Implementing Abstract MAC Layer in Dynamic Networks

Dongxiao Yu, *Member, IEEE,* Yifei Zou, Jiguo Yu, *Senior Member, IEEE,* Yong Zhang *Senior Member, IEEE*, Feng Li, Xiuzhen Cheng, *Fellow, IEEE,* Falko Dressler, *Fellow, IEEE,* Francis C.M. Lau

Abstract—Dynamicity is one of the most challenging, yet, central aspects of wireless networks. Dynamicity can come in many guises, such as churn (node insertion/deletion) and node mobility. Although the study of dynamic networks has been popular in distributed computing domains, previous works considered only partial factors causing dynamicity. In this work, we propose a comprehensive dynamic model that includes crucial dynamic factors on nodes and links. Our model defines dynamicity in terms of localized topological changes in the vicinity of each node, rather than for the whole network globally. Obviously, a localized dynamic model suits distributed algorithm studies better than a global one. The proposed dynamic model uses the more realistic SINR model to describe wireless interference, instead of the oversimplified graph-based models adopted in most existing works. Under the proposed dynamic model, we provide an efficient distributed algorithm accomplishing local broadcast services in the abstract MAC layer that was first presented by Kuhn et al. [24]. Our solution paves the road for many new fast algorithms for solving high-level problems in dynamic networks, such as consensus, single-message broadcast, and multiple-message broadcast. Extensive simulations show that our algorithm exhibits good performance in realistic environments with dynamic behaviors.

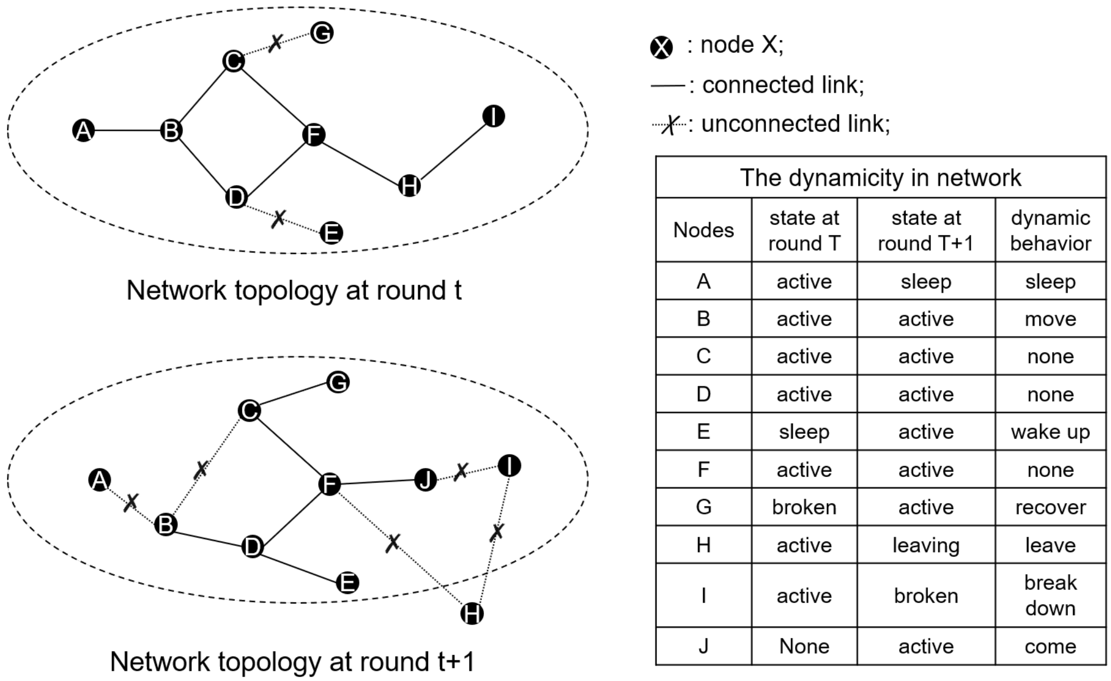
Index Terms—Abstract MAC layer, dynamic wireless network, SINR model, distributed algorithms

