δ. Δ is only used in determining the number of channels that can help the algorithm to get the best result. Instead of Δ, we could use n instead, which will only change the last additive term $\log \Delta \log n$ in the running time to $\log^2 n$.

In our algorithm, nodes send out their packets indirectly. Each node v only needs to receive messages from a small number of neighboring nodes (called broadcasters and bounded by $O(\mathcal{F})$) which have collected the packets from v's R-neighbors. By making use of multiple channels in packet collection, the indirect strategy can break the $\Omega(\Delta)$ lower bound.

The main results are summarized as follows.

(1) We present a distributed algorithm that can accomplish information exchange in $O((\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log n}{\mathcal{P}}) \log n + \log \Delta \log n)$ timeslots with high probability ($1 - n^{-\epsilon}$ for some constant $\epsilon > 0$). Our algorithm demonstrates that if $\mathcal{P} \in \Omega(\mathcal{F} \log n)$, information exchange can be speeded up by $\Theta(\mathcal{F})$ times in dense networks, using the best known $O(\Delta \log n)$ result [9] in single-channel networks under the SINR model as the baseline.

(2) Through simulation, we observe that the constant hidden in the Big-O notation of the running time is very small. The simulation results also show that our algorithm is robust to the changing parameters in the SINR model. Furthermore, in comparison with algorithms using one single channel, the simulation confirms that our algorithm yields an approximate $\mathcal{F}$ times speedup or more given $\mathcal{F}$ available channels, and our algorithm is robust against network topologies with different densities.

II. Related Work

In the distributed computing community, the information exchange problem has been studied since 1990s [2]. All known works focus on single-channel networks, where the information exchange problem is also known as the local broadcast problem [9]. Alon et al. are either the first or among the first ones to study the information exchange problem [2]. Assuming a graph-based model in which nodes wake up and start the algorithm synchronously, they gave a randomized algorithm that can accomplish information exchange in $O(\Delta \log n)$ rounds. Derbel and Talbi [7] later generalized their algorithm to work in the case where there is no knowledge of Δ. Goussevaskaia et al. initiated the studies on distributed information exchange under the SINR model in [9]. Under an asynchronous setting similar to this work, they gave randomized distributed algorithms with running time $O(\Delta \log n)$ and $O(\Delta \log^3 n)$ for cases with and without knowing Δ, respectively. The latter result was improved in [22], where Yu et al. proposed an algorithm that can accomplish information exchange in $O(\Delta \log^2 n)$ timeslots with high probability. Under the same model, the result was further improved to $O(\Delta \log n + \log^2 n)$ in [12] and [19] independently.

There are also some other recent works facing up to the challenges posed by the SINR model, and efficient algorithms have been proposed for various fundamental problems in wireless networks, e.g., the dominating set problem [16], the contention resolution problem [11], the broadcast problem [14], [20], [21] and the node coloring problem [6], [23]. But all these works are cast in single-channel networks.

The study of distributed algorithms in multi-channel wireless networks is relatively recent. All known works are done under graph-based models. It has been shown that the availability of multiple channels can help get faster algorithms for some fundamental problems, such as the maximal independent set problem [3], the leader election problem [4], information dissemination in single-hop networks [17], [18] and the broadcast problem [8], but multiple channels cannot help the case of the multi-hop wake-up problem [5].

III. Information Exchange Algorithm

In this section, we present our Multi-channel Information Exchange (MIE) algorithm. In the algorithm, we assume that $\mathcal{F} \leq 2\sqrt{\Delta / \log n} + 1$, or otherwise, we would just use this number of channels since more channels does not help improve the running time of the algorithm, as shown in the analysis. The algorithm uses an indirect strategy to accomplish information exchange that makes effective use of the multiple channels. At the beginning, the nodes elect a small set of leaders called *broadcasters* whose transmission ranges cover all the other nodes which are the *submitters*. Each submitter v then transmits its packet to one of its R-neighboring broadcasters. It is easy to see that this broadcaster's $2R$-neighborhood covers all v's R-neighbors; v's packet is then sent to all its R-neighbors by this broadcaster using a transmission range of $2R$. For broadcasters, they directly send their packets to all their R-neighbors.

We make submitters use multiple channels to send their packets to neighboring broadcasters. In each timeslot, each submitter will select an operating channel (from $\frac{\mathcal{F}-1}{2}$ channels) uniformly at random; this should reduce the contention on each channel. But the problem then is that a submitter's chosen channel may not coincide with any of its neighboring broadcasters' channels. Likewise, we need also to ensure that the broadcasters and the submitters operate on the same channel when the broadcasters disseminate their stored packets subsequently, such that each broadcaster can send the packets to all its $2R$-neighbors. Our solutions to these problems are: (1) We elect multiple sets of broadcasters. Specifically, for each channel that the submitters use to send their packets, a *maximal* independent set of broadcasters is elected, which will listen on this particular channel to collect packets from R-neighboring submitters. In this way, a submitter can send its packet to an R-neighboring broadcaster as long as it successfully transmits on an arbitrary channel. This is also the key feature contributing to the efficiency of our algorithm. (2) We set aside a special channel (the $\mathcal{F}$-th channel) for broadcasters to disseminate received packets to their $2R$-neighbors. All nodes will listen on this special dissemination channel to receive packets from neighboring broadcasters. Note that although multiple sets of broadcasters are elected, the number of broadcasters in each node's $2R$-neighborhood is not that large due to the independence of each set of broadcasters. This characteristic ensures that the contention on the dissemination channel would not be high and broadcasters can disseminate their packets quickly, which makes multiple dissemination channels unnecessary. Moreover, if using multiple dissemination channels, a submitter may fail to receive the message from its neighboring broadcasters for the reason that they choose different channels.

