

Distributed Broadcasting in Dynamic Networks

Dongxiao Yu, *Member, IEEE*, Yifei Zou, Jiguo Yu, *Senior Member, IEEE*, Yu Wu, Weifeng Lv, Xiuzhen Cheng, *Fellow, IEEE*, Falko Dressler, *Fellow, IEEE*, Francis C.M. Lau



Abstract—In this paper, we investigate distributed broadcasting in dynamic networks, where the topology changes continually over time. We propose a network model that captures the dynamicity caused by both churn and mobility of nodes. In contrast to existing work on dynamic networks, our model defines the dynamicity in terms of localized topological changes in the vicinity of each node, rather than a global view of the whole network. Obviously, a local dynamic model suits distributed algorithms better than a global one. The proposed dynamic model uses the more realistic SINR model to depict wireless interference, instead of oversimplified graph-based models adopted in most existing work. We consider the fundamental communication primitive of global broadcast, which is to disseminate a message from a source node to the whole network. Specifically, we present a randomized distributed algorithm that can accomplish dynamic broadcasting in an asymptotically optimal running time of $O(D_T)$ with a high probability guarantee, under the assumption of reasonably constant *dynamicity rate*, where D_T is the *dynamic diameter*, a parameter proposed to depict the complexity of dynamic broadcasting. We believe our local dynamic model can greatly facilitate distributed algorithm studies in mobile and dynamic wireless networks.

Index Terms—Dynamic network; SINR model; distributed algorithm; global broadcast.

1 INTRODUCTION

WIRELESS networks are inherently challenged by dynamicity, especially with the advent of the Internet-of-Things and the ubiquity of mobile devices with communication capabilities. Dynamicity comes in many disguises, as nodes join, leave, and move around,

which causes network topology changes that are continuous and unpredictably over time.

Dynamic networks have gained much attention from different domains. Especially, as the network becomes large and decentralized, such as for typical applications in the Internet-of-Things, the study of dynamic networks has become popular in the distributed computing domain. Distributed solutions can inherently tolerate dynamicity in some sense. In distributed algorithms, nodes only communicate with their neighbors. If network changes happen in some region of the network, only a local part of the whole network is affected. The whole network could function correctly from a global perspective.

Traditionally, studies in distributed algorithms focus on stabilizing algorithms that can recover after a network change. However, the use of such algorithms in realistic scenarios may be limited, as networks can change continually, and it is impossible to expect the network eventually stops changing. More recent studies, therefore, turn attention to study computation algorithms whose correctness and termination even hold in networks that change continuously. Various dynamic network models have been proposed, such as the unstructured model [24], the dual graph model [7], [21], and the T -interval connectivity model [23]. However, all these models define the dynamicity from a global view. In other words, these models define the network change in terms of the whole network, rather than the network change in the vicinity of each node. Though this definition is very helpful for analyzing the complexity of problem solving, however, on the one hand, as distributed algorithms rely on local coordination and local communications when considering the communications between a particular node with its neighbors, the worst case of the network change in its local vicinity mapped from the global network change has to be considered, which usually results in an overpessimistic estimation of local network changes. Consequently, it is impossible to devise efficient algorithms even if the local network change is marginal. On the other hand, defining global network change cannot fully make use of the power of distributed algorithms in tolerating dynamicity. Hence, local dynamic models, which describe the network change in the vicinity of nodes, are more suitable for distributed

- D. Yu and X. Cheng are with School of Computer Science and Technology, Shandong University, Qingdao, 266510, P.R. China. E-mail: dxyu@sdu.edu.cn, xzcheng@sdu.edu.cn.
Y. Zou (Corresponding Author) and F. C.M. Lau are with the Department of Computer Science, The University of Hong Kong, Hong Kong, P.R. China. E-mail: yfzou,fcmlau@cs.hku.hk.
- J. Yu is with Qilu University of Technology (Shandong Academy of Sciences), Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan, Shandong, 250014, P. R. China, and School of Information Science and Engineering, Qufu Normal University, Rizhao, Shandong, 276826, P. R. China. E-mail: jiguo@qnu.edu.cn.
- Y. Wu is with School of Computer Science and Network Security, Dongguan University of Technology, Dongguan, P.R. China. E-mail: wuyu@dgut.edu.cn.
- W. Lv is with School of Computer Science and Engineering, Beihang University, Beijing, 100191. E-mail: lwf@buaa.edu.cn.
- F. Dressler is with the Heinz Nixdorf Institute and the Dept. of Computer Science, Paderborn University, Paderborn, 33102, Germany. E-mail: dressler@ccs-labs.org.

algorithm design.

Furthermore, communications in wireless network suffer from interference and collisions. Most previous works on distributed algorithm study adopt graph-based models, in which the interference is defined in a binary and local way. However, such a definition manner neglects interference from far-away nodes, as well as the cumulation and fading features of wireless interference, which may make the designed algorithms perform significantly different from analysis. Recently, a more realistic model, the Signal-to-Interference-plus-Noise (SINR) model, becomes popular in the algorithmic domain due to its accuracy in reflecting crucial features of wireless interference, despite it is challenging for distributed algorithm design and analysis due to the global definition approach of interference.

In this work, we propose a local dynamic model that uses the SINR model to depict wireless interference. It must be noted that the SINR model defines the signal (as well as interference) fades with the distance according to some path-loss exponent. Hence, when depicting the local network change, it not only needs to reflect the change on the neighborhood of each particular node, but also to describe the distance change between the node and its neighbors. In addition, it is better to define the local network topology change between nodes as independent as possible, for the convenient of analyzing the algorithm performance.

The local dynamic network model we suggest obeys the above considerations. In particular, the network area is divided into a grid, ensuring any pair of nodes in each cell of which are neighbors. Then, in each cell, the local topology of the nodes is defined based on the distance of each node with its nearest neighbor in the same cell. The impact of dynamicity is then depicted by the magnitude of changing on the local topology of each cell that is caused by churn (node insertion/deletion) and mobility of nodes.

Under the proposed dynamic model, we study the possibility of devising efficient algorithms for basic communication primitives. In particular, we focus on the problem of network-wide message broadcast, which requires to disseminate a message \mathcal{M}_s of a source node s to the whole network. Broadcast is a fundamental primitive. It can be used to simulate a single-hop network on top of a multi-hop network, to greatly simplify the design and analysis of higher-level algorithms. Furthermore, it has been shown that unicast communication has substantial problems in highly dynamic networks [5].

Comparison with existing results. To the best of our knowledge, the dynamic global broadcast (DGB) results most relevant to our work are proposed in [1], [6], [22], which require $O((1 + \frac{n}{k * \min\{T, \tau\}}) \cdot n \log^3 n)$, $O(n^2 / \log n)$, and $O(n \log^2 n)$ time steps¹ respectively to complete the DGB task based on their own dynamic models. However,

it deserves to noting that all the protocols mentioned above are designed in the graph-based model, which is not as realistic as the SINR model adopted in our work. Also, the dynamic models in [1], [6], [22] mainly focus on the dynamicity of links in the network, but they rarely consider the dynamicity of nodes. In our dynamic model, both the dynamicity from links and from nodes are depicted. Thus, our model is more comprehensive. Besides, our technique absolutely differs from the techniques in [1], [6], [22] for designing algorithms. Specifically, in [1], [6], [22], all nodes participate in the message dissemination. To avoid the flooding phenomenon, nodes in [1], [22] initially transmit the message with probability 1 and then gradually decrease the transmission probability. For the message dissemination process in [1], [22], each node receives the message only when its neighbors reach a proper contention by adjusting their transmission probability. Thus, in a dense network, it takes a long time for one-hop message dissemination in [1], [22]. In other words, the protocols in [1], [22] may have a trivial result in dense network. In contrary, by letting each node transmit with probability $\frac{\ln n}{n}$, the protocol in [6] suits dense networks well. However, in a sparse network, e.g., each node has only a constant number of neighbors, it takes $O(\frac{n}{\ln n})$ time steps to complete one-hop message dissemination in expectation. Differing from the previous protocols, our work disseminates the message by electing a leader in each local area and letting the leader transmit the message, which is insensitive to the density of the nodes. In detail, no matter the network is dense or sparse, it takes $O(\log n + \log R_c)$ time steps to guarantee a one-hop message dissemination with a high probability². Finally, considering a lower bound $O(D + \log n)$ for DGB in a static network with diameter D , it can be seen that there is a huge gap between the time complexity of current results in [1], [6], [22] and the lower bound $O(D + \log n)$. In this work, we present for a dynamic network, in the proposed comprehensive model, that it is the diameter of the dynamic network but no longer the number of nodes n that linearly impacts the time complexity of DGB, which represents a bigger step towards the optimal result than previous work.

Our main contributions can be summarized as follows.

- We propose a dynamic network model that is defined from a local view. The dynamicity rate is used to reflect the magnitude of changing on the local network topology around each node. The dynamic model we propose is general enough to model various dynamic networks. Specially, we are interested in mobile networks, where the nodes move around unpredictably.
- We present a randomized distributed algorithm for dynamic broadcasting. We focus on the hard case of non-spontaneous broadcasting, where a node can join the algorithm only after receiving the message.

1. a time step is the time unit for agents in network to transmit/receive one message

2. R_c is the transmission range and we say an event occurs with high probability if it occurs with probability $1 - n^{-c}$ for some constant $c > 1$

We use a parameter T to depict the time a link can keep stable, and propose a parameter, *dynamic diameter* D_T , to depict the complexity of dynamic broadcasting. Our algorithm ensures that under the reasonable constant dynamicity rate, broadcasting can be accomplished in $O(D_T)$ time with a high probability guarantee, i.e., with probability $1 - n^{-c}$ for some constant $c > 0$. The algorithm achieves an asymptotically optimal running time when $T \in \Omega(\log n)$, as D_T is a natural lower bound for dynamic broadcasting. Besides, constant dynamicity rate restricts the local network change in one round. The network topology can change significantly after only a few rounds.

Roadmap: The remainder of this paper is organized as follows. Sec. 2 outlines the related work and Sec. 3 introduces the network model. The dynamic broadcast algorithm and its performance analysis are presented in Sec. 4 and Sec. 5, respectively. Sec. 6 shows the simulation and the paper is concluded in Sec. 7.

2 RELATED WORK

Distributed algorithm design and analysis has been a hot topic in wireless distributed computing domain, due to the popularity of large-scale mobile wireless networks. Many dynamic models are proposed to reflect the dynamicity in wireless networks. In [24], Kuhn *et al.* proposed the unstructured model, to describe the nodes' insertions under unit disk setting. Later, this model was extended to SINR model and bounded independence graphs in [14] and [26] respectively. The node crash failures were considered in [4]. Other models mainly focus on modeling the impact of unreliable links, and assume the node set to be static. The dual graph model was introduced in [7], [21]. It defines two graphs on the same node set, one is composed by reliable links, and the other is composed by unreliable links. This model extends the radio network model to the dynamic case. The T -interval connectivity model given in [23] models dynamic networks in an adversarial manner, under the constraint that the network contains a stable connected spanning subgraph in every interval of T consecutive rounds. The pairing model, introduced in [8], [11] assumes that the links in the network constitute a matching in each round. Considering that the channel varies with time because of random fading, shadowing and node mobility, a simple ON-OFF channel model was introduced in [27]. In this model, the network-configuration follows a stationary ergodic process with the stationary distribution, and in each slot the network controller can only activate a set of non-interfering links. More recently, Yu *et al.* [32] proposed a dynamic model that admits both node and link changes, under the SINR model. However, this model is not a general one. So it cannot model various dynamic scenarios. A survey on dynamic network models is given in [25].

Broadcasting is one of the most extensively studied communication primitives. In static networks, many results have been proposed for broadcasting, in both graph-based models [2], [9], [13], [19], [20] and the SINR model [10], [17], [18], [28], [30]. For non-spontaneous broadcasting, under the graph-based model, the best randomized results are $O(D \log(n/D) + \log^2 n)$ [9], [20] and $O(D + \log^6 n)$ [13] without and with collision detection. Under the SINR model, the best known algorithm was given in [15], which can accomplish broadcast in $O(D \log^2 n)$ rounds.