1 Introduction

Wireless networks are ubiquitous and are getting increasingly more so as Internet-of-Things is becoming more and more a definite reality. Dynamicity, or variability with time, is a natural phenomenon and property of wireless networks. Dynamicity comes from many reasons, among which the main ones include churn and node mobility. Churn, i.e., nodes coming and leaving, can be caused by failures or intermittent participation in network tasks; mobility is the predominant mode for wireless networks nowadays (think of cell phones), which also changes the network topology. An example about how churns and mobility of nodes impact the network topology is given in Fig. 1. As a network becomes large and has to be operated in a decentralized fashion, which is typical in the context of Internet-of-Things, the study of dynamic networks becomes an important one in the distributed computing domain. Notwithstanding, existing works usually just pick on one particular factor among multiple crucial ones causing dynamicity, such as churn [5], [16], [27], [35], [43], or link changes due to node mobility [6], [23], [26]. However, as just mentioned, the dynamicity of wireless networks comes in many guises, and focusing just on one of multiple dynamic factors may produce algorithms that perform dramatically differently in real situations from theoretical analysis. Hence, it is important that a comprehensive model that incorporates crucial dynamic factors is used when designing the needed efficient algorithms.

In this work, we provide efficient distributed algorithms for implementing local broadcast primitives defined in the abstract MAC (absMAC) layer under a comprehensive dynamic network model. The concept of abstract MAC layer was first proposed by Kuhn et al. [24], which expresses key guarantees of real MAC layers with respect to the local broadcast operation. These guarantees include two message delivery latency bounds: the *acknowledgement* bound f_{ack}, which is the time for a sender's message to be received by all its neighbors, and the *progress* bound f_{prog}, which is the time for a receiver to receive one message when there is at least one neighbor sending. The absMAC layer divides wireless algorithm design and analysis into two independent and manageable components, i.e., to implement the absMAC layer over a physical network and to solve higher-level problems based on the local broadcast services and time guarantees provided by the absMAC layer. The approach of abstract MAC layer can help solve the problems of algorithm design and analysis, which are extremely complicated when considering issues of message dissemination at high levels together with contention management at the physical level. Benefiting from the absMAC layer approach, many new efficient algorithms have been developed for certain fundamental problems, including consensus [33], single-message broadcast, and multiple-message broadcast [15],

- *D. Yu F. Li and X. Cheng are with School of Computer Science and Technology, Shandong University, Qingdao, 266510, P.R. China. E-mail: {dxyu,fli,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: jiguoyu@sina.com.
Y. Zhang is with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, P.R. China. E-mail: zhangyong@siat.ac.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.



The dynamicity in network			
Nodes	state at round T	state at round T+1	dynamic behavior
A	active	sleep	sleep
B	active	active	move
C	active	active	none
D	active	active	none
E	sleep	active	wake up
F	active	active	none
G	broken	active	recover
H	active	leaving	leave
I	active	broken	break down
J	None	active	come

Fig. 1: Dynamicity in network. In this figure, the churns of nodes include nodes sleeping, waking up, joining, leaving, breaking down, and recovering from breaking down. Together with the mobility, churn greatly impacts the connection of links and the topology of the network in each round.

TABLE 1: CS is the task of consensus and D is the diameter; SMB and MMB are the tasks of single and multiple message broadcast, respectively, and k is the number of messages.

Task	Time bound
CS	$O(D * f_{ack})$ [17]
SMB	$O((D + \log n)f_{prog})$ [20]
MMB	$O(kf_{ack} + (D + k(\log n + \log k))f_{prog})$ [20]

[17], [20], as shown in Table 1. Also, it is very likely for the results in [2], [3], [4], [18], [29] to be renewed if applying an efficient dynamic abstract MAC layer. Although there have been many algorithms proposed to implement the local broadcast primitives in the abstract MAC layer, it is unknown whether they still work efficiently in dynamic networks. For example, in [17], an efficient implementation algorithm for the abstract MAC layer was proposed. However, this implementation relies heavily on the computation of a special node set, which will be ruined by the dynamicity of nodes. So, the implementation algorithm may not perform well if we directly apply the protocol of [17] in dynamic networks. The algorithm given in [31] can tolerant unreliable links to some extent, but its performance is unpredictable when facing other dynamic behaviors as considered in this work.