Nodes go through four different states in the process: the competing state $\{\mathbb{C}_i\}$, the broadcaster state $\{\mathbb{B}_i\}$, the (submitter's) transmitting state $\mathbb{T}$ and the (submitter's) listening state $\mathbb{L}$. Meanwhile, the first $\mathcal{F}-1$ channels are divided into $(\mathcal{F}-1)/2$ pairs. [2] Each pair of channels consists of a primal channel and a dual channel, and the i-th pair of channels is denoted as $\{i_p, i_d\}$. The primal channel i_p is for packet transmissions by the submitters, and the dual channel i_d is used for the competition of nodes to become a broadcaster that listens to the primal channel i_p. A node in state $\mathbb{C}_i$ means that it is competing for becoming a broadcaster on the primal channel i_p by executing a Maximal Independent Set (MIS) algorithm. A node in state $\mathbb{B}_i$ means that it is a broadcaster that listens to channel i_p to collect packets from R-neighboring submitters. Therefore, $\{\mathbb{C}_i\}$ and $\{\mathbb{B}_i\}$ both have $(\mathcal{F}-1)/2$ states in total. A node in state $\mathbb{T}$ means that it is a submitter and it is trying to transmit its packet to an R-neighboring broadcaster. A node in state $\mathbb{L}$ means that it has sent its packet to an R-neighboring broadcaster and is listening to the special dissemination channel to receive packets from $2R$-neighboring broadcasters.

For each node v, after waking up, it first goes through states $\{\mathbb{C}_i\}$ in sequence. If v is elected as a broadcaster in some state $\mathbb{C}_i$ after executing an MIS algorithm on the dual channel i_d, it joins state $\mathbb{B}_i$. Otherwise, v joins state $\mathbb{T}$, which means that v has at least one R-neighboring broadcaster in each state $\mathbb{B}_i$ for $1 \leq i \leq (\mathcal{F}-1)/2$. Then after transmitting its packet on some primal channels for $\Theta(\log n)$ times, v joins state $\mathbb{L}$, and listens on channel $\mathcal{F}$ to receive packets from $2R$-neighboring broadcasters from then on. We next introduce the operations in each state, which are also given in full details in Algorithm 1 and 2. In the algorithm, Greek letters represent constants. To ensure the high probability guarantees of the algorithm and to simplify the presentation of the analysis, we assume that all constant parameters are sufficiently large. But in practice, these constant parameters can be set as reasonable values.

Competing state $\mathbb{C}_i$: Nodes in $\mathbb{C}_i$ operate on the dual channel i_d on which they determine whether to become broadcasters in state $\mathbb{B}_i$. From a global view, the broadcasters in each broadcaster state $\mathbb{B}_j$ (collecting packets on the primal channel j_p) need to be a dominating set, whose R-neighborhoods cover all submitters in the network. So here broadcasters are elected by letting submitters execute an MIS algorithm such that the broadcasters on i_p constitute an R-maximal independent set.

For a node v in state $\mathbb{C}_i$, after listening for $\gamma \log n$ timeslots to determine whether there has been any R-neighboring broadcaster in state $\mathbb{B}_i$, it executes the MIS algorithm given in [23] to determine to join states $\mathbb{B}_i$, $\mathbb{C}_{i+1}$ or $\mathbb{T}$. The basic idea of the MIS algorithm is that through competition, the number of competitors is reduced until there is only one active node left in each local range. After joining the MIS, a node makes all its neighbors stop the competition by sending a controlling message. For further details, please refer to [23].

The transmission power of the nodes in state $\mathbb{C}_i$ is set to $P_C = cN\beta R^\alpha$, whose transmission range is R by the definition in Section I-A.

Broadcaster state $\mathbb{B}_i$: The broadcasters perform three operations: first, they transmit on the dual channel i_d to make neighboring nodes that newly join state $\mathbb{C}_i$ stop their competition; second, they collect packets transmitted by submitters on the primal channel i_p; third, they disseminate the received packets on the special dissemination channel (Channel $\mathcal{F}$). In each timeslot, we let each node v in state $\mathbb{B}_i$ select an operating channel from $\{i_d, i_p, \mathcal{F}\}$ uniformly at random. On channel i_d, v transmits a controlling message $\mathcal{M}_B^i$ with a constant probability $\frac{1}{\rho}$. On channel i_p, v only listens to receive packets from neighboring submitters. It stores all received packets in a set B_v; initially B_v only contains v's own packet. On the special dissemination channel $\mathcal{F}$, if $|B_v| \geq \mathcal{P}$, the broadcaster transmits a message consisting of the first $\mathcal{P}$ packets of B_v (i.e., in FIFO fashion) with probability p_b. If $|B_v| < \mathcal{P}$, we set a threshold T on the number of timeslots during which v was not in the transmission process ($s_v = 0$ in Algorithm 2) to determine whether v should transmit or not. The setting of the threshold T ensures that each broadcaster does not transmit messages containing less than $\mathcal{P}$ packets too frequently.

To ensure that the transmitted message can be received by neighboring nodes with high probability, v transmits each message for $\theta \log n$ times. v adjusts the transmission probability after every $\gamma \log n$ timeslots based on the number of received messages (the value of m_c) from neighboring broadcasters. Furthermore, note that broadcasters may stop and restart their transmission during the algorithm's execution. To reduce the impact of a broadcaster restarting a transmission on the other broadcasters' transmissions, the transmission probability of a broadcaster is reset to the initial value every time when it restarts.

The transmission power P_B of broadcasters is set as follows: if they operate on a dual channel, $P_B = cN\beta R^\alpha$ with a transmission range of R; if they operate on the dissemination channel, $P_B = cN\beta(2R)^\alpha$ with a transmission range of $2R$.

Submitter transmitting state $\mathbb{T}$**.** A submitter in state $\mathbb{T}$ endeavors to send its packet to an R-neighboring broadcaster by transmitting on primal channels, and listens to the special dissemination channel $\mathcal{F}$ to receive packets transmitted by broadcasters. For a submitter v in state $\mathbb{T}$, it selects to operate on the special dissemination channel and primal channels with probability 1/2, respectively. If v selects channel $\mathcal{F}$, it just listens; otherwise, it chooses an operating primal channel from $\{1_p, 2_p, \ldots, (\frac{\mathcal{F}-1}{2})_p\}$ uniformly at random, and transmits its packet on this selected channel with probability p_s. v's transmission probability p_s is adjusted after every $\Theta(\log n)$ timeslots based on the number of messages received from neighboring submitters.

As discussed before, each submitter v in state $\mathbb{T}$ has an R-neighboring broadcaster in each state $\mathbb{B}_i$. Thus, the transmission power of v is set as $P_T = cN\beta R^\alpha$ with a transmission range of R.