As for broadcasting in dynamic networks, in [6], Clementi *et al.* presented a randomized algorithm that can solve the broadcast problem in $O(n^2/\log n)$ rounds, under a basic assumption that there exists at least one stable link between nodes with and without the message. In the dual graph model, an $O(n^{3/2}\sqrt{\log n})$ -time deterministic algorithm and an $O(n \log^2 n)$ -time randomized algorithm were given in [22]. In the T -interval connectivity model, Ahmadi *et al.* [1] showed that for any T and τ -oblivious adversary, broadcasting can be accomplished in $O((1 + \frac{n}{k * \min\{T, \tau\}}) \cdot n \log^3 n)$, where k is a connectivity requirement. However, all three works mentioned above are based on the graph-based model, which is not as realistic as the SINR model adopted in our work. Also, their dynamic models focus more on the reliability of links, but rarely consider the dynamicity of nodes, which is not as comprehensive as our model.

3 MODEL

We consider a network on a two-dimensional Euclidean space, where nodes can be placed arbitrarily, possibly in the worst case. n is a given upper bound on the number of nodes in the network. Denote by V the set of nodes in the network. For any two nodes u and v , let $d(u, v)$ be the Euclidean distance between u and v . Each node has a unique identifier ID_v . The nodes communicate via a shared channel. The practical uniform power assignment is assumed, i.e., all nodes have the same transmit power P .

Communication model It is assumed that nodes in the network operate synchronously in rounds. A round consists of a *constant* number of time units that are used for nodes to send a message, e.g., a multiple of the $50\mu s$ units in IEEE 802.11. Each node is equipped with a half-duplex transceiver, i.e., in each time unit, a node can transmit or listen but cannot do both. With this assumption, our algorithm could be implemented in either network with half-duplex or full-duplex transceiver.

Concurrent transmissions on the shared channel interfere with each other. We use SINR model to depict the interference. Specifically, in a round, let \mathcal{T} be the set of transmitting nodes. Consider a receiver v and a transmitter u . Let $I_u(v)$ be the interference v experienced in this round, with respect to the transmitter u . In the SINR model, $I_u(v)$ and SINR rate $SINR(v, u, \mathcal{T})$ are

defined as follows

$$\begin{aligned} I_u(v) &= \sum_{w \in \mathcal{T} \setminus \{u\}} P \cdot d(w, v)^{-\alpha}; \\ \text{SINR}(v, u, \mathcal{T}) &= \frac{P \cdot d(v, u)^{-\alpha}}{N + I_u(v)}. \end{aligned} \quad (1)$$

where the path-loss α is normally between 2 and 6 and the ambient noise $N > 0$. If $\text{SINR}(v, u, \mathcal{T}) \geq \beta$, then v can receive the message from u . The threshold β is determined by hardware and larger than 1.

The transmission range R_T of a node v is defined as the maximal distance at which node u can clearly receive a message from v when there are no other nodes transmitting simultaneously. By the condition that $\text{SINR}(u, v, \mathcal{T}) \geq \beta$, $R_T = (P/\beta N)^{1/\alpha}$. However, nodes which has a distance very close to R_T can only communicate when other transmitting nodes are sufficiently faraway. Then a standard assumption, like that in [3], [16], [31], [33], [34], is to define a *communication range* $R_c = (1 - 2\epsilon)R_T$, where the constant $\epsilon \in (0, 0.5)$ is a model parameter, to make sure the communication can tolerate some interference. To make the description and analysis brief, we normalize the smallest distance between nodes to equal to 1. Then, the transmission range R_c can be sufficiently large, which is usually polynomially bounded by n in reality.

We say two nodes are d -neighbors, if the distance between them is not larger than d . Two R_c -neighbors are simply said to be neighbors.

A set of nodes I is called a d -independent set if for any pair of nodes $u, v \in I$, $d(u, v) > d$. A set of nodes S is called a d -dominating set if for each node v , there exists a node $u \in S$ such that $d(u, v) \leq d$. A set of nodes M is called a d -maximal independent set if M is a d -independent set as well as a d -dominating set.

Dynamicity It is assumed that both churn and mobility of nodes may occur in the network. We define the dynamic behaviors in the network in a local view.

The 2-dimensional network area is divided into a grid \mathcal{G} , which consists of square cells of size $\frac{\epsilon R_c}{\sqrt{2}} \times \frac{\epsilon R_c}{\sqrt{2}}$. We assume that point $(0, 0)$ is the grid origin. Each cell includes its left side without the top endpoint, and its bottom side without the right endpoint, and does not include its right and top sides. A cell is given a coordinate axes (i, j) when its bottom left corner located at $(\frac{\epsilon R_c}{\sqrt{2}} * i, \frac{\epsilon R_c}{\sqrt{2}} * j)$ for $(i, j) \in \mathbb{Z}^2$, and is denoted as $g(i, j)$. For a node v locating at position (x, y) on the network, when $i * \frac{\epsilon R_c}{\sqrt{2}} \leq x < (i+1) \frac{\epsilon R_c}{\sqrt{2}}$ and $j * \frac{\epsilon R_c}{\sqrt{2}} \leq y < (j+1) \frac{\epsilon R_c}{\sqrt{2}}$, it has the grid coordinate $g(i, j)$.

As the signal fades with distance as defined in the SINR model, we give an approach to define the local network topology in a cell that can reflect the distance between nodes. Consider the network at a time point, divide the nodes in each cell g into classes $\{V_i^g : i = 0, 1, \dots, \log \epsilon R_c\}$. To be more detailed, for a cell g and a node $v \in g$, let u be v 's nearest neighbor in g if there are at least two nodes in g . v is in class V_i^g for $0 \leq i \leq$

$\log \epsilon R_c - 1$ if $d(u, v) \in [2^i, 2^{i+1})$. If v is the only node in cell g , v is in class $V_{\log \epsilon R_c}^g$.

Based on the local topology depiction given above, we now present our dynamic model. The model admits both node churns (node arrivals/departures) and node mobility (move from one cell to another cell). We assume that the network change occurs at the beginning of every round.

We define a *dynamicity rate* to measure the change of network topology. Consider a period of rounds I . For $i \in \{0, 1, \dots, \log \epsilon R_c\}$ and $t \in I$, let $V_i^g(t)$ and $\hat{V}_i^g(t)$ denote the set of active nodes³ in cell g at the beginning and the end of a round t respectively, and $n_i^g(t) = |V_i^g(t)|$, $\hat{n}_i^g(t) = |\hat{V}_i^g(t)|$. Then the *dynamicity rate* λ is defined as

$$\lambda = \max_{t \in I, g \in \mathcal{G}, 0 \leq i \leq \log \epsilon R_c} \{|n_i^g(t+1) - \hat{n}_i^g(t)| / \hat{n}_i^g(t)\}.$$

It can be seen that as the dynamicity rate changes, our dynamic model can model various dynamic networks. Furthermore, it is required that when a node joins a new cell, it needs to stay in the cell for $\Omega(\log n)$ rounds. This assumption is necessary, as successfully disseminating a message needs $\Omega(\log n)$ rounds to achieve a high probability guarantee [29].

Stable Diameter. To measure the complexity of broadcast algorithms, we need to depict the connectivity of the dynamic network. We here use a concept of stable path to depict the connectivity.

In particular, given a positive integer T , which is called the *stability parameter*, we define a T -stable path from node u to node v as follows: for a sequence $v_0 = u, v_1, \dots, v_k = v$, if there is a sequence I_0, I_1, \dots, I_{k-1} of time intervals with $I_i = [b_i, e_i]$, such that for each i , $e_i - b_i \geq T$, and $e_i - e_{i-1} \geq T$, nodes v_{i-1} and v_i keep active and being neighbors during I_{i-1} , then $v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_k$ is a T -stable path. The *length* of the T -stable path from u to v is then defined as $\cup_{i=0}^{k-1} |I_i|$. Each link on a T -stable path is called a *stable link*.

The stability parameter T depicts the time duration for two nodes to be connected. Larger T means the connection between two nodes can be stable for a longer time. We here consider the case that $T \in \Omega(\log n)$, since $\Omega(\log n)$ is the minimum time needed for two nodes to communicate successfully with high probability [29].

Given the stability parameter T , we define the stable T -distance $D_T(u, v)$ as the minimum length of T -stable paths between u, v . If there is not any T -stable path connecting u, v , then $D_T(u, v) = \infty$. The *T -stable diameter* of the network is then defined as $D_T = \max_{u, v \in V} D_T(u, v)$. If D_T is finite, then the network is called *T -stable connected*. Clearly, D_T is a natural lower bound for dynamic broadcasting.

Knowledge and Capability of node Each node has the values of n , R_c , N and an estimation on the SINR

3. Active nodes are those that participate in the algorithm execution. For non-spontaneous broadcasting, active nodes are those that have received the message of the source node.

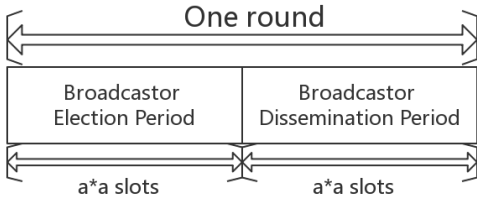


Fig. 1: Two periods in one round, each of which consists of $a * a$ slots.

parameters α, β .⁴ The nodes can acquire location information by some services, such as GPS. But physical carrier sensing is not needed, i.e., nodes cannot detect whether there are transmissions on the channel if it does not receive a message. Considering that it is very likely for some errors to occur in location estimation in reality, we account for these errors in our model. Without loss of generality, we assume that for any node, its estimated location is within $\frac{\epsilon}{2}R_c$ from its real location.

4 BROADCAST ALGORITHM

4.1 Algorithm Overview

As defined in the non-spontaneous broadcast problem, only nodes that possess the message become active. Basically, the algorithm disseminates the message by letting an active node locally broadcast the message to its neighboring inactive nodes. The difficulty in implementing the strategy is that if the active nodes are dense, there will be heavy contention in a local region, and plenty of collisions are caused, which hinders the message dissemination. Hence, we select a set of *broadcasters* from active nodes, and only let broadcasters disseminate the message instead, to accelerate the dissemination process. To make sure that for each inactive node, if it has an active neighbor, it can receive the message from a broadcaster, the broadcasters are selected ensuring:

- The broadcasters constitute a $\frac{3}{2}\epsilon R_c$ -dominating set in terms of active nodes, i.e., for each active node, there exist a broadcaster within distance $\frac{3}{2}\epsilon R_c$. In this case, for each inactive node with a neighboring active node, there exists at least one broadcaster within distance $(1 + \frac{3}{2}\epsilon)R_c$. Hence, if we can make the broadcasters disseminate the message within distance $(1 + \frac{3}{2}\epsilon)R_c$, the inactive node will get the message;
- The density of broadcasters is constant bounded, i.e., there are a constant number of broadcasters in the neighborhood of each node, such that the contention is bounded.