Model and Our Results. The basis of a wireless network model is its modeling of signal propagation and reception. We construct our dynamic model based on the Signal-to-Interference-plus-Noise (SINR) model. The SINR model defines signal (as well as interference) fading with distance which is moderated by a path-loss exponent, and hence the message reception is not determined by nearby nodes but by all simultaneously transmitting nodes in the network. Comparing with graph-based models that simplify interference to be a binary and local phenomenon, the SINR model accurately reflects the most important features of wireless interference, fading, and accumulation. Hence, the SINR model is getting increasingly popular despite it posing a great challenge for distributed algorithm design and analysis because of its global definition of interference.

We present a localized dynamic model that incorporates the dynamic factors in networks. In other words, the model reflects the impact of dynamic factors within a locality surrounding every node. Because the SINR model defines the signal (as well as interference) that fades with the distance to some path-loss exponent, when depicting local network changes, the model not only needs to reflect the changes in the neighborhood of each particular node, but also to describe the distance changes between the node and its neighbors. In addition, it should define the local network topology changes of the nodes as independently as possible, in order to make the analysis of the algorithm performance more manageable.

The localized dynamic network model we propose obeys all the above considerations. It treats the network area as a grid (of cells), which ensures that any pair of nodes in a cell are neighbors (within the transmission range of each other). In each cell, the local topology of the nodes is defined based on the distance of each node from its nearest neighbor in the cell. The impact of churn (node insertion/deletion) and mobility of nodes is then depicted by the magnitude of change (called dynamic rate) on the local topology of each cell due to these two factors.

Our main contribution is a randomized distributed algorithm for implementing the abstract MAC layer, with the acknowledgement time bound $f_{ack} = |I|$, where I is the interval from the beginning (round 0) to a round t such that $(\Delta^g(I)+1)*T_P \le t$ holds for all non-empty cells g ($\Delta^g(I)$ is the number of active nodes in cell g during I; $T_P = O(\log n + \log R_c)$ is the running time of each phase in our algorithm, where R_c is the required local broadcast range); and progress time bound $f_{prog} = O(\log n + \log R_c)$. In static networks, our results imply an acknowledgement time bound of $f_{ack} = O(\Delta(\log n + \log R_c))$ (Δ is the maximum degree) and $f_{prog} = O(\log n + \log R_c)$. By the lower bounds $\Omega(\Delta)$ and $\Omega(\log n)$ for f_{ack} and f_{prog} respectively given in [41], our algorithms attain an acknowledgement bound inferior to the best solution by a logarithmic factor and they attain an asymptotically optimal progress bound.

Our algorithm assumes a reasonable restriction of constant dynamic rate (cf. Section 3). The performance is guaranteed with high probability, i.e., with probability $1-n^{-c}$ for some constant $c>0$. We also show that the constraint of constant dynamic rate is necessary to get asymptotically optimal algorithms, if a leader election procedure is involved.

Roadmap The remaining part of the paper is organized as follows. We present the related work in Section 2. The network model and problem definitions are then given in Section 3. In Section 4, we present a leader election algorithm, which is used as a subroutine in the implementation algorithm. In Section 5, the implementation algorithm is proposed. The necessity of the restriction on the dynamic rate is discussed in Section 6. Section 7 shows the simulation and its results, and Section 8 concludes the paper.

2 Related Work

Dynamic Wireless Network. Distributed algorithm design and analysis has been a hot topic in the field of wireless computing, due to the increasing popularity of large-scale mobile wireless networks. Many dynamic models have been proposed to reflect the dynamicity in wireless networks. In [27], Kuhn et al. proposed the unstructured model, to describe the nodes' insertions under a unit disk setting. Later, this model was extended to the SINR model and bounded independence graphs in [16] and [35], respectively. The node crash failures were considered in [5]. 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 [6], [23]. It defines two graphs on the same node set, one consisting of reliable links, and the other of unreliable links. This model extends the radio network model to the dynamic case. The T-interval connectivity model given in [26] 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 [7], [12] 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 [37]. In this model, the network configuration follows a stationary ergodic process with a stationary distribution, and in each slot the network controller can only activate a set of non-interfering links. More recently, Yu et al. [39] 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 was given in [28].