[2] We assume that $\mathcal{F}$ is odd. Otherwise, we use only the first $\mathcal{F}-1$ channels.

Submitter listening state $\mathbb{L}$. In each timeslot, nodes in state $\mathbb{L}$ listen on Channel $\mathcal{F}$.

```
Algorithm 1: MIE: a node $v$ in states $\mathbb{C}_i, \mathbb{T}, \mathbb{L}$
Initialization: $state = \mathbb{C}_1; t_c = m_c = l_c = s_v = 0; r_c = 1;$
$p_b = \frac{1}{\eta\mathcal{F}}; p_s = \frac{\lambda(\mathcal{F}-1)}{\Delta}; T = \psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta\log^2 n}{\mathcal{P}} + \log\Delta\log n);$

upon entering state $\mathbb{C}_i$:
for $\gamma\log n$ timeslots do
    listen on $i_d$;
    if received $\mathcal{M}_B^i$ then
        if $i < \frac{\mathcal{F}-1}{2}$ then $state = \mathbb{C}_{i+1}$;
        else $state = \mathbb{T}$;
execute MIS algorithm in [23] on $i_d$;
if elected to the MIS then $state = \mathbb{B}_i$;
else
    if $i < \frac{\mathcal{F}-1}{2}$ then $state = \mathbb{C}_{i+1}$;
    else $state = \mathbb{T}$;

upon entering state $\mathbb{T}$:
$s = 0, 1$ with equal probability;
if $s = 0$ then listen on $\mathcal{F}$;
else
    select from $\{1_p, 2_p, \ldots, (\frac{\mathcal{F}-1}{2})_p\}$ uniformly at
    random, transmit on the selected channel with
    probability $p_s$ and listen otherwise;
    if received a message then $m_c = m_c + 1$;
    if transmitted a message then $t_c = t_c + 1$;
    if $t_c = \theta\log n$ then $state = \mathbb{L}$;
    if $r_c = \gamma\log n$ then
        if $m_c < 16\log n$ then $p_s = \max\{2p_s, 1/4\}$;
        else $p_s = p_s/2$;
        $m_c = 0; r_c = 0$;
$r_c = r_c + 1$;

upon entering state $\mathbb{L}$:
listen on $\mathcal{F}$;
```

IV. Analysis of Algorithm MIE

In this section, we show that with high probability, each node's packet can be sent to all its R-neighbors after executing the algorithm MIE for $O((\frac{\Delta}{\mathcal{F}} + \frac{\Delta\log n}{\mathcal{P}})\log n + \log\Delta\log n)$ timeslots. In the following, we use D_v and E_v^r to denote the disk centered at a node v with radii $R/2$ and r, respectively. The analysis consists of three steps: we first show that each node takes at most $O(\frac{\Delta\log n}{\mathcal{F}})$ timeslots in states $\{\mathbb{C}_i\}$ before joining state $\mathbb{T}$ or a state $\mathbb{B}_i$ for some i with $1 \leq i \leq \frac{\mathcal{F}-1}{2}$ (Lemma 4), and then prove that each submitter in state $\mathbb{T}$ can send its packet to an R-neighboring broadcaster in $O(\frac{\Delta\log n}{\mathcal{F}} + \log\Delta\log n)$ timeslots (Lemma 7); finally, we show that each packet can be disseminated by a broadcaster that has received the packet for $O(\frac{\Delta}{\mathcal{F}} + \frac{\Delta\log^2 n}{\mathcal{P}} + \log\Delta\log n)$ timeslots (Lemma 12). Combining everything, the main result can be proved.

```
Algorithm 2: MIE: a node $v$ in state $\mathbb{B}_i$
upon entering state $\mathbb{B}_i$:
select from $\{i_d, i_p, \mathcal{F}\}$ uniformly at random;
if selected $i_d$ then
    transmit $\mathcal{M}_B^i$ on $i_d$ with probability $1/\rho$;
else if selected $i_p$ then
    listen on $i_p$ and store the received packets into $B_v$;
else
    if $s_v = 0$ then
        $l_c = l_c + 1$;
        if $|B_v| \geq \mathcal{P}$ or $l_c = T$ then
            $s_v = 1$; $l_c = 0$; $p_b = \frac{1}{\eta\mathcal{F}}$;
    else
        transmit on $\mathcal{F}$ with probability $p_b$ and listen
        otherwise;
        if transmitted a message then $t_c = t_c + 1$;
        if received a message then $m_c = m_c + 1$;
        if $t_c = \theta\log n$ then
            discard the current message;
            if $|B_v| \geq \mathcal{P}$ then
                insert the first $\mathcal{P}$ packets into message;
            else $s_v = 0$;
        if $r_c = \gamma\log n$ then
            if $m_c < 16\log n$ then $p_b = \max\{2p_b, 1/4\}$;
            else $p_b = p_b/2$;
            $m_c = 0; r_c = 0$;
```

Before the analysis, we first present some lemmas and properties that will be used in the subsequent analysis. The following Lemma 1 is proved in [9].

Lemma 1 ([9]): Consider two disks D_1 and D_2 of radii R_1 and R_2, respectively, $R_1 > R_2$, we define χ^{R_1,R_2} to be the smallest number of disks D_2 needed to cover the larger disk D_1. It holds that $\chi^{R_1,R_2} \leq \frac{2\pi}{3\sqrt{3}} \cdot \frac{(R_1+2R_2)^2}{R_2^2}$.

The following Lemma 2 and Property 1 are proved in [23].

Lemma 2 ([23]): With probability $1 - O(n^{-3})$, every node v decides whether it should join the computed independent set after executing the MIS algorithm for at most $O(\log^2 n)$ timeslots. Furthermore, with probability at least $1 - O(n^{-3})$, in any timeslot t, the independent set computed by the MIS algorithm is correct.

Property 1 ([23]): For any disk D_v and in any timeslot t throughout the execution of the algorithm, the sum of transmission probabilities of nodes that are executing the MIS algorithm is at most $3 \cdot 2^{-\omega}$, where $\omega = 6.4$.

Now the analysis begins. We first show a property of the broadcasters in each state $\mathbb{B}_i$. The proof of Lemma 3 can be found in the technical report [24].