With the above strategy, the algorithm execution is divided into phases, each of which consists of $k * (\log n + \log R_c)$ rounds, where k is a constant given in the analysis. There are two periods in each round: the *broadcaster election period* and the *broadcaster dissemination period*,

Algorithm 1: Broadcast Algorithm for a node v with color i

Initialization:

```

1 if  $v$  is the source node then
2    $\lfloor state_v = \mathbb{A};$ 
3 else
4    $\lfloor state_v = \mathbb{I};$ 


---


In each phase,  $v$  does:
5 if  $v$  has the message  $\mathcal{M}_s$  then
6    $\lfloor state_v = \mathbb{A};$ 
7 else
8    $\lfloor state_v = \mathbb{I};$ 
9  $round_1 = 0;$ 
10 for  $k * (\log n + \log R_c)$  rounds do
11    $slot_1 = slot_2 = 0;$ 
12   for  $slot_1 < a * a$  do
13     Broadcaster-election-period( $v, i,$ 
14        $slot_1, round_1$ );
15      $slot_1++;$ 
16   for  $slot_2 < a * a$  do
17     Broadcaster-dissemination-period( $v, i,$ 
18        $slot_2$ );
19      $slot_2++;$ 
20    $round_1++;$ 

```

Broadcaster-election-period($v, i, slot_1, round_1$):

```

19 if  $state_v = \mathbb{A}$  and  $slot_1 = i$  then
20   transmit a message  $\mathcal{M}$  with probability  $p$ ;
21   if received a message  $\mathcal{M}$  from nodes in the same
22     cell then
23      $\lfloor state_v = \mathbb{S};$ 
24 else
25    $\lfloor$  keep silent;
26 if  $round_1 = k * (\log n + \log R_c) - 1$  then
27   if  $state_v = \mathbb{A}$  then
28      $\lfloor state_v = \mathbb{B};$ 

```

Broadcaster-dissemination-period($v, i, slot_2$):

```

28 if  $state_v = \mathbb{B}$  and  $slot_2 = i$  then
29   transmit a message  $\mathcal{M}_s$ ;
30 else
31    $\lfloor$  listen;

```

4. $\alpha_{min} < \alpha < \alpha_{max}, \beta_{min} < \beta < \beta_{max}$, and nodes only need to know $\alpha_{min}, \alpha_{max}, \beta_{min},$ and β_{max} .

all of which consist of constant slots. The periods in one round are illustrated in Fig 1. In the broadcaster election period, active nodes elect broadcasters in each non-empty cell as desired; and in the broadcaster dissemination period, the broadcasters locally broadcast the message of the source node to $(1 + \frac{3\epsilon}{2})R_c$ -neighbors.

To implement the broadcast strategy, in the broadcaster election period, besides the difficulties in interference and collision control posed by the global SINR model, we also need to consider the influence of dynamicity of the network on transmissions. The difficulty mainly comes from mobile nodes, as the states (transmission probability) of these nodes are unknown, which may cause unpredictable interference and collisions. Surprisingly, we show that with a very simple contention balancing strategy, the broadcasters can be elected very efficiently.

Furthermore, to avoid the heavy interference between nodes in adjacent cells, we use a TDMA-like scheme to arrange the node transmissions. In particular, we first color the grid based on the coordinates of cells, and assign different time for nodes in different cells to transmit based on the coloring. A cell g with coordinate (x, y) gets the color $color(g) = a * (x \bmod a) + y \bmod a$, where a is a constant which will be given later. Notice that there are totally $a*a$ colors ($\{0, 1, \dots, a*a-1\}$) used. A node will get the color of the cell when it moves into the cell. Because each node can get its coordinates, it can then know its color after waking up or every moving. With the above coloring, we then define a TDMA scheme as follows:

- Each round is divided into $2 * a * a$ slots, the first $a * a$ slots for broadcaster election period, and the next $a*a$ slots for broadcaster dissemination period;
- Nodes in a cell with color i for $i \in [0, a*a-1]$ execute the first and the second periods in the i -th slot and the $(a * a + i)$ -th slot respectively in each round.

With the above TDMA-scheme, it ensures that in two periods, nodes in nearby cells will not transmit simultaneously, such that the collisions between nearby cells are avoided. But interference and collisions between nodes in the same cell and cells with same color cannot be handled with the TDMA scheme. In the algorithm, our assumption on dynamicity makes sure that once a broadcaster is elected, it will not move out of current cell during the remaining time of the phase. Notice that this assumption is necessary, as it needs $\Omega(\log n)$ rounds to disseminate a message for a node, if we require a high probability guarantee.

We next describe the algorithm in more detail subsequently.

4.2 Detailed Algorithm

The broadcast algorithm is given in Algorithm 1, which is similar to the leader election scheme in [12]. Comparing with the leader election in single hop static networks

in [12], our dynamic and multi-hop leader election algorithm here is more complex. The setting of parameters a, k, p can be found in Table 1. As introduced before, the algorithm execution is divided into phases, and each phase contains $k * (\log n + \log R_c)$ rounds. Each round is divided into two periods: broadcaster election period and broadcaster dissemination period. In each phase, nodes may stay in four types of states:

- State \mathbb{A} means that the node has the message \mathcal{M}_s of the source node. Nodes in state \mathbb{A} compete to become the broadcaster in broadcaster election period;
- State \mathbb{I} means that the node does not have the message \mathcal{M}_s . Nodes in state \mathbb{I} always keeps listening in current phase;
- State \mathbb{B} means that the node is the broadcaster in it's cell. Nodes in state \mathbb{B} broadcast in broadcaster dissemination period;
- State \mathbb{S} means that the node has the message \mathcal{M}_s , but failed in broadcaster competition of current phase. Nodes in \mathbb{S} will keep silent until the end of the current phase.

Broadcaster Election Period. In this period, inactive nodes do nothing and active nodes (nodes in state \mathbb{A}) transmit \mathcal{M} with probability p to compete for broadcasters. \mathcal{M} is a message with competition signal and location information, and the value of p is given in Table 1. For any active node v in this period, when v receives \mathcal{M} from other nodes in the same cell, it will join state \mathbb{S} , and keep silent until the end of the current phase. In the last round of the current phase, if v is still active, then v becomes a broadcaster in its own cell and move into state \mathbb{B} ;

Broadcaster Dissemination Period. In the broadcaster dissemination period, broadcasters in the same color transmit message \mathcal{M}_s in the same slots as defined in the TDMA scheme. Transmission message \mathcal{M}_s contains the message of the source node. After this period, it can be shown that all inactive nodes within $(1 + \frac{3\epsilon}{2})R_c$ from a broadcaster can receive \mathcal{M}_s . Hence, after each phase, the message \mathcal{M}_s can be propagated for one hop.

5 ANALYSIS OF THE BROADCAST ALGORITHM

In this section, we analyze the correctness and efficiency of the broadcast algorithm. The parameter $\gamma, \gamma_1, \gamma_2$ and λ_1 are given in the Table 1.

Theorem 1: If the dynamicity rate $\lambda < \lambda_1$ and the stability parameter $T \geq k * (\log n + \log R_c)$, each node can get the message of the source node in D_T rounds. Notice that our result has the optimal time complexity, as any algorithm needs D_T time to complete dynamic broadcast.

We next prove Theorem 1. Basically, we will show that at the final round of each phase, the broadcasters constitute a $\frac{3\epsilon}{2}R_c$ -dominating set with respect to active nodes. As discussed before, this means that for each inactive node, if a stable link exists between it and an active node at the final round of the phase, there exists

TABLE 1: Parameters in model, algorithm and analysis

Parameter	Value	Parameter	Value
R_T	$(P/\beta N)^{1/\alpha}$	R_c	$(1-\epsilon)R_T$
ϵ	$\in (0, 1)$	λ	$\in (0, \lambda_1)$
p	$c/(4c_{max})$	c	$\frac{1-(1-\epsilon)^\alpha}{2\alpha+1\beta}$
c_{max}	$\frac{48}{1-2^{1-\alpha/2}}$	ζ	$\frac{1-(2^{1-\alpha/2})}{2}$
γ	$\frac{p(1-p)}{8*(2s+5)^2}$	γ_1	$1-\gamma$
τ	$\gamma_2/\gamma_1 > 1$	\hat{a}	$a-1-2\sqrt{2}$
λ_1	$\frac{\gamma_1+\rho/(1-\rho)}{1-\rho} - 1 > 0$		
$f(\alpha, \beta)$	$\lceil (\frac{(32*\frac{\alpha-1}{\alpha-2}+4)*\beta*(1-\epsilon)^{-\alpha}*(\frac{\sqrt{2}}{\epsilon})^\alpha}{(1+\frac{\zeta}{2})^{-\alpha}*(1-\epsilon)^{-\alpha}-1})^{1/\alpha} + 1 + 2\sqrt{2} \rceil$		
a	$\max\{f(\alpha, \beta)\}$, for $\alpha_{min} < \alpha < \alpha_{max}$, $\beta_{min} < \beta < \beta_{max}$		
s	$(\frac{3*2^{\alpha+5}*\beta}{(1-(1-\epsilon)^\alpha)(1-2^{1-\alpha/2})})^{1-\alpha/2}$		
γ_2	$(\gamma_1 + \rho/(1-\rho))(1+\lambda)^{2\zeta} < 1$		
ς	$\lceil \max\{(4\tau)/(a_1(1-\gamma_2)) + 1, 4\tau + 1\} \rceil$		
n	unknown but sufficiently large		
α, β	determined by hardware and environment		
N	determined by environment		
P	determined by hardware		
D_T	determined by network itself		
k	sufficiently large constant		
ρ	sufficiently small constant		
a_1	the constant hidden behind the Ω notation in the probability guarantee in Corollary 2		

a broadcaster within distance $(1 + \frac{3\epsilon}{2})R_c$. With the TDMA scheduling of broadcaster's transmission, it ensures that at the final round of the phase, each broadcaster can disseminate the message within distance $(1 + \frac{3\epsilon}{2})R_c$. In other words, as long as a stable link exists at the final round of a phase between an inactive node and an active node, the inactive node will receive the message. By the condition that stability parameter $T \geq k * (\log n + \log R_c)$, each stable link on the stable path between the source node and a particular node always be stable at the final round of a phase. Hence, after D_T time, every node can receive the message.

We first analyze the algorithm execution in a particular phase. For broadcasters, we have the following result.

Lemma 1: If the dynamicity rate $\lambda < \lambda_1$, w.h.p., at the final round of each phase, exactly one broadcaster will be elected in each cell where there are active nodes competing in current phase.

The proof of Lemma 1 is very technical. To make the proof of the main result clear, we give the proof of Lemma 1 later in Section 5.1.

Based on the above Lemma 1, we analyze the message dissemination of broadcasters in the broadcaster dissemination periods in each round.

Lemma 2: At the final round of each phase, when a broadcaster u locally broadcasts a message, the TDMA scheme ensures that u can send its message to all nodes within distance $(1 + \frac{3\epsilon}{2})R_c$.

Proof: Considering the case at the final round of a phase, we assume that u is in state B , and the cell u stays has color i . As mentioned before, a cell g with coordinate (x, y) gets the color $color(g) = a * (x \bmod a) + y \bmod a$, where a is a constant which will be given later. By

Lemma 1, there is at most one node in each cell joins state \mathbb{B} . By the scheduling of broadcasters' transmissions, in the broadcaster dissemination period, u will simultaneously transmit with other broadcasters in the cells with color i . This means that for any two broadcasters transmitting together, the distance between them is at least $(a - 1 - 2\sqrt{2}) * \frac{\epsilon R_c}{\sqrt{2}}$ even considering the noise in location estimation. Let $\hat{a} = a - 1 - 2\sqrt{2}$. In other words, the simultaneously transmitting broadcasters constitute a $\hat{a} * \frac{\epsilon R_c}{\sqrt{2}}$ -independent set.