Abstract MAC Layer. The absMAC layer was proposed by Kuhn et al. in [24], [25]. Thereafter, several variants of the basic absMAC layer model have been proposed for different deployment scenarios, such as the conditional absMAC layer [11], the enhanced absMAC layer [15] and the probabilistic absMAC layer [20], [21]. Similar to PHY/MAC approaches on the upper-layers operations in different scenario [10], [13], [30], the abstract MAC layer can also support many higher-level operations. Specifically, based on the abstraction of the absMAC layer, several fundamental problems have been studied and efficient algorithms were proposed, including Neighbor Discovery [8], [9], Single-Message Broadcast, Multiple-Message Broadcast [15], [17], [20], [21], leader election [32], and Consensus [33], [34].

For the implementation of absMAC layers, basic implementations of a probabilistic absMAC layer were given by Khabbazian et al. [20] using the classical decay strategy and [22] using Analog Network Coding. All these implementation algorithms are devised under the graph-based models, where interference is oversimplified as a local and binary phenomenon. Halldórsson et al. first studied the approximate implementation of the probabilistic absMAC layer under the SINR model in [17], and the exact implementation was studied by Yu et al. in [41]. But these two results are both for static networks. The only known implementation algorithm under dynamic networks was proposed by Lynch and Newport in [31], which is designed in a dual graph model, based on a graph-based interference definition and just considering unreliable links.

In summary, all of the implementations on abstract MAC layer mentioned above except [31] are considered in a network without any dynamicity, and the abstract MAC layer in [31] is implemented in a very simple dynamic model that just takes into account unreliable links. Thus, the performance of the above protocols in a real dynamic network is unknown, while our work here implements the abstract MAC layer in a comprehensive model.

3 Model and Definition

We consider a network in 2-dimensional Euclidean space, where n nodes are deployed arbitrarily. The time is divided into rounds. A round may contain a constant number of slots, and each slot can be the time unit for nodes to send a message. Synchronous communications are assumed. Each node is equipped with a half-duplex transceiver, i.e., in each time slot, the node can transmit or listen but cannot do both. We assume that the nodes use the same transmission power P, which is known as the *uniform* power assignment. In each round, the network topology may change due to nodes' joining/leaving and mobility. Let n denote the upper bound of nodes in any round. Each node has a unique identifier ID_v. For any two nodes u and v, let $d(u,v)$ be the Euclidean distance between them.

Communication Model. Nodes transmit on a shared channel and the interference between simultaneous transmissions is depicted by the SINR model. For a transmitting node u and a receiving node v, let $S_{u,v}$ be the strength of