Lemma 3: In any timeslot t during the algorithm execution, with probability $1 - n^{-1}$, the broadcasters in each state $\mathbb{B}_i$ constitute an R-independent set.

We next consider the time each node takes in states $\{\mathbb{C}_i\}$. By Lemma 2 and the algorithm, a node v takes at most $O(\log^2 n)$ timeslots in each state $\mathbb{C}_i$ and there are totally $\frac{\mathcal{F}-1}{2}$

states in $\{\mathbb{C}_i\}$. This results in the following Lemma 4.

Lemma 4: For each node v, with probability $1-n^{-1}$, v joins state $\mathbb{T}$ or $\mathbb{B}_i$ for some i with $1 \leq i \leq \frac{\mathcal{F}-1}{2}$ after executing the algorithm for $O(\mathcal{F}\log^2 n) \in O(\frac{\Delta \log n}{\mathcal{F}})$ timeslots.

By the algorithm, each node leaves state $\mathbb{C}_i$ only if there is an R-neighboring node joining state $\mathbb{B}_i$ or when it receives a message $\mathcal{M}_B^i$ from an R-neighbor in state $\mathbb{B}_i$. Thus, a node finally joins state $\mathbb{T}$ only if it has R-neighbors in each state $\mathbb{B}_i$ for $1 \leq i \leq \frac{\mathcal{F}-1}{2}$. Then we have the following property.

Property 2: For each node v in state $\mathbb{T}$ and each state $\mathbb{B}_i$ for $1 \leq i \leq \frac{\mathcal{F}-1}{2}$, there exists a node u in state $\mathbb{B}_i$ such that $d(u, v) \leq R$.

We next consider how long a submitter in state $\mathbb{T}$ takes to send its packet to R-neighboring broadcasters on a channel i_p with $1 \leq i \leq \frac{\mathcal{F}-1}{2}$. Subsequently, we first show that $\theta \log n$ transmissions can ensure that a submitter transmits its packet to an R-neighboring broadcaster with high probability, and then bound the time a submitter takes for performing $\theta \log n$ transmissions.

We next present an upper bound on the sum of the transmission probabilities of submitters in a disk with radius $R/2$, which is crucial in the remaining analysis. The proof of Property 3 can be found in the technical report [24]. Denote by $\mathbb{T}_v$ the set of submitters in D_v, and $P_{\mathbb{T}_v}$ the sum of the transmission probabilities of nodes in $\mathbb{T}_v$.

Property 3: For each node v, during the execution of the algorithm, $P_{\mathbb{T}_v} \leq \lambda(\mathcal{F}-1)$.

Lemma 5: For each submitter v in state $\mathbb{T}$, after transmitting for $\theta \log n$ times, with probability $1 - O(n^{-1})$, it can send its packet to an R-neighboring broadcaster on a primal channel.

Proof: The proof is based on the following Claim 1, which is proved using a hierarchical method that bounds the expected interference at the receivers based on Property 3. The detailed proof can be found in [24].

Claim 1: In a timeslot, if a submitter v is the only transmitting node within the range $2R$ on a primal channel, with probability $\frac{1}{2}$, it can send its message to all R-neighboring nodes that are on the same channel.

We next bound the probability P_{only} that v is the only transmitting node in a timeslot when v transmits. By Lemma 1, E_v^{2R} can be covered by at most 44 disks with radius $R/2$. Then by Property 3, the sum of transmission probability of submitters in E_v^{2R} is at most $44\lambda(\mathcal{F}-1)$. Based on this,

$$P_{only} = \prod_{w \in E_v^{2R} \setminus \{v\}} (1 - p_s^w \cdot \frac{1}{\frac{\mathcal{F}-1}{2}}) \geq 4^{-\frac{2}{\mathcal{F}-1}\sum_{w\in E_v^{2R}} p_s^w}$$
$$\geq 4^{-88\lambda}$$

Then by Claim 1, in a timeslot, if v transmits on a primal channel, with probability $\frac{1}{2} \cdot 4^{-88\lambda}$, v can successfully send its packet to all nodes that operate on the same channel. Because in each timeslot, a broadcaster in a state $\mathbb{B}_i$ operates on the primal channel i_p with probability 1/3, after $\theta \log n$ transmissions, the probability that v still has not sent its packet to an R-neighboring broadcaster is at most $(1 - \frac{1}{3} \cdot \frac{1}{2} \cdot 4^{-88\lambda})^{\theta \log n} \leq n^{-2}$ if $\theta \geq 12 \cdot 4^{-88\lambda}$. By the union bound, the lemma is proved. ■

We still need to bound the time for a submitter v in state $\mathbb{T}$ which transmits for $\theta \log n$ times. Here we consider the case that v transmits less than $\theta \log n$ times before the transmission probability of v increases to 1/4. Clearly, the time needed in other cases is upper bounded by this case.

For a submitter v in state $\mathbb{T}$, we divide the time into intervals $\{T_j\}$, each of which contains $\gamma \log n$ timeslots. For each interval T_j, we call it an increasing interval if the transmission probability of v is doubled in the last timeslot of T, and a decreasing one otherwise. Let $\mathbb{E}_v$ denote the set of R-neighboring submitters of v that are in state $\mathbb{T}$. We first give a lower bound for the number of transmissions performed by R-neighboring broadcasters during a decreasing interval. For the proof of Lemma 6, please refer to [24].

Lemma 6: For a decreasing interval T_j, submitters in $\mathbb{E}_v$ transmits for $\Omega(\mathcal{F}\log n)$ times during T_j with probability $1 - n^{-3}$.

Lemma 7: For each node v, after joining state $\mathbb{T}$ for $O(\frac{\Delta \log n}{\mathcal{F}} + \log \Delta \log n)$ timeslots, it will send its packet to at least one R-neighboring broadcaster with probability $1 - O(n^{-1})$.

Proof: Let c_d be the number of decreasing intervals for v. We consider the number of intervals needed for v to increase its transmission probability to 1/4. Because each submitter in $\mathbb{E}_v$ transmit for $\theta \log n$ times, submitters in $\mathbb{E}_v$ transmit for $O(\Delta \log n)$ times in total. Therefore, by Lemma 6, $c_d \in O(\frac{\Delta \log n}{\mathcal{F} \log n}) = O(\frac{\Delta}{\mathcal{F}})$ with probability $1 - n^{-2}$. Thus, after at most $2c_d + O(\log \Delta) \in O(\frac{\Delta}{\mathcal{F}} + \log \Delta)$ intervals, with probability $1 - n^{-2}$, there are enough increasing intervals such that the transmission probability of v is increased to $1/4$.