Now we consider a node v with $d(u, v) \leq (1 + \frac{3\epsilon}{2})R_c$, and prove that v can receive the message when u locally broadcasts. We divide the whole space into annuluses $\{C_b : b \geq 1\}$, where C_b denotes the annulus with distance from v between $(b-1)\hat{a} * \frac{\epsilon R_c}{\sqrt{2}}$ and $b\hat{a} * \frac{\epsilon R_c}{\sqrt{2}}$. Let B be the set of simultaneously transmitting broadcasters that are in state \mathbb{B} and located in C_b for $b \geq 2$. By the above analysis, the disks centered at broadcasters in B with radius $\hat{a} * \frac{\epsilon R_c}{2\sqrt{2}}$ are disjoint, and these disks are in the annulus with distance from v between $(b - \frac{3}{2})\hat{a} * \frac{\epsilon R_c}{\sqrt{2}}$ and $(b + \frac{1}{2})\hat{a} * \frac{\epsilon R_c}{\sqrt{2}}$. Then we can bound the number of broadcasters in B_j as follows

$$|B_j| \leq \frac{\pi((b + \frac{1}{2})\hat{a} * \frac{\epsilon R_c}{\sqrt{2}})^2 - \pi((b - \frac{3}{2})\hat{a} * \frac{\epsilon R_c}{\sqrt{2}})^2}{\pi(\hat{a} * \frac{\epsilon R_c}{2\sqrt{2}})^2} \leq 16 * b \quad (2)$$

Furthermore, it is easy to get that in C_1 , the number of broadcasters that simultaneously transmit with u is at most 4, and the interference caused by these broadcasters is at most $I_{b=1} = 4P * (\hat{a} * \frac{\epsilon R_c}{\sqrt{2}})^{-\alpha}$. Then we have

$$\begin{aligned} SINR(v, u, \mathcal{T}) &\geq \frac{P * ((1 + \frac{3\epsilon}{2}) * R_c)^{-\alpha}}{N + \sum_{b=2}^{\infty} 16b * P * ((b-1)\hat{a} * \frac{\epsilon R_c}{\sqrt{2}})^{-\alpha} + I_{b=1}} \\ &\geq \frac{P * ((1 + \frac{3\epsilon}{2}) * R_c)^{-\alpha}}{N + (32 * \frac{\alpha-1}{\alpha-2} + 4)(P * \hat{a}^{-\alpha} * (\frac{\epsilon R_c}{\sqrt{2}})^{-\alpha})} \\ &= \frac{\beta N ((1 + \frac{3\epsilon}{2})(1-\epsilon))^{-\alpha}}{N + (32 * \frac{\alpha-1}{\alpha-2} + 4) * N \beta (\hat{a} * (1-\epsilon) * (\frac{\epsilon}{\sqrt{2}}))^{-\alpha}} \\ &\geq \beta \end{aligned}$$

According to the above, $SINR(v, u, \mathcal{T}) \geq \beta$, v receives the message from u . \square

Proof of Theorem 1: We claim that an inactive node v can receive the message \mathcal{M}_s , w.h.p., if its has a stable link with an active node u . By the definition of a stable link, the link exists for at least $T \geq k * (\log n + \log R_c)$ rounds, which means that a stable link exists on at least a final round in a phase. We assume that the stable link between u, v exists on the final round of phase t . By Lemma 1, after the broadcaster election period in the final round of phase t , w.h.p., there is a broadcaster elected in u 's cell. Assume this broadcaster is w , $d(w, u) \leq \frac{3\epsilon R_c}{2}$ since u and w are in same cell. We can get $d(w, v) \leq d(w, u) + d(u, v) \leq (1 + \frac{3\epsilon}{2})R_c$. Then by the algorithm, each broadcaster will locally broadcast

the message \mathcal{M}_s during the broadcaster dissemination period in the final round. And by Lemma 2, when w locally broadcast, v will receive \mathcal{M}_s .

We then consider how long it takes from the beginning of the algorithm till a node v receives the message. Let $\mathcal{P} = \{e_1, e_2, \dots, e_d\}$ be the stable path between the source s and v . By the definition of stable path, the stable links in \mathcal{P} keep stable for T rounds successively. Then based on above analysis, it can be inductively shown that for each stable link $e_l = (u_l, v_l)$ with $1 \leq l \leq d$, v_l will get the message \mathcal{M}_s after u_l becomes active for $O(\log n + \log R_c)$ rounds, w.h.p. Hence, after at most D_T rounds, node v will receive \mathcal{M}_s w.h.p.

Combining all above together, by well tuning the constant parameters in the algorithm, it can be shown that after D_T rounds, all nodes can receive the message \mathcal{M}_s , w.h.p., which completes the proof. \square

5.1 Proof of Lemma 1

According to the TDMA scheduling scheme in the algorithm, nodes in cells with the same color execute the broadcaster election process together. Hence, in subsequence, we analyze the broadcaster election in cells with a particular color j for $0 \leq j < a^2$.

We first define some notations. Consider a cell g with color j . Nodes in cell g are divided into classes $\{V_i^g : i = 0, 1, \dots, \log(3\epsilon R_c)\}$ as defined in the model. Let $V_g(r)$ and $\hat{V}_g(r)$ be the set of active nodes in cell g at the beginning and at the end of a round r respectively. Define $V_i(r) = \cup_{color(g)=j} V_i^g(r)$, $\hat{V}_i(r) = \cup_{color(g)=j} \hat{V}_i^g(r)$, $V_{<i}(r) = \cup_{i' < i} V_{i'}(r)$, $\hat{V}_{<i}(r) = \cup_{i' < i} \hat{V}_{i'}(r)$, $n_i(r) = |V_i(r)|$, $n_{<i}(r) = |V_{<i}(r)|$. And $\hat{n}_i(r)$, $\hat{n}_{<i}(r)$, $V_{>i}(r)$ and $\hat{V}_{>i}(r)$ are defined similarly. Let the dynamicity rate $\lambda = \psi * \lambda_1$, where ψ is a constant and $0 < \psi < 1$.

Now we consider a phase h for the algorithm execution of nodes in cell g . To prove Lemma 1, we assume that at the beginning of the phase, the set of active nodes in cell g is not empty. By the algorithm, active nodes will transmit \mathcal{M} with constant probability p in the broadcaster election period of each round in phase h . Active nodes receiving \mathcal{M} joins state \mathcal{S} and stop the competition in the current phase. The nodes that are still active at the end of the phase become broadcasters. Hence, equivalently, we only need to show that at the end of the phase, all sets V_i for $i = 0, \dots, \log(3\epsilon R_c) - 1$ become empty. This means that only one active node left at the end of the phase, i.e., exactly one broadcaster is elected in the cell.

For any active node v , let $A(v, d)$ be the set of active nodes within distance d from v and $E_t^i(u) = A(u, 2^{t+1}2^i) \setminus A(u, 2^t2^i)$. We say a node is *sparse* if for every $t \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, $E_t^i(u) \leq 48 * 2^{t(\alpha/2+1)}$. Let $S_i \subseteq V_i$ be the largest subset of sparse nodes in V_i that satisfies for any nodes u, v in S_i , $d(u, v) \geq (s+2)2^i$ with $s = \left(\frac{3 \cdot 2^{\alpha+5} \cdot \beta}{(1-(1-\epsilon)^\alpha)(1-2^{1-\alpha/2})}\right)^{\frac{1}{1-\alpha/2}}$.

In the following we show that after each round, with certain moderate probability guarantee, the number of

nodes in V_i for each $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$ will be reduced by a constant factor. This result is proved in two steps: first, it is shown that after each round, the size of S_i will be reduced by a constant factor, and then it is proved that S_i contains a constant fraction of nodes of V_i .

Lemma 3: In each round, for each non-empty set S_i , $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, a constant fraction of nodes in S_i can receive a message from their nearest neighbors and become inactive, with probability of $1 - e^{-\Omega(|S_i|)}$.

Proof: We consider a node u in S_i , and assume that its nearest neighbor is v . The probability of u listens and v transmits is $p(1-p)$. Assuming the event that u listens and v transmits happens, we focus on the interference u experiences. Let T_i be the set of nearest neighbors of all nodes in S_i . The interference on u can be divided into two parts: interference from nodes in $S_i \cup T_i$ and interference from other nodes.

We first bound the interference of the first part, i.e., those from nodes in $S_i \cup T_i$. By the definition of sparse nodes, any pair of nodes in S_i have distance at least $(s+2)2^i$ from each other, and each node in S_i has distance in the range $[2^i, 2^{i+1})$ with its nearest neighbor. Hence, nodes in $(S_i \cup T_i) \setminus \{u, v\}$ have distance at least $s * 2^i$ from u . Thus the interference I_1 at u that is from nodes in $(S_i \cup T_i) \setminus \{u, v\}$ is bounded by:

$$I_1 = \sum_{t=\log s} |E_t^i(u)| \frac{P}{(2^i 2^t)^\alpha} \leq \frac{48P}{2^{i\alpha}} \cdot \frac{1}{s^{\alpha/2-1}} \cdot \frac{1}{1-2^{1-\alpha/2}}. \quad (3)$$

We next bound interference from nodes not in $S_i \cup T_i$. Define $I(v)$ as the interference at all nodes in S_i that is generated by a node v with $v \notin S_i \cup T_i$. Let I_{sum} be the sum of all $I(v)$, $v \notin S_i \cup T_i$. We next upper bound I_{sum} . Then, it can be obtained that for at least $\frac{1}{c}$ fraction of nodes in S_i , the interference outside from $S_i \cup T_i$ on them is no larger than $\frac{c}{|S_i|} * I_{sum}$.

For a node $v \notin S_i \cup T_i$, as we have defined, $I(v)$ is the sum of interference caused by v and on all nodes in $E_t^i(v) \cap S_i$ over all annuluses. With an area argument similar with Inequality (2), it can be obtained that $|E_t^i(v) \cap S_i| \leq 24 * 2^{2t}$. Then

$$\begin{aligned} I(v) &= \sum_{t=0}^{\infty} |E_t^i(v) \cap S_i| \frac{P}{(2^i 2^t)^\alpha} = \frac{P}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{|E_t^i(v) \cap S_i|}{2^{t\alpha}} \\ &\leq \frac{P}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{24 * 2^{2t}}{2^{t\alpha}} = \frac{24P}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{1}{2^{t(\alpha-2)}} \\ &< \frac{24 * P}{2^{i\alpha}} \left(\frac{1}{1-2^{2-\alpha}} \right) \end{aligned}$$

Let $c_{max} = \frac{48}{1-2^{1-\alpha/2}}$ then $I(v) < c_{max} P / 2^{i\alpha}$.

Claim 1: Given any constant c , by setting $p = c/(4c_{max})$, with probability $1 - e^{-\frac{c^2}{24c_{max}^2}|S_i|}$, at least half of nodes in S_i experience the interference I_2 from nodes not in $S_i \cup T_i$ no larger than $cP/2^{i\alpha}$.

Proof: Consider a node $u \in S_i$. Let $I_2(u)$ denote the interference experienced by u that are caused by nodes not in $S_i \cup T_i$. Then we prove the Claim in two cases.

Case 1. $c \geq c_{max}$.

$$\begin{aligned} I_2(u) &\leq \sum_{t=0}^{\infty} |E_t^i(u)| \frac{P}{(2^t 2^i)^\alpha} = \frac{P}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{|E_t^i(u)|}{2^{t\alpha}} \\ &\leq \frac{P}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{48 * 2^{t(\alpha/2+1)}}{2^{t\alpha}} = \frac{48P}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{1}{2^{t(\alpha/2-1)}} \\ &< \frac{48P}{2^{i\alpha}} \left(\frac{1}{1-2^{1-\alpha/2}} \right) \leq c_{max} P / 2^{i\alpha} \leq cP / 2^{i\alpha} \end{aligned}$$

Case 2. $c < c_{max}$.

We define a random variable x_v

$$x_v = \begin{cases} I(v)2^{i\alpha} / (c_{max}P) & \text{when node } v \text{ transmits} \\ 0 & \text{when node } v \text{ listens} \end{cases}$$

Then we have

$$\begin{aligned} \mathbb{E} \left[\sum_{v \notin S_i \cup T_i} x_v \right] &= \sum_{v \notin S_i \cup T_i} p * I(v)2^{i\alpha} / (c_{max}P) \\ &= p \sum_{v \notin S_i \cup T_i} I(v)2^{i\alpha} / (c_{max}P) \end{aligned}$$

Considering that $\frac{|S_i|}{2} * c * P / 2^{i\alpha} \leq \sum I(v) \leq |S_i| * c_{max} * P / 2^{i\alpha}$, we can get $(c^2 / 8c_{max}^2) |S_i| \leq \mathbb{E} \left[\sum_{v \notin S_i \cup T_i} x_v \right] \leq c |S_i| / (4c_{max})$. Let $\mu = \mathbb{E} \left[\sum_{v \notin S_i \cup T_i} x_v \right]$. Notice that $x_v \in [0, 1]$, applying the standard Chernoff bound for the set of independent random variable $\{x_v : v \notin S_i \cup T_i\}$, it can be obtained that

$$Pr \left(\sum_{v \notin S_i \cup T_i} x_v \geq 2 * (c |S_i| / (4c_{max})) \right) \leq e^{-\frac{c^2}{24c_{max}^2} |S_i|}$$

Thus, with probability at least $1 - e^{-\frac{c^2}{24c_{max}^2} |S_i|}$,

$$\begin{aligned} I_{sum} &= \sum_{v \notin S_i \cup T_i} I(v) = \sum_{v \notin S_i \cup T_i} x_v * c_{max} P / 2^{i\alpha} \\ &\leq (2c |S_i| / (4c_{max})) * c_{max} P / 2^{i\alpha} \\ &= c |S_i| P / 2^{i\alpha+1} \end{aligned}$$

Hence, it is impossible for more than half of nodes in S_i experiencing interference from nodes not in $S_i \cup T_i$ larger than $cP / 2^{i\alpha}$. \square

Considering the interference outside from and inside from set $S_i \cup T_i$, with probability at least $1 - e^{-\frac{c^2}{24c_{max}^2} |S_i|}$, at least half of nodes in S_i experience the interference no larger than $2cP / 2^{i\alpha}$, by setting $c = \frac{1-(1-\epsilon)^\alpha}{2^{\alpha+1}\beta}$. Then u can receive a message from its nearest neighbor v by the SINR condition as follows.

$$SINR(u, v) > \frac{P / 2^{\alpha(i+1)}}{2cP / 2^{i\alpha} + N} \geq \beta$$

Combining the assumption that u listens and its nearest neighbor v transmits, which occurs with probability $p(1-p)$, and at least half of nodes in S_i can receive messages from their nearest neighbors, it can be proved that $p(1-p) * |S_i| / 2$ nodes become inactive in expectation. Applying Chernoff bound, the Lemma is then proved. \square

After proving that in each round, a constant fraction of nodes in S_i become inactive, the next step is to prove that for $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, a constant fraction of nodes in V_i is in S_i .