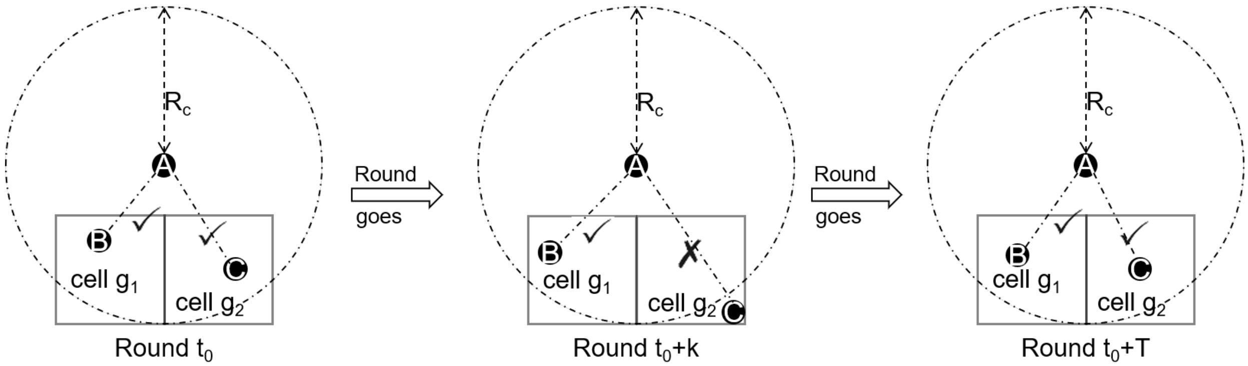


Fig. 2: Stable connection. For example, nodes B and C move around A in the interval from round t_0 to round $t_0 + T$. Since node B always moves within A's transmission range, node A always has a stable connection with cell g_1, i.e., $d(v, g_1) \leq R_c$; meanwhile, at round $t_0 + k$, node C moves out of A's transmission range. Thus, for this example, we say node A has a T-stable connection with cell g_1, and does not have a T-stable connection with cell g_2.

the signal from u and received by v. We use the Rayleigh-fading pattern to depict the uncertainty in signal reception. Specifically, $S_{u,v}$ is a random variable that is exponentially distributed with mean $\bar{S}_{uv} = P_u/d_{uv}^{\alpha}$, where P_u is the transmission power and α is the path-loss exponent whose value is usually between 2 and 6. Let W be the set of nodes transmitting simultaneously with u, but not including u itself, $\mathcal{I}_W(v)$ be the sum of the interference at v caused by nodes in set W, $SINR(v, u, W)$ be the SINR rate at node v for transmitting node u, then

$$\mathcal{I}_W(v) = \sum_{w \in W} S_{w,v}, \quad SINR(v, u, S) = \frac{S_{u,v}}{N + \mathcal{I}_S(v)} \qquad (1)$$

where N is the ambient noise. The SINR model defines that v can receive the message of u if $SINR(v, u, S) \geq \beta$, where $\beta > 1$ is a hardware-determined constant parameter.

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 ($SINR(u, v, S) \geq \beta$) when there are no other simultaneous transmissions on the same channel. From the SINR condition (1), $R_T = (P/\beta N)^{1/\alpha}$.

Since nodes very close to R_T can only communicate if other transmissions in the network are sufficiently far away, the standard assumption is to assume that it suffices to communicate using a smaller range, $R_c = (1 - \epsilon)R_T$ [1], [19], [38], [40], [42], [44], where $\epsilon \in (0, 1)$ is a fixed model parameter, such that the communication can tolerate some amount of interference. We call R_c the communication range.[1] For two nodes u and v, we say they are R_c-neighbors if $d(u, v) \leq R_c$.

Abstract MAC Layer. Since we focus on randomized solutions, the probabilistic absMAC layer model [17], [20] is adopted here, which provides an interface to higher layer with input $bcast(m)_i$ and outputs $ack(m)_i$, $rcv(m)_i$ for any node $i \in V$ and message $m \in M$. When a node $u \in V$ broadcasts a message m, the model delivers the message to all its R_c-neighbors. If u successfully performs a local broadcast, the abstract MAC layer returns an acknowledgement $ack(m)_u$ to higher layer informing that the broadcasting of u is completed. Similarly, it returns $rcv(m)_v$ to the higher layer when v receives message m from a R_c-neighbor. The model provides two time bounds, the *acknowledgement bound* f_{ack} and the *progress bound* f_{prog}. In particular, the acknowledgement bound guarantees each node's local broadcast can be successfully performed within f_{ack} time. The progress bound bounds the time for a node to receive a message when there is at least one neighbor sending. For further details about the absMAC layer and motivations for these delay bounds, please refer to [15], [20], [24].

In the probabilistic absMAC layer, two parameters ξ_{prog} and ξ_{ack} are defined to indicate the error probabilities for satisfying the delay bounds f_{prog} and f_{ack}, respectively. In this paper, we require that the progress and acknowledgement primitives be accomplished with high probability, i.e., $\xi_{prog}, \xi_{ack} \in n^{-c}$ for some constant $c > 0$.

Dynamicity. As mentioned above, we use a local view to depict churns and mobility of nodes, by dividing the network into a grid and describing the change of the local network topology in each cell. Next, we describe the dynamic model in more detail.