We next bound the time needed for v to transmit for $\theta \log n$ times after the transmission probability is increased to 1/4. Let T_s be the first interval in which the transmission probability of v is 1/4. During the interval, v transmits for $\gamma \log n/4$ times in expectation. By the Chernoff bound, with probability $1 - n^{-2}$, v transmits for $\theta \log n$ times if γ is a large enough constant.

Putting everything together and by Lemma 5, v can send its packet to at least one R-neighboring broadcaster after at most $O(\frac{\Delta \log n}{\mathcal{F}} + \log \Delta \log n)$ timeslots with probability $1 - O(n^{-2})$. This is true for all nodes with probability $1 - O(n^{-1})$. ■

We next consider how long it takes to disseminate a packet $\mathscr{P}$ after $\mathscr{P}$ is received by a broadcaster v. This is done in two steps: we first show that $\theta \log n$ transmissions can ensure that a packet will be disseminated successfully with high probability, and then we bound the time for $\mathscr{P}$ to be transmitted by v for $\theta \log n$ times.

In the following lemma, we first bound the number of broadcasters that are within distance R from a node. Lemma 8 is proved based on the independence of broadcasters in each state $\mathbb{B}_i$. For the detailed proof, please refer to [24].

Lemma 8: For each node u, the number of its R-neighboring broadcasters in each state $\mathbb{B}_i$ with $1 \leq i \leq \frac{\mathcal{F}-1}{2}$ is at most 6, and the total number of its R-neighboring broadcasters is at most $3(\mathcal{F}-1)$.

For a node v, let $P_{\mathbb{B}_v}$ be the sum of transmission probability of broadcasters in E_v^R. We can get an upper bound on $P_{\mathbb{B}_v}$

using a similar argument as that in the proof of Property 3. The detailed proof can be found in [24].

Lemma 9: In each timeslot during the algorithm execution, for each node v, $P_{\mathbb{B}_v} \le \frac{14}{\eta}$ with probability $1 - O(n^{-1})$.

Based on the above Lemma 9, we now show that $\theta \log n$ transmissions ensure a successful dissemination for each broadcaster. The proof of Lemma 9 is similar to that for Lemma 5. For the detailed proof, please refer to [24].

Lemma 10: With probability $1 - n^{-1}$, each broadcaster will send each stored packet to all its $2R$-neighbors after transmitting the packet for $\theta \log n$ times.

For a packet $\mathscr{P}$, we next bound the duration from the time when $\mathscr{P}$ is received by a broadcaster v until $\mathscr{P}$ is transmitted for $\theta \log n$ times by v.

Lemma 11: With probability $1 - O(n^{-3})$, a broadcaster v transmits a message containing packet $\mathscr{P}$ for $\theta \log n$ times after v receives $\mathscr{P}$ for $3\psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)$ timeslots.

Proof: Denote by t_p the timeslot in which $\mathscr{P}$ is sent to v and let t_s be the timeslot in which v starts transmitting $\mathscr{P}$. We next show that v transmits $\mathscr{P}$ for $\theta \log n$ times before $t_s + \psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)$, and then $t_s - t_p \le 2\psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)$. Putting everything together, the lemma is proved.

Claim 2: With probability $1 - n^{-3}$, v will have transmitted $\mathscr{P}$ for $\theta \log n$ times before the timeslot $t_s + \psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)$.

Proof: Let T_s be the interval of timeslots during which the message containing $\mathscr{P}$ is transmitted by v. Divide T_s into intervals $\{I_i\}$ of lengths $\gamma \log n$. For an interval I_i, we call it an increasing interval if v increases the transmission probability at the end of I_i, and a decreasing interval otherwise. Let c_i and c_d be the number of increasing intervals and decreasing intervals in $\{I_i\}$, respectively.

We first claim that $c_i - c_d \le \log \mathcal{F} + \log \eta$ with probability $1 - n^{-3}$. Otherwise, because the transmission probability is decreased for c_d times and increased for c_i times, it is easy to show that there are at least two intervals I_{j_1} and I_{j_2} in which the transmission probability of v is 1/4. Without loss of generality, assume that $j_1 < j_2$. In I_{j_1}, v transmits for $\gamma \log n / 12$ times. Then if γ is sufficiently large, by the Chernoff bound, we can get that v transmits for $\theta \log n$ times during I_{j_1} with probability $1 - n^{-3}$. This means that v has stopped transmitting packet $\mathscr{P}$ before I_{j_1} ends, which contradicts the existence of I_{j_2}.

Now we consider the interval $T'_s = [t_s, t_s + \psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n))$, and show that $T_s \subset T'_s$. Similarly, we divide T'_s into intervals $\{I'_i\}$ of lengths $\gamma \log n$. Let c'_i and c'_d be the number of increasing intervals and decreasing intervals in $\{I'_i\}$, respectively. We next show that in T'_s, $c'_i - c'_d > \log \mathcal{F} + \log \eta$. As shown above, this means that v has transmitted the message containing packet $\mathscr{P}$ for $\theta \log n$ times before the end of T'_s, and the claim is proved.

We first bound the number of messages v receives from its $2R$-neighboring broadcasters during T'_s. In Lemma 8, we have shown that for a submitter, the number of R-neighboring broadcasters in a state $\mathbb{B}_i$ is at most 6. Thus, for each submitter w, its packet can be sent to at most $O(\log n)$ broadcasters. Furthermore, note that the broadcasters in E_v^{2R} receive packets from submitters in E_v^{3R}. By Lemma 1, the number of submitters in E_v^{3R} is $O(\Delta)$. Thus, the number of packets that the broadcasters in E_v^{2R} receive from submitters is at most $O(\Delta \log n)$. Furthermore, note that there are at most $O(\mathcal{F})$ broadcasters in E_v^{2R} by Lemma 8 and Lemma 1. So the total number of packets that the broadcasters in E_v^{2R} need to disseminate is at most $O(\Delta \log n + \mathcal{F}) \in O(\Delta \log n)$. By the algorithm, each broadcaster transmits for at most one message with less than $\mathcal{P}$ packets during T'_s, and each message is transmitted for $\theta \log n$ times. Thus, the total number of messages transmitted by the broadcasters in E_v^{2R} is at most $O(\frac{\Delta \log^2 n}{\mathcal{P}} + \mathcal{F} \log n) \in O(\frac{\Delta \log^2 n}{\mathcal{P}} + \frac{\Delta}{\mathcal{F}})$.