Lemma 4: In a round, for $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, if $n_{<i} \leq \zeta n_i$ with $\zeta = \frac{1-(2^{1-\alpha/2})}{2}$, then $\frac{|S_i|}{|V_i|} \geq \frac{1}{2(2s+5)^2}$

Proof: We prove the Lemma by showing that a constant fraction of nodes in V_i are sparse nodes and a constant fraction of sparse nodes are in S_i .

Claim 2: In a round, for $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, if $n_{<i} \leq \zeta n_i$ with $\zeta = \frac{1-(2^{1-\alpha/2})}{2}$, then at least half of nodes in V_i are sparse nodes.

Proof: Define an *Excellent Sparse Node* $u \in V_i$ as follows: for every $t \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, node u satisfies that $|E_t^i(u) \cap V_{\geq i}| \leq 24 * 2^{t(\alpha/2+1)}$ and $|E_t^i(u) \cap V_{<i}| \leq 24 * 2^{t(\alpha/2+1)}$. Obviously, an *Excellent Sparse Node* must be a sparse node. So the fraction of *Excellent Sparse Nodes* in V_i is a lower bound of the fraction of sparse node in V_i . We next focus on the fraction of *Excellent Sparse Node* in V_i .

We first show that the condition $|E_t^i(u) \cap V_{\geq i}| \leq 24 * 2^{t(\alpha/2+1)}$ holds for any node in V_i . Noting that nodes in $V_{\geq i}$ have distance at least 2^i with each other, the disks centered at these nodes and with radii 2^{i-1} are disjoint. For any node $u \in V_i$ and any annulus $E_t^i(u)$, using an area argument in the following Eq.4, the condition can be proved.

$$\begin{aligned} \frac{\pi(2^{t+1}2^i + 2^{i-1})^2 - \pi(2^t 2^i - 2^{i-1})^2}{\pi 2^{2(i-1)}} &= 3 * 2^{t+2} * (2^t + 1) \\ &\leq 3 * 2^{2t+3} \\ &< 24 * 2^{t(\alpha/2+1)} \end{aligned} \quad (4)$$

We next consider the condition about $|E_t^i(u) \cap V_{<i}|$ on node $u \in V_i$. Define Γ_t^i be the sum of nodes in set $E_t^i(u) \cap V_{<i}$ for each node u in V_i . Then we get

$$\begin{aligned} \Gamma_t^i &= \sum_{u \in V_i} |E_t^i(u) \cap V_{<i}| = \sum_{u \in V_{<i}} |E_t^i(u) \cap V_i| \\ &\leq n_{<i} * 24 * 2^{2t} \leq \zeta * n_i * 24 * 2^{2t} \end{aligned}$$

Based on above, it can be obtained there are at most $\zeta * 2^{t(1-\alpha/2)}$ fraction of nodes in V_i that are not excellent sparse nodes in annulus $E_t^i(u)$ for each node $u \in V_i$, otherwise $\Gamma_t^i > \zeta * n_i * 24 * 2^{2t}$. Then we sum up the number of nodes in each annulus that are not excellent sparse, to bound the total number of non-excellent sparse nodes in V_i .

$$\begin{aligned} \sum_{t=0}^{\log(3\epsilon R_c)-1} n_i * \zeta * 2^{t(1-\alpha/2)} &= n_i * \zeta * \sum_{t=0}^{\log(3\epsilon R_c)-1} (2^{1-\alpha/2})^t \\ &\leq n_i * \zeta * \frac{1}{1 - (2^{1-\alpha/2})} \\ &= \frac{1}{2} n_i. \end{aligned}$$

Combining all above together, the Claim is proved. \square

Claim 3: For $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, at least $\frac{1}{(2s+5)^2}$ fraction of *sparse nodes* in V_i are in set S_i .

Proof: By the maximality of S_i , the disks centered at nodes in S_i with radii $(s+2)2^i$ cover all sparse nodes in V_i . Furthermore, notice that the distance between any pair of sparse nodes in V_i is at least 2^i . Hence, the disks with radii 2^{i-1} that are centered at sparse nodes in V_i are disjoint. Let D_v^d denote the disk centered at node v that has radius d .

Now consider a node $u \in S_i$, we upper bound the number of sparse nodes in $D_u^{(s+2)2^i}$. Let B_u be the set of sparse nodes in V_i that locate in $D_u^{(s+2)2^i}$. By above analysis, the disks $\{D_v^{2^{i-1}} : v \in B_u\}$ are all in the disk $D_u^{(s+2)2^i + 2^{i-1}}$. Then using an area argument, we can bound B_u as follows,

$$\frac{\pi * ((s+2)2^i + 2^{i-1})^2}{\pi * (2^{i-1})^2} = (2s+5)^2.$$

We sum up the bound on the number of sparse nodes within distance $(s+2)2^i$ of nodes in S_i , to get an upper bound on the total number of sparse nodes in V_i , as follows,

$$\sum_{u \in S_i} |B_u| \leq (2s+5)^2 |S_i|, \quad (5)$$

which completes the proof. \square

By Claim 2, 3, Lemma 4 can be proved. \square

The following result is a direct corollary of Lemma 3 and Lemma 4.

Corollary 2: In a round r , for $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, if $n_{<i} \leq \zeta n_i$ with $\zeta = \frac{1-(2^{1-\alpha/2})}{2}$, with probability $1 - e^{-\Omega(|V_i|)}$, γ fraction of nodes in V_i will become inactive, where $\gamma = \frac{p(1-p)}{8*(2s+5)^2}$.

In the above, we have shown that when $n_{<i} \leq \zeta n_i$, $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, with probability $1 - e^{-\Omega(|V_i|)}$, γ fraction of nodes in V_i will become inactive. However, even we get the reduction ratio γ for V_i , V_i will not always be reduced for three reasons: first, some new nodes may join or stop the algorithm execution; second, the mobility of nodes; third, some active nodes in $V_{<i}$ may fall into V_i because their nearest neighbors become inactive. We need to show that even with these unpredictable fluctuations, all V_i for $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$ will be finally reduced to empty in $O(\log n + \log R_c)$ rounds, i.e., the time in a phase.

Firstly, we divide the running process of algorithm into consecutive intervals Λ_i for $i = 0, 1, \dots$. Each interval consists of $\varsigma = \max\{(4\tau)/(a_1(1-\gamma_2)) + 1, 4\tau + 1\}$ rounds, as is given in Table 1.

Then, we define a series of vectors as the upper-bound for each class $|V_i|$. To be more detailed, we define $\{m_i(t) : t \geq 0 \text{ and } 0 \leq i \leq \log(3\epsilon R_c) - 1\}$ as follows, and let $\hat{m}_i(t+1) = \gamma_1 m_i(t)$.

$$\forall t \geq 0 : m_i(t) = \begin{cases} n/\gamma_1 & t \leq T_i \\ \lfloor m_i(t-1) * \gamma_2 \rfloor & t > T_i \end{cases}$$

Here $T_i = i * h$ and $h = \lceil \log_{\gamma_2} \rho \rceil$.

Define T_c to be the earliest round when all $\hat{m}_i(T_c)$ become 0. Then according to the definition, $T_c \in O(\log n + \log R_c)$.

We define random events $\mathcal{E}(j)$ for $j \geq 0$: $\mathcal{E}(j)$ occurs when in some round r , $\hat{n}_i(r) \leq m_i(j)$ for all $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$. If for any round in interval Λ_a , $\mathcal{E}(j)$ occurs, we say $\mathcal{E}(j)$ always occurs in interval Λ_a . Obviously, all $|V_i|$ become empty when $\mathcal{E}(T_c)$ occurs. So in the following, we only need to analyze when $\mathcal{E}(T_c)$ occurs.

Note that $\mathcal{E}(0)$ always occurs for any round r , since for all $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, $m_i(0) = n/\gamma_1 \geq \hat{n}_i(r)$. With a general assumption that $\mathcal{E}(j)$ always occurs in interval Λ_a , we analysis the situations in which $\mathcal{E}(j+1)$ and $\mathcal{E}(j-1)$ occurs via Lemma 5, 6, 7 and 8.

Lemma 5: For interval Λ_a and Λ_{a+1} , if $\mathcal{E}(j)$ always occurs in interval Λ_a , then $\mathcal{E}(j)$ or $\mathcal{E}(j-1)$ always occurs in interval Λ_{a+1} .

Proof: We generally assume r and r_1 to be any round in Λ_a and Λ_{a+1} respectively. If $0 \leq j \leq 1$, obviously $\mathcal{E}(0)$ occurs at r_1 ; And we consider the case that $j \geq 2$. Since $\mathcal{E}(j)$ occurs at round r , then for any $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, $\hat{n}_i(r) \leq m_i(j)$, we have

$$\begin{aligned} \hat{n}_i(r_1) &\leq \hat{n}_i(r) * (1+\lambda)^{r_1-r} + \sum_{s=0}^{i-1} n_s^g(r) * (1+\lambda)^{r_1-r} \\ &\leq \hat{n}_i(r) * (1+\lambda)^{2\varsigma} + \sum_{s=0}^{i-1} n_s(r) * (1+\lambda)^{2\varsigma} \\ &\leq m_i(j) * (1+\lambda)^{2\varsigma} + \sum_{s=0}^{i-1} m_s(j) (1+\lambda)^{2\varsigma} \\ &\leq m_i(j) * (1+\lambda)^{2\varsigma} + (1+\lambda)^{2\varsigma} \sum_{s=0}^{i-1} m_s(j) \end{aligned} \quad (6)$$

The analysis here is divided into two cases that $m_{i-1} < n/\gamma_1$ and $m_{i-1} = n/\gamma_1$. Notice that it is impossible for $m_{i-1} > n/\gamma_1$.

If $m_{i-1}(j) < n/\gamma_1$, $\forall s \in \{0, 1, \dots, i-1\}$, $m_s(j) = \rho m_{s+1}(j)$, and $\sum_{s=0}^{i-1} m_s(j) \leq m_i(j)\rho/(1-\rho)$. Thus,

$$\begin{aligned} \hat{n}_i(r_1) &\leq m_i(j) * (1+\lambda)^{2\varsigma} + (1+\lambda)^{2\varsigma} \sum_{s=0}^{i-1} m_s(j) \\ &\leq m_i(j) * (1+\lambda)^{2\varsigma} + (1+\lambda)^{2\varsigma} m_i(j)\rho/(1-\rho) \\ &= \frac{(1+\lambda)^{2\varsigma}}{1-\rho} m_i(j) = \frac{(1+\lambda)^{2\varsigma} \gamma_2}{1-\rho} m_i(j-1) \end{aligned} \quad (7)$$

Since $\frac{(1+\lambda)^{2\varsigma} \gamma_2}{1-\rho} \leq 1$, $\hat{n}_i(r_1) < m_i(j-1)$ in the case that $m_{i-1}(j) < n/\gamma_1$.