By the algorithm, in each interval I'_i that is decreasing, v receives at least $16 \log n$ messages from its $2R$-neighboring broadcasters. Thus, $c'_d \le \frac{O(\frac{\Delta \log^2 n}{\mathcal{P}} + \frac{\Delta}{\mathcal{F}})}{16 \log n} \in O(\frac{\Delta \log n}{\mathcal{P}} + \frac{\Delta}{\mathcal{F} \log n})$. Clearly, if ψ is large enough, we can get that $c'_i - c'_d > \log \mathcal{F} + \log \eta$, which completes the proof. ■

Claim 3: With probability $1 - O(n^{-3})$, v will have started transmitting $\mathscr{P}$ by the timeslot $t_p + 2\psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)$.

Proof: Let l_p be the number of packets that are stored at v and need to be transmitted before $\mathscr{P}$ in timeslot t_p. We prove the claim in two cases.

Case 1. $l_p < \mathcal{P}$.

In this case, if v does not transmit in timeslot t_p, then by the algorithm, v starts transmitting a message containing $\mathscr{P}$ after receiving $\mathscr{P}$ for at most $\psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)$ timeslots, which proves the claim. So we can assume that when $\mathscr{P}$ arrives at v in timeslot t_p, v is in the process of transmitting a message. By Claim 2, v must have stopped transmitting the message by $t' \le t_p + \psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)$ with probability $1 - n^{-3}$. And then by the algorithm, if the number of packets is larger than $\mathcal{P}$ in t', v will start transmitting a message containing $\mathscr{P}$ in timeslot $t' + 1$. Otherwise, v waits for at most $\psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)$ timeslots and then transmits a message containing $\mathscr{P}$. Thus, by the timeslot $t_p + \psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)$, v will have started transmitting a message containing $\mathscr{P}$ with probability $1 - O(n^{-3})$.

Case 2. $l_p \ge \mathcal{P}$.

Let l'_p be the number of packets that are stored at v in t_p and in messages transmitted before v transmits a message containing $\mathscr{P}$. We have $l'_p \le l_p \le \Delta \cdot \theta \log n$, and each of these messages contains $\mathcal{P}$ packets. Let T_l be the interval in which v transmits these l'_p packets. We consider the interval $T'_l = (t_s, t_s + \psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)]$ and show that $T_l \subset T'_l$. Then by the algorithm, after v completes transmitting these l'_p packets and waits for at most $\psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)$ timeslots, v starts transmitting a message containing $\mathscr{P}$. Putting everything together, we get that $t_s \le t_p + 2\psi(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)$.

Divide T'_l into intervals $\{I^l_i\}$ of lengths $\gamma \log n$, and let c^l_i

and c_d^l be the number of increasing intervals and decreasing intervals in $\{I_i^l\}$, respectively. Similar to the proof of Claim 2, we can show that $c_d^l \in O(\frac{\Delta \log n}{\mathcal{P}} + \frac{\Delta}{\mathcal{F} \log n})$. Thus, if ψ is large enough, $c_i^l - c_d^l \geq \log \mathcal{F} + \log \eta + c' \cdot \frac{\Delta \log n}{\mathcal{P}}$, where $c' \cdot \frac{\Delta \log n}{\mathcal{P}} \geq \frac{l'_p}{\mathcal{P}}$. This means that in at least $c' \cdot \frac{\Delta \log n}{\mathcal{P}}$ intervals I_i^l, the transmission probability is $1/4$. As shown in Claim 2, in each of these intervals, v transmits for $\theta \log n$ times with probability $1 - n^{-4}$. So in these intervals, v can transmit for $\frac{l'_p}{\mathcal{P}} \cdot \theta \log n$ times with probability $1 - n^{-3}$, and complete the transmission of all l'_p packets, since each message contains $\mathcal{P}$ packets and each message is transmitted for $\theta \log n$ times. This means that v has transmitted all l'_p packets before T'_l ends. In other words, $T_l \subset T'_l$.

Combining the above two cases proves Claim 3. ■

■

Note that broadcasters need to disseminate at most $O(n \log n)$ packets. Then based on Lemma 10 and Lemma 11, and by the union bound on the error probability, we can get the following result.

Lemma 12: For each packet $\mathscr{P}$, with probability $1 - O(n^{-1})$, a broadcaster v will send $\mathscr{P}$ to all its $2R$-neighbors after $\mathscr{P}$ is sent to v for $O(\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log^2 n}{\mathcal{P}} + \log \Delta \log n)$ timeslots.

Theorem 1: With probability $1 - O(n^{-1})$, each node's packet will be sent to its R-neighbors after it executes the algorithm for $O((\frac{\Delta}{\mathcal{F}} + \frac{\Delta \log n}{\mathcal{P}}) \log n + \log \Delta \log n)$ timeslots.

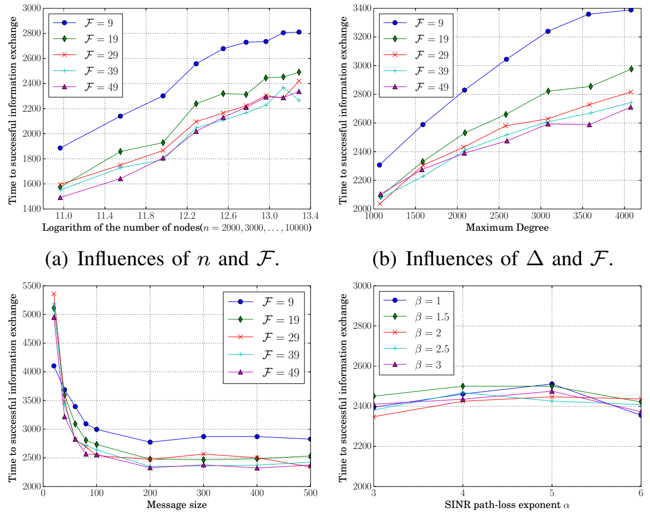
Proof: The running time of the algorithm can be obtained by Lemma 4, Lemma 7 and Lemma 12.