If $m_{i-1}(j) = n/\gamma_1$, we can see that $m_i(j-1) = m_i(j) = m_{i-1}(j) = n/\gamma_1$, then $\hat{n}_i(r_1) \leq m_i(j-1)$.

Until then, we get that when $j \geq 2$, $\hat{n}_i(r_1) < m_i(j-1)$ for all $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$. And Combined all above together, the lemma is proved. \square

Lemma 6: For any round r and r_1 in Λ_a and Λ_{a+1} respectively, if $\hat{n}_i(r) \leq \hat{m}_i(j+1)$, then $\hat{n}_i(r_1) \leq m_i(j+1)$.

Proof: Case 1: if $m_i(j+1) = n/\gamma_1$, then it is easy to get $\hat{n}_i(r_1) \leq m_i(j+1)$. Case 2: if $m_i(j+1) < n/\gamma_1$, as

defined in the model, for each cell g , at the beginning of round r_1 , $n_i^g(r_1) \leq (1 + \lambda)^{r_1 - r} \hat{n}_i^g(r)$. Summing up $n_i^g(r_1)$ over all cells, we get $n_i(r_1) \leq (1 + \lambda)^{r_1 - r} \hat{n}_i(r)$, for each $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$. Also for $\forall i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, $n_i(r) \leq \hat{m}_i(j + 1) \leq m_i(j)$. Thus,

$$\begin{aligned} \hat{n}_i(r_1) &\leq \hat{n}_i(r) * (1 + \lambda)^{r_1 - r} + \sum_{s=0}^{i-1} n_s(r) * (1 + \lambda)^{r_1 - r} \\ &\leq \hat{m}_i(j + 1) * (1 + \lambda)^{r_1 - r} + \sum_{s=0}^{i-1} m_s(j) (1 + \lambda)^{r_1 - r} \\ &\leq m_i(j) \frac{\gamma_2}{(1 + \lambda)^{2\zeta - r_1 + r}} - (1 + \lambda)^{r_1 - r} m_i(j) \rho / (1 - \rho) \\ &\quad + (1 + \lambda)^{r_1 - r} m_i(j) \rho / (1 - \rho) \\ &\leq m_i(j + 1) \end{aligned}$$

□

Lemma 7: For any round r and $r + 1$ in interval Λ_a , if $\mathcal{E}(j)$ occurs in round r , then with probability at least $1 - e^{-\Omega(n_i(r+1))}$, $\hat{n}_i(r + 1) \leq \hat{m}_i(j + 1)$, where $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$.

Proof: The Lemma can be proved in three cases.

Case 1: if $n_i(r + 1) < \hat{m}_i(j + 1)$, then $\hat{n}_i(r + 1) \leq n_i(r + 1) \leq \hat{m}_i(j + 1)$. **Case 2:** if $m_{i-1}(j) = n/\gamma_1$, then $m_i(j) = n/\gamma_1$, $\hat{m}_i(j + 1) = n$. and $\hat{n}_i(r + 1) \leq \hat{m}_i(j + 1)$. We next consider the case that $n_i(r + 1) \geq \hat{m}_{j+1}(i)$ and $m_{i-1}(j) < n/\gamma_1$.

Since $\mathcal{E}(j)$ occurs at r , we know that $\hat{n}_i(r) \leq m_i(j)$, then $\hat{n}_{<i}(r) \leq m_{<i}(j) \leq m_i(j) \rho / (1 - \rho)$. Also considering that $n_{<i}(r + 1) \leq (1 + \lambda) \hat{n}_{<i}(r)$. We get

$$\begin{aligned} n_{<i}(r + 1) &\leq (1 + \lambda) \hat{n}_{<i}(r) \\ &\leq (1 + \lambda) m_i(j) \rho / (1 - \rho) \\ &\leq \hat{m}_i(j + 1) * \frac{(1 + \lambda) \rho}{\gamma_1 (1 - \rho)} \\ &\leq n_i(r + 1) \frac{(1 + \lambda) \rho}{\gamma_1 (1 - \rho)} \end{aligned}$$

By setting ρ to be small enough to make sure $\rho / (1 - \rho) < \gamma_1 \zeta / (1 + \lambda)$ and $\gamma_2 < 1$, we obtain $n_{<i}(r + 1) < \zeta n_i(r + 1)$ from the above inequality. Then, in round $r + 1$, by Corollary 2, with probability $1 - e^{-\Omega(n_i(r+1))}$,

$$\hat{n}_i(r + 1) \leq \gamma_1 n_i(r + 1) \leq \gamma_1 m_j(i) = \hat{m}_i(j + 1).$$

□

With the above two results, we are going to bound the probability that $\mathcal{E}(j + 1)$ always occurs in interval Λ_{a+1} when $\mathcal{E}(j)$ always occurs in interval Λ_a .

Lemma 8: If $\mathcal{E}(j)$ always occurs in Λ_a , with probability at least $3/4$, $\mathcal{E}(j + 1)$ always occurs in Λ_{a+1} .

Proof: Combining the assumption that $\mathcal{E}(j)$ occurs in Λ_a and Lemma 7, we get a result that for each round r_2 except the first round in Λ_a , it has the probability at least $1 - e^{-\Omega(n_i(r_2))}$ that $\hat{n}_i(r_2) \leq \hat{m}_i(j + 1)$, where $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$. When $\hat{n}_i(r_2) \leq \hat{m}_i(j + 1)$ is true for $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, from Lemma 6, $\mathcal{E}(j + 1)$ always occurs in Λ_{a+1} . Note that the constant behind the Ω notation in the probability guarantee in

Lemma 7 is the same with that in Corollary 2 and it is recorded as a_1 . We also have set $\varsigma = \lceil \max\{(4\tau)/(a_1(1 - \gamma_2)) + 1, 4\tau + 1\} \rceil$ rounds, where $\tau = \gamma_2/\gamma_1 > 1$. Then for $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$, the probability that $\hat{n}_i \leq \hat{m}_i(j + 1)$ never happened in interval Λ_a is

$$\begin{aligned} e^{-4\tau \hat{n}_i / (1 - \gamma_2)} &\leq (1 - \gamma_2) / (4\tau \hat{n}_i) \\ &\leq (1 - \gamma_2) / (4\tau \hat{m}_i(j + 1)) \\ &= (1 - \gamma_2) / (4m_i(j + 1)). \end{aligned}$$

Applying a union bound on the error probabilities above for all i s, we obtain the probability that at least one \hat{n}_i is always larger than $\hat{m}_i(j + 1)$ in interval Λ_a is at most

$$\begin{aligned} \sum_{i=0}^{\log(3\epsilon R_c) - 1} (1 - \gamma_2) / (4m_i(j + 1)) &\leq \frac{1 - \gamma_2}{4} \sum_{i=0}^{\log(3\epsilon R_c) - 1} \frac{\gamma_1}{n} \cdot \gamma_2^{ih - (j+1)} \\ &\leq \frac{(1 - \gamma_2) \gamma_1}{4n * \gamma_2^{j+1}} \sum_{i=0}^{\log(3\epsilon R_c) - 1} \gamma_2^{ih} \\ &\leq \frac{(1 - \gamma_2) \gamma_1}{4n * \gamma_2^{j+1}} \cdot \frac{1}{1 - \gamma_2^h} \\ &\leq \frac{1}{4} \end{aligned}$$

Hence, with probability at least $3/4$, $\mathcal{E}(j + 1)$ always occurs in Λ_{a+1} when $\mathcal{E}(j)$ always occurs in Λ_a , which completes the proof. □

Lemma 9: $\mathcal{E}(T_c)$ occurs within $O(\log n + \log R_c)$ rounds with high probability.

Proof: Combining what we have known: 1) $T_c \in O(\log n + \log R_c)$; 2) $\mathcal{E}(0)$ always occurs for any round; 3) When $\mathcal{E}(j)$ always occurs in an interval Λ_a , which has constant rounds, in the following Λ_{a+1} , with probability at least $3/4$, $\mathcal{E}(j + 1)$ always occurs and in the rest probability, which is no more than $1/4$, $\mathcal{E}(j)$ or $\mathcal{E}(j - 1)$ always occurs in Λ_{a+1} . It is easy to show that in expectation, after $O(T_c)$ rounds, $\mathcal{E}(T_c)$ occurs. Using Chernoff bound, the Lemma can be proved. □

Now we are ready to prove Lemma 1

Proof of Lemma 1: By setting constant k in the algorithm to be larger than the constant behind the O notation in the time bound of $O(\log n + \log R_c)$ given in Lemma 9, at the final round of a phase, for each non-empty cell g (a cell with active nodes), all V_i^g for $i \in \{0, 1, \dots, \log(3\epsilon R_c) - 1\}$ become empty with high probability. So by the algorithm, there is exactly one active node left for each non-empty cell at the final round of a phase, which completes the proof of lemma 1. □

6 SIMULATION

Simulation results for our global broadcast algorithm in dynamic network are presented in this section. In reality, the efficiency of global broadcast in a network is determined by two factors: the diameter of the network and the time cost for one-hop message dissemination. Considering that the diameter is an inherent feature for the network topology and differ for different networks,

TABLE 2: Parameters in simulation

Parameters	Value	Parameter	Value
n_0	[1000, 10000]	R	30m
λ	[0, 0.3]	ϵ	0.25
α	{3.0, 4.0}	a	9
β	{1.5, 2.0}	p	0.2

we evaluate the efficiency of our global broadcast algorithm in terms of the time needed for one-hop message dissemination. Furthermore, we also present a global broadcast complete ratio, i.e., the ratio of the nodes with the source message among all nodes in the network, under different settings. The broadcast ratio can help learn the speed of global broadcast. Also, a comparison simulation with existing algorithms and the impact of SINR parameters for our algorithm performance are also considered.

Parameter and dynamicity setting. In the simulation, n_0 nodes, which includes an active node with source message and $n_0 - 1$ inactive nodes, are randomly distributed in a network with area of $300m \times 300m$. Each node has the uniform transmission range $30m$. Tab. 2 presents all the parameters in simulation. By giving values of n_0 , the network area, and the transmission range, we have the average number of neighbors for nodes within the transmission range increase from 30 to 300, i.e., the simulated network gradually changes from a sparse network to a dense one. Also, the setting of $30m$ for the transmission range in a $300m \times 300m$ network area makes sure that the diameter of the network is at least 15, which is enough to simulate a multi-hop network. For the dynamicity of nodes in the network, when the dynamicity rate is satisfied, each node has a probability to leave/join the network. Since the movement of nodes from point A to B can be regarded as nodes leaving the network from point A and then joining the network at point B, the dynamicity in our simulation covers leaving, movement of nodes, and joining of new nodes.

Our whole simulation is written as a C++ program with multiple functions, some of which are from the C++ standard library while some are our own creations. Fig. 2 illustrates the flow chart of our simulation. Over 20 runs of the simulation have been carried out for each reported result. All experiments are conducted on a Linux machine with Intel Xeon CPU E5-2670@2.60GHz and 64 GB main memory, implemented in C++ and compiled by the GCC compiler.

6.1 Algorithm performance in dynamic networks

Time for one-hop message dissemination. In each phase, our global broadcast algorithm completes a one-hop message dissemination. In this process leaders will be elected out from active nodes, and disseminate the source message for one hop. The inactive nodes become active when receiving the source message. Observing all one-hop message dissemination in the simulation, we figure out and show the average and maximum number

of rounds used for one-hop message dissemination in Fig. 3 (1) and (2) respectively. In Fig. 3, the x -axes and y -axes represent the number of nodes initially in the network and the number of rounds the algorithm has executed respectively. Fig. 3 (1) depicts the average number of rounds needed for one-hop message dissemination under different dynamicity rate λ . When λ is larger, the dynamicity in the network gets heavier. In Fig. 3 (1), we can see that the average number of rounds increases first and keeps stable when the number of nodes get larger.