If a node u is a broadcaster, its packet can be sent to all R-neighbors in the time as stated in Lemma 12. If u is a submitter, by Lemma 7, its packet can be sent to at least one R-neighboring broadcaster v on a primal channel. Then by Lemma 12, u's packet will be disseminated to v's $2R$-neighbors. Note that u's R-neighbors are all v's $2R$-neighbors. So u's packet can be sent to all its R-neighbors. Note that the above analysis is based on the correctness of the MIS algorithm and the independence of broadcasters in each state $\mathbb{B}_i$. Then considering the error probability of Lemma 2 and Lemma 3, the result is proved. ■

V. Simulation Results

We study the empirical performance of our distributed algorithm here. We investigate (i) the impacts of different parameters in the model, (ii) the influences of the network topology, and (iii) the speedup due to multiple channels by comparison with other information exchange algorithms using only one single channel. In our experiments, n nodes are distributed in a square area of 10000×10000. The wake-up time of each node is randomly generated, and the measure of the algorithm performance is the maximum running time over all nodes.

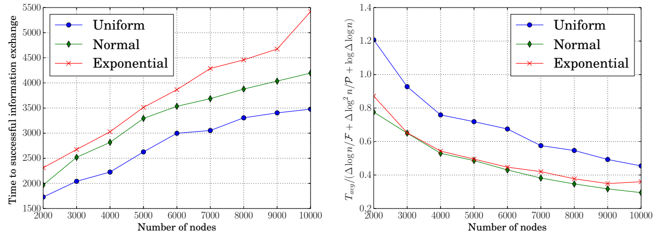
Consider Figure 1 where n nodes are *uniformly* and randomly distributed. In Figure 1(a), we observe that the execution time is proportional to $\log n$. Figure 1(b) clearly demonstrates the linear relationship between Δ and the number of timeslots needed. As shown in the analysis, the running



(a) Influences of n and $\mathcal{F}$. (b) Influences of Δ and $\mathcal{F}$.

(c) Influences of $\mathcal{P}$ and $\mathcal{F}$. (d) Influences of α and β.

Fig. 1. Influences of different parameters. The setting: $n = 10000$, average degree $\Delta_{avg} = 2000, \mathcal{F} = 39, \mathcal{P} = 1000, \alpha = 3, \beta = 1$, unless specified otherwise.

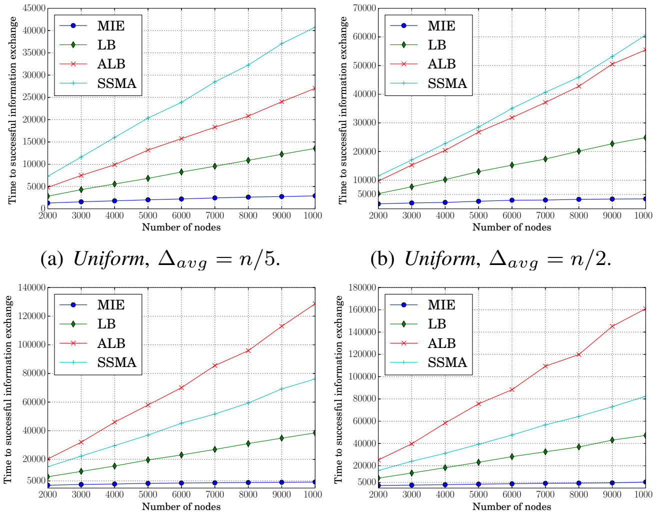


(a) The maximum of all nodes' running times. (b) Constants hidden in the Big-O notation.

Fig. 2. Profile of our distributed algorithm in terms of different node distributions, where $R = 10000\sqrt{1/2\pi}, \mathcal{F} = 9, \mathcal{P} = 1000, \alpha = 3, \beta = 1$.

time of the algorithm is no longer determined by $\mathcal{P}$ if $\Delta \log n/\mathcal{F} > \Delta \log^2 n/\mathcal{P} \Rightarrow \mathcal{P} > \mathcal{F} \log n$. Indeed, it can be seen in Figure 1(c) that the running time drops sharply at first as $\mathcal{P}$ increases, but remains relatively flat afterwards. The above three figures also show that more channels can help accomplish information exchange more quickly. But these figures do not demonstrate an ideal speedup for different numbers of channels. This is due to the limited network size in the simulation. As we can see, $\mathcal{F} = 9$ channels are already enough to reduce the running time to a very low level, so there is little room for further runtime reduction by using more channels. Moreover, the plot in Figure 1(b) shows that the $\mathcal{F}$ value that minimizes the running time is close to the theoretically analyzed value of $\mathcal{F}^* = 2\sqrt{\Delta/\log n} + 1$ (e.g., $\mathcal{F}^* \approx 19$ and 23 when $\Delta \approx 1077.2$ and 1587.1 respectively). Figure 1(d) shows that our algorithm is insensitive to the SINR parameters α and β, which is certainly an advantage if we were to implement the algorithm in real networks. All these simulation results are in agreement with our theoretical results.

Besides the uniform distribution, we also investigated the performance of our algorithm under another two typical node distributions: *Normal* and *Exponential*, as shown in Figure 2(a). We observe that the time needed for successful



(a) *Uniform*, $\Delta_{avg} = n/5$.

(b) *Uniform*, $\Delta_{avg} = n/2$.

(c) *Normal*

(d) *Exponential*

Fig. 3. Performance of our distributed algorithm compared with the other three algorithms, where $\mathcal{F} = 9, \mathcal{P} = 1000, \alpha = 3, \beta = 1$.

information exchange is longer in the *Exponential* and *Normal* cases whose Δ's are much larger than that in the *Uniform* topology, indicating that it is Δ that dominates the time complexity of our algorithm when the other parameters are fixed. Figure 2(b) illustrates that the hidden constant in the Big-O notation is very small.