It can be known that with a larger λ , it requires more time to complete one-hop message dissemination. As shown in Fig. 3 (1), when $\lambda \leq 0.057$, the average number of rounds for one-hop message dissemination is always smaller than 30, and the average number of rounds rapidly increase in the dynamic scenarios of $\lambda = 0.059$ and $\lambda = 0.061$, which indicates that the largest dynamicity rate our algorithm can efficiently handle is between 0.057 and 0.059. Fig. 3 (2) illustrates a similar results on the maximum number of rounds for one-hop message dissemination.

Broadcast complete ratio. Fig. 4 depicts the broadcast complete ratio as the algorithm executes, in which the x -axes represents the number of rounds the algorithm has executed and the y -axes represents the ratio. In Fig. 4 (1), where $n_0 = 2000$ and $\lambda = 0.010, 0.030, 0.050, 0.055$, the broadcast complete ratio increases from 0 to 1. It can be seen that for broadcast processes with two different λ , the ratio in the case with a larger λ is larger first and then becomes smaller later, which means that the broadcast process with a larger λ goes faster at the beginning and gradually becomes slower than that with a smaller λ . According to our observation, this phenomenon is caused by the dynamicity of network. One the one hand, dynamicity in a network takes challenge to the stable connections between nodes, but on the other the dynamicity facilitates the global broadcast since nodes with source message can move around to deliver source message. At the beginning, the positive impact of dynamicity is larger, so the broadcast process with larger λ goes faster. Gradually when the number of nodes without source message become small, the stable connections from nodes with source message to those without source message become important, and hence the negative impact of dynamicity matters more. Fig. 4 (2)-(4) provide the broadcast complete ratio in the cases of $n_0 = 4000, 6000, 8000$ respectively, from which similar conclusion can be obtained.

Comparison with existing algorithms. In the simulation, we also compare our work with two existing dynamic global broadcast protocols in [1] and [6], named DGB-1 and DGB-2, respectively. In [1], the execution of DGB-1 is divided into successive phases, each of which consists of $\Omega(\log^3 n)$ rounds. In each phase, the nodes that received a message in last phase decrease their transmission probability from 1 to $\frac{1}{\log^* n}$. In [6], nodes that hold the message in DGB-2 always transmit with the probability $\ln n/n$ in each round. We compare

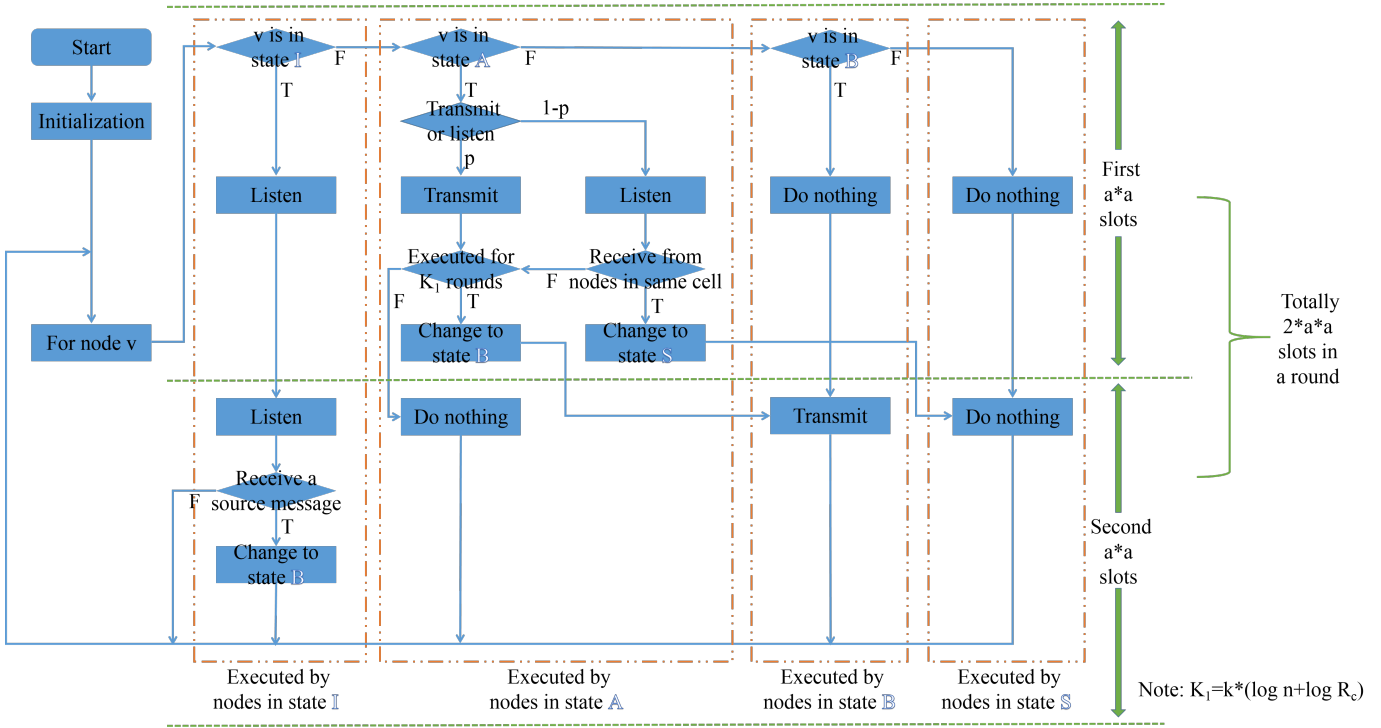
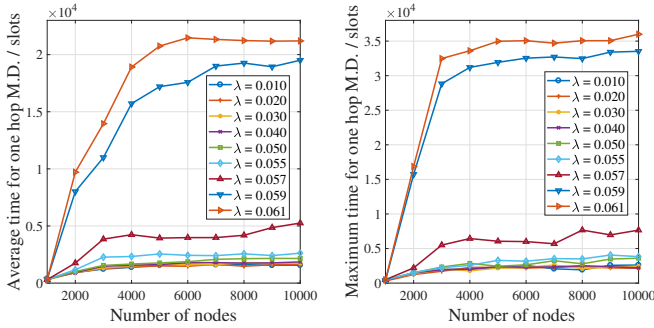
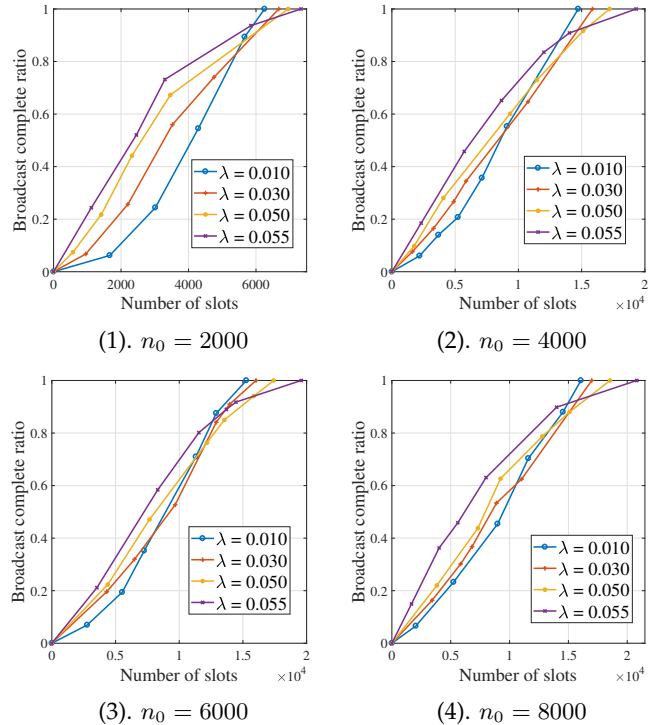


Fig. 2: Flow chart of our simulation



(1). Average number of rounds (2). Max. number of rounds
 Fig. 3: Time for one-hop message dissemination when dynamicity rate varies

our dynamic global broadcast protocol, named DGB-3, with protocols DGB-1 and DGB-2 mentioned above. The comparison results are given in Fig. 5, in which the x -axes and the y -axes represent the number of nodes in network and the average/maximal time for one hop message dissemination respectively. Specifically, in Fig 5 (1), where n_0 varies from 1000 to 10000 and $\lambda = 0.02$, we can see that (1) the average time for one hop M.D. used by DGB-3 approximately increases from 300 to 1600, and gradually becomes stable, (2) DGB-1/DGB-2 have some trivial results when the network is dense/sparse, and (3) our algorithm DGB-3 has better performances than DBG-1 and DGB-2.



(1). $n_0 = 2000$ (2). $n_0 = 4000$
 (3). $n_0 = 6000$ (4). $n_0 = 8000$
 Fig. 4: Broadcast complete ratio in different dynamicity rates

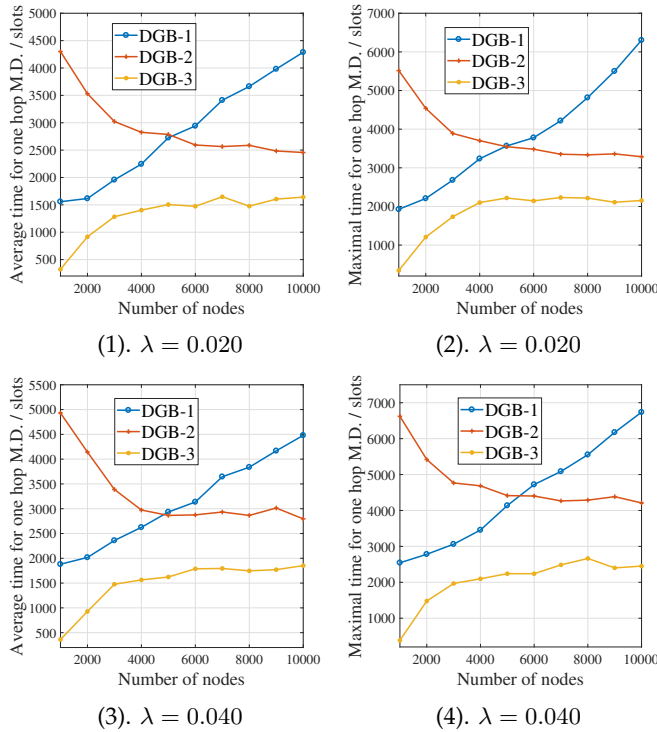


Fig. 5: Time for one-hop message dissemination in various dynamic global broadcast protocols

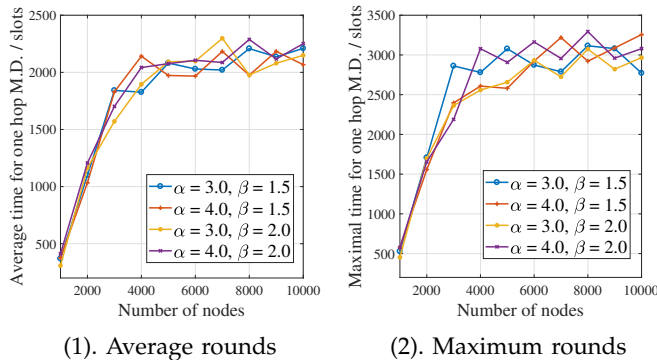


Fig. 6: Time for one-hop message dissemination in different SINR parameters

6.2 Impact of SINR parameters

WLOG fixing $\lambda = 0.050$, we investigate the impact of SINR parameters on our global broadcast algorithm in Fig. 6, where the average/maximum number rounds for one-hop message dissemination is given under different α and β . It can be seen that when α and β vary, the average/maximum rounds used for one-hop message dissemination is almost the same. So, our global broadcast algorithm is insensitive to SINR parameters.

6.3 Summary

In the simulation, our algorithm can complete a global broadcast task under a constant dynamicity rate. Considering that the dynamicity rate is with respect to the dynamicity in every round, although the dynamic rate is not very large, the dynamicity can be significant

after only a few rounds. The global broadcast algorithm with time complexity of $O(D_T)$ is believed to be very efficient in reality since simulation results indicate that the constant hidden behind the big O notation is smaller than 30. Also, our algorithm is insensitive to the SINR parameters.