We have also conducted experiments to verify the $\Theta(\mathcal{F})$ speedup of our algorithm (*MIE*) by comparison with three existing single-channel algorithms: *LocalBroadcast1* (*LB*) [12], *Asynchronized Local Broadcasting Algorithm* (*ALB*) [22] and *Slow-Start Media Access* (*SSMA*) [9]. In Figure 3(a) and Figure 3(b), it can be seen that our algorithm performs better in dense networks, and achieves a 5 to 17 times speedup. More surprisingly, our algorithm achieves up to 30 times speedup in Figure 3(c) and Figure 3(d) as the performance of the other algorithms is badly undermined by the large Δ.

VI. Conclusion

We presented the first asynchronous distributed algorithm for information exchange in multi-channel wireless ad hoc networks under the physical SINR interference model. Using previous best results in single-channel networks as a contrast, our results show that the availability of multiple channels can indeed speed up information exchange, in fact substantially in some cases, as illustrated by the simulation results.

Given that our indirect strategy makes use of "long" messages (each can carry $\mathcal{P}$ packets), one may expect a possible speedup of more than $\mathcal{F}$ times. But because a node can receive at most $\mathcal{P}$ packets in a received message, it needs $\Omega(\Delta/\mathcal{P})$ timeslots to receive all packets of its neighbors. This lower bound says that no matter how many channels we have, the optimal speedup is $\mathcal{P}$ times. The running time of our algorithm when $\mathcal{P} < \mathcal{F} \log n$ is $O(\frac{\Delta \log^2 n}{\mathcal{P}})$, which is short of the optimal speedup by an $O(\log^2 n)$ term. Thus one future direction is to derive faster algorithms with a greater speedup when $\mathcal{P}$ is small. On the other hand, it is also unclear whether we can get a speedup greater than $\mathcal{F}$ under the SINR model when $\mathcal{P}$ is sufficiently large with respect to $\mathcal{F}$. Looking at the indirect approach we use, whether a better than $\mathcal{F}$ times speedup is possible seems to depend on whether it is possible to transmit to a particular neighbor as opposed to all neighbors. Some future study on this point may help derive a lower bound on the optimal speedup or a faster algorithm.

VII. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China Grants 61073174 and 61373027, Hong Kong RGC GRF Grant 714311 and the Shu Shengman Special Fund.

References

[1] Bluetooth Consortium. Bluetooth Specification Version 2.1. July 2007.

[2] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. On the complexity of radio communication. In *STOC'89*, 1989.

[3] S. Daum, M. Ghaffari, S. Gilbert, F. Kuhn, and C. Newport. Maximal independent sets in multichannel radio networks. In *PODC'13*, 2013.

[4] S. Daum, S. Gilbert, F. Kuhn, and C. Newport. Leader election in shared spectrum radio networks. In *PODC'12*, 2012.

[5] S. Daum, F. Kuhn, and C. Newport. Efficient symmetry breaking in multi-channel radio networks. In *DISC'12*, 2012.

[6] B. Derbel and E. Talbi. Distributed node coloring in the sinr model. In *ICDCS'10*, 2010.

[7] B. Derbel and E. Talbi. Radio network distributed algorithms in the unknown neighborhood model. In *ICDCN'10*, 2010.

[8] S. Dolev, S. Gilbert, M. Khabbazian, and C. Newport. Leveraging channel diversity to gain efficiency and robustness for wireless broadcast. In *DISC'11*, 2011.

[9] O. Goussevskaia, T. Moscibroda, and R. Wattenhofer. Local broadcasting in the physical interference model. In *DIALM-POMC'08*, 2008.

[10] P. Gupta, P. R. Kumar. The capacity of wireless networks. IEEE Transaction on Infromation Theorey, Vol. 46, No. 2, pp. 388-404, 2000.

[11] M. M. Halldórsson, S. Holzer, P. Mitra, and R. Wattenhofer. The power of non-uniform wireless power. In *SODA'13*, 2013.

[12] M. M. Halldórsson and P. Mitra. Towards tight bounds for local broadcasting. In *FOMC'12*, 2012.

[13] I802.11. Wireless lan mac and physical layer specifications. June 1999.

[14] T. Jurdzinski, D. R. Kowalski, M. Rozanski, and G. Stachowiak. On the Impact of Geometry on Ad Hoc Communication in Wireless Networks. In *PODC'14*, 2014.

[15] L. G. Roberts. Aloha Packet System with and without Slots and Capture. *ACM SIGCOMM, Computer Communication Review*, 5(2):28-42, 1975.

[16] C. Scheideler, A. Richa, and P. Santi. An $O(\log n)$ dominating set protocol for wireless ad-hoc networks under the physical interference model. In *MobiHoc'08*, 2008.

[17] Y. Wang, Y. Wang, D. Yu, J. Yu and F.C.M Lau. Information exchange with collision detection on multiple channels. *Journal of Combinatorial Optimization*, to appear.

[18] Y. Yan, D. Yu, Y. Wang, J. Yu, F.C.M Lau. Bounded information dissemination in multi-channel wireless networks. to appear.

[19] D. Yu, Q-S. Hua, Y. Wang, and F.C.M. Lau. An $O(\log n)$ distributed approximation algorithm for local broadcasting in unstructured wireless networks. In *DCOSS'12*, 2012.

[20] D. Yu, Q-S. Hua, Y. Wang, H. Tan, and F.C.M. Lau.Distributed multiple-message broadcast in wireless ad-hoc networks under the SINR model. In *SIROCCO'12*, 2012.

[21] D. Yu, Q-S. Hua, Y. Wang, J. Yu and F.C.M. Lau. Efficient distributed multiple-message broadcasting in unstructured wireless networks. In *INFOCOM'13*, 2013.

[22] D. Yu, Y. Wang, Q-S. Hua and F.C.M Lau. Distributed local broadcasting algorithms in the physical interference model. In *DCOSS'11*, 2011.

[23] D. Yu, Y. Wang, Q-S. Hua and F.C.M. Lau. Distributed $(\Delta+1)$-coloring in the physical model. In *ALGOSENSORS'11*, 2011.

[24] D. Yu, Y. Wang, Y. Yan, J. Yu and F.C.M. Lau. Speedup of Information Exchange using Multiple Channels in Wireless Ad Hoc Networks. *Technical Report*. https://www.dropbox.com/s/ln2i5mj8c35ba54/IE.pdf.