7 CONCLUSION

In this paper, we have studied the problem of broadcasting a single message in dynamic networks. We proposed a dynamic model based on a local view, which is more suitable for distributed algorithm design. Our local model adopts the realistic SINR model for wireless interference. Under the local dynamic model, we presented a distributed algorithm that can accomplish global broadcast in $O(D_T)$ time with a high probability guarantee, under the constraint of constant dynamicity rate, where D_T is the dynamic diameter proposed for depicting the complexity of dynamic broadcasting. The algorithm can be shown to be asymptotically optimal, by the natural lower bound of D_T for dynamic broadcasting.

The complexity of implementing other fundamental communication primitives under the proposed local dynamic model, such as aggregation, local broadcast, and multiple-message broadcast deserves further investigation. Furthermore, it is also significant to re-visit our algorithm in a jamming scenario, since jamming is a common phenomenon in wireless networks and our algorithm currently relies on reliable communication channels.

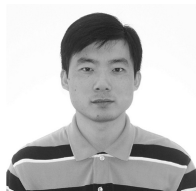
ACKNOWLEDGMENT

This work is partially supported by National Key R&D Program of China with grant No. 2019YFB2102600 and NSFC (Nos. 61971269, 61832012, 61672321, 61771289, 61702304).

REFERENCES

- [1] M. Ahmadi, A. Ghodselahi, F. Kuhn, and A.R. Molla. The Cost of Global Broadcast in Dynamic Radio Networks. In *PODC*, 2013.
- [2] R. Bar-Yehuda, O. Goldreich, and A. Itai. On the Time-Complexity of Broadcast in Multi-hop Radio Networks: An Exponential Gap Between Determinism and Randomization. In *J. Comput. Syst. Sci.*, 45(1): 104-126, 1992.
- [3] M.H.L. Bodlaender, M. Halldórsson, and P. Mitra. Connectivity and aggregation in multihop wireless networks. In *PODC*, 2013.
- [4] B.S. Chlebus, D.R. Kowalski, and M. Strojnowski. Fast scalable deterministic consensus for crash failures. In *PODC*, 2009.
- [5] F. Klingler, F. Dressler, and C. Sommer. The Impact of Head of Line Blocking in Highly Dynamic WLANs. In *TVT*, 2018.
- [6] A.E.F. Clementi, A. Monti, F. Pasquale, and R. Silvestri. Broadcasting in dynamic radio networks. In *J. Comput. Syst. Sci.*, 75(4): 213-230, 2009.
- [7] A.E.F. Clementi, A. Monti, and R. Silvestri. Round Robin is optimal for fault-tolerant broadcasting on wireless networks. In *J. Parallel Distrib. Comput.*, 64(1): 89-96, 2004.
- [8] A. Cornejo, S. Gilbert, and C.C. Newport. Aggregation in dynamic networks. In *PODC*, 2012.
- [9] A. Czumaj, and W. Rytter. Broadcasting Algorithms in Radio Networks with Unknown Topology. In *FOCS*, 2003.
- [10] S. Daum, S. Gilbert, F. Kuhn, and C.C. Newport. Broadcast in the Ad Hoc SINR Model. In *DISC*, 2013.

- [11] M. Dinitz, J.T. Fineman, S. Gilbert, and C.C. Newport. Smoothed Analysis of Dynamic Networks. In *DISC*, 2015.
- [12] J.T. Fineman, S. Gilbert, F. Kuhn, C.C. Newport. Contention Resolution on a Fading Channel. In *PODC*, 2016.
- [13] M. Ghaffari, B. Haeupler, and M. Khabbazian. Randomized broadcast in radio networks with collision detection. In *PODC*, 2013.
- [14] O. Goussevskaia, T. Moscibroda, and R. Wattenhofer. Local broadcasting in the physical interference model. In *DIALM-POMC*, 2008.
- [15] T. Jurdzinski, D.R. Kowalski, M. Rozanski, and G. Stachowiak. On the impact of geometry on ad hoc communication in wireless networks. In *PODC*, 2014.
- [16] T. Jurdzinski, D.R. Kowalski, M. Rozanski, and G. Stachowiak. On setting-up asynchronous ad hoc wireless networks. In *INFOCOM*, 2015.
- [17] T. Jurdzinski, D.R. Kowalski, and G. Stachowiak. Distributed Deterministic Broadcasting in Uniform-Power Ad Hoc Wireless Networks. In *FCT*, 2013.
- [18] T. Jurdzinski, D.R. Kowalski, and G. Stachowiak. Distributed Deterministic Broadcasting in Wireless Networks of Weak Devices. In *ICALP*, 2013.
- [19] D.R. Kowalski and A. Pelc. Deterministic broadcasting time in radio networks of unknown topology. In *FOCS*, 2002.
- [20] D.R. Kowalski and A. Pelc. Broadcasting in undirected ad hoc radio networks. In *Distributed Computing*, 18(1): 43-57, 2005.
- [21] F. Kuhn, N.A. Lynch, and C.C. Newport. Brief announcement: hardness of broadcasting in wireless networks with unreliable communication. In *PODC*, 2009.
- [22] F. Kuhn, N.A. Lynch, C.C. Newport, R. Oshman, and A. Richa. Broadcasting in unreliable radio networks. In *PODC*, 2010.
- [23] F. Kuhn, N.A. Lynch, and R. Oshman. Distributed computation in dynamic networks. In *STOC*, 2010.
- [24] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Initializing newly deployed ad hoc and sensor networks. In *MOBICOM*, 2004.
- [25] F. Kuhn and R. Oshman. Dynamic networks: models and algorithms. In *SIGACT News*, 42(1): 82-96, 2011.
- [26] J. Schneider and R. Wattenhofer. Coloring Unstructured Wireless Multi-hop Networks. In *PODC*, 2009.
- [27] A. Sinha, L. Tassioulas, and E. Modiano. Throughput-optimal broadcast in wireless networks with dynamic topology. In *MobiHoc*, 2016.
- [28] X. Tian, J. Yu, L. Ma, G. Li, and X. Cheng. Distributed deterministic broadcasting algorithms under the SINR model. In *INFOCOM*, 2016.
- [29] D. Yu, Q. Hua, Y. Wang, F.C.M. Lau. An $O(\log n)$ Distributed Approximation Algorithm for Local Broadcasting in Unstructured Wireless Networks. In *DCOSS* 2012.
- [30] D. Yu, Q. Hua, Y. Wang, H. Tan, and F.C.M. Lau. Distributed multiple-message broadcast in wireless ad hoc networks under the SINR model. In *Theor. Comput. Sci.*, 610: 182-191, 2016.
- [31] D. Yu, L. Ning, Y. Zou, J. Yu, X. Cheng, F.C.M. Lau. Distributed Spanner Construction With Physical Interference: Constant Stretch and Linear Sparseness. In *IEEE/ACM Trans. Netw.*, 25(4): 2138-2151, 2017.
- [32] D. Yu, Y. Wang, T. Tonoyan, and M.M. Halldórsson. Dynamic Adaptation in Wireless Networks Under Comprehensive Interference via Carrier Sense. In *PDPS*, 2017.
- [33] D. Yu, Y. Zou, J. Yu, X. Cheng, Q. Hua, H. Jin, F.C.M. Lau. Stable Local Broadcast in Multihop Wireless Networks Under SINR. In *IEEE/ACM Trans. Netw.*, 26(3): 1278-1291, 2018.
- [34] Y. Zou, D. Yu, L. Wu, J. Yu, Y. Wu, Q. Hua, F.C.M. Lau. Fast Distributed Backbone Construction Despite Strong Adversarial Jamming. In *INFOCOM*, 2019.



distributed computing and graph algorithms.

Dongxiao Yu received the BSc degree in 2006 from the School of Mathematics, Shandong University and the PhD degree in 2014 from the Department of Computer Science, The University of Hong Kong. He became an associate professor in the School of Computer Science and Technology, Huazhong University of Science and Technology, in 2016. He is currently a professor in the School of Computer Science and Technology, Shandong University. His research interests include wireless networks, distributed computing and graph algorithms.



Yifei Zou received the B.E. degree in 2016 from Computer School, Wuhan University. He is currently a PhD student in Department of Computer Science, The University of Hong Kong. His research interests include wireless networks, ad hoc networks and distributed computing.



include privacy-aware computing, wireless networking, distributed algorithms, peer-to-peer computing, and graph theory. Particularly, he is interested in designing and analyzing algorithms for many computationally hard problems in networks. He is a senior member of IEEE, a member of ACM and a senior member of the CCF (China Computer Federation).

Jiguo Yu received his Ph.D. degree in School of mathematics from Shandong University in 2004. He became a full professor in the School of Computer Science, Qufu Normal University, Shandong, China in 2007. Currently he is a full professor in Qilu University of Technology (Shandong Academy of Sciences), Shandong Computer Science Center (National Supercomputer Center in Jinan), and a professor in School of Information Science and Engineering, Qufu Normal University. His main research interests



University of Technology, China. His research interests include network virtualization, internet of things and blockchain technology.

Yu Wu received his B.E and M.E. degrees in 2006 and 2009 respectively, from Department of Computer Science and Technology, Tsinghua University, China, and his Ph.D. degree in 2013 from the Department of Computer Science, The University of Hong Kong. He was a postdoctoral scholar in the School of Electrical, Computer and Energy Engineering (ECEE), ASU, from Nov 2013 to Oct 2015. Yu Wu is currently an Associate Professor in the School of Computer Science and Network Security in the Dongguan

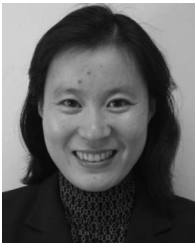


Weifeng Lv received his Ph.D. degree in computer science from Beihang University in 1998. His current research interests include massive information system, urban cognitive computing, swarm intelligence, and smart cities. He is a professor of computer science and the dean of the School of Computer Science and Engineering, as well as the vice director of the State Key Laboratory of Software Development Environment, at Beihang University. He also serves as the Vice-President and Secretary-General of

the China Software Industry Association and the director of National Engineering Research Center for Science and Technology Resources Sharing Service. He received multiple internationally renowned awards, including the second prize of the 2016 China National Science and Technology Invention Award and the first prize of the 2010 Beijing Science and Technology Award.



Francis C.M. Lau received the PhD degree in computer science from the University of Waterloo. He is currently a professor in computer science at The University of Hong Kong. He is the editor-in-chief of the Journal of Interconnection Networks. His research interests include computer systems, networks, programming languages, and application of computing in arts.



Xiuzhen Cheng received her M.S. and Ph.D. degrees in computer science from the University of Minnesota – Twin Cities in 2000 and 2002, respectively. She is a professor in the School of Computer Science and Technology, Shandong University. Her current research interests include cyber physical systems, wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis. She has served on the editorial boards of several technical journals and the technical

program committees of various professional conferences/workshops. She also has chaired several international conferences. She worked as a program director for the US National Science Foundation (NSF) from April to October in 2006 (full time), and from April 2008 to May 2010 (part time). She received the NSF CAREER Award in 2004. She is Fellow of IEEE and a member of ACM.



Falko Dressler received his M.S. and Ph.D. degrees in computer science from the University of Erlangen in 1998 and 2003, respectively. He is a full professor of computer science and chair for Distributed Embedded Systems at the Heinz Nixdorf Institute and the Dept. of Computer Science, Paderborn University. His research objectives include adaptive wireless networking, self-organization techniques, and embedded system design with applications in ad hoc and sensor networks, vehicular networks, industrial wireless

networks, and nano-networking. Dr. Dressler is associate editor-in-chief for Elsevier Computer Communications as well as an editor for journals such as IEEE Trans. on Mobile Computing, IEEE Trans. on Network Science and Engineering, Elsevier Ad Hoc Networks, and Elsevier Nano Communication Networks. He has been chairing conferences such as IEEE INFOCOM, ACM MobiSys, ACM MobiHoc, IEEE VNC, and IEEE GLOBECOM. He authored the textbooks Self-Organization in Sensor and Actor Networks published by Wiley & Sons and Vehicular Networking published by Cambridge University Press. He has been an IEEE Distinguished Lecturer as well as an ACM Distinguished Speaker. He is Fellow of IEEE and Distinguished Member of ACM.