The 2-dimensional network area is modeled as a grid $\mathcal{G}$ consisting of square cells each of size $\frac{\epsilon R_c}{\sqrt{2}} \times \frac{\epsilon R_c}{\sqrt{2}}$ for some specified constant ϵ. 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. Assuming that point $(0, 0)$ is the grid origin, a cell is given coordinates (i, j) and denoted as $g(i, j)$ when its bottom left corner is located at $(\frac{\epsilon R_c}{\sqrt{2}} * i, \frac{\epsilon R_c}{\sqrt{2}} * j)$ for $(i, j) \in \mathbb{Z}^2$. A node v with position (x, y) on the network is located at cell $g(i, j)$ only when $i * \frac{\epsilon R_c}{\sqrt{2}} \leq x < (i + 1) * \frac{\epsilon R_c}{\sqrt{2}}$ and $j * \frac{\epsilon R_c}{\sqrt{2}} \leq y < (j + 1) * \frac{\epsilon R_c}{\sqrt{2}}$.

Since in the SINR model, signal fades with distance, and a nearby neighbor matters more than those further away, we define the local network topology in a cell according to the nodes' distances from their nearest neighbors. Consider the local network of cell g in a fixed slot, nodes in cell g are divided into classes $\{V_i^g : i = 0, 1, \ldots, \log \epsilon R_c\}$. More

1. Note that R_c may not be a constant anymore after we normalize the minimum distance between nodes to 1. But because usually the rate of the maximum distance and the minimum distance between nodes cannot be exponentially large, R_c can be set to be bounded by $poly(n)$, and hence $\log R_c \in O(\log n)$.

specifically, 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 $0 \le i \le \log \epsilon R_c - 1$ if $d(u,v) \in [2^i, 2^{i+1})$. If v is the only node in cell g, v is in class $V_{\log \epsilon R_c}^g$. *Node churns and mobility.* We consider both node churns (node arrivals/departures) and node mobility (movement inside a cell or from one cell to another cell). It is assumed that network change is round-based, i.e., the network changes at the beginning of each round and remains unchanged during the round.

A *dynamic rate* is defined to quantify the change of network topology. In a round t, a cell g is called a non-empty one, if it contains active nodes (those that are participating in the algorithm execution). Considering a period of round I, for $i \in \{0, 1, \ldots, \log \epsilon R_c\}$ and $t \in I$, let $V_i^g(t)$ and $\hat{V}_i^g(t)$ denote the set of active nodes in class V_i^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 *dynamic rate* λ is defined as

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

It can be seen that as the dynamic rate changes, our dynamic model can model different extents of dynamicity caused by churns and node mobility.

For an interval I from round t to t', define $\Delta^g(I) = |\cup_{r=t}^{t'} \cup_{i=0}^{\log \epsilon R_c} V_i^g(r)|$, i.e. $\Delta^g(I)$ is the number of active nodes in cell g during I. We will use this parameter to bound the acknowledgement time in our abstract MAC layer implementation algorithm.

Stable connection. To gurantee a node can receive messages from neighbors, we have to give some constraints on the connection between it and the active nodes (that are excuting the algorithm and have messages to disseminate). We restrict the stable connection between a node and a non-empty cell. Specifically, the distance between a cell g and a node v is defined as the minimum distance between nodes in g and v, i.e., $d(v,g) = \min_{w \in g} d(w,v)$. v is connected to cell g if $d(v,g) \le R_c$. Then we say a node v is T'-stablely connected to non-empty cells if in T' consecutive rounds, there is always a cell connected with v. Obviously, if v does not connect to non-empty cells for a long enough time, it is hard to ensure successful message reception at v. Based on the dynamic model, we only need to consider the case of $T' \in \Omega(\log n)$, as $\Omega(\log n)$ is the minimum time needed for two nodes to communicate successfully with high probability [36], even if without interference and dynamicity. An example about the stable connection is given in Fig. 2.

Knowledge and Capability of Node Each node has the values of n, R_c, N, and the SINR parameters α, β. The nodes can acquire location information by some services, such as GPS. But physical carrier sensing is not needed, i.e., nodes know nothing about the transmissions on the channel when it receives no message.

4 Leader election algorithm

In this section, we present a leader election algorithm to elect a leader for each non-empty cell, which will be used as a subroutine in our abstract MAC layer implementation algorithm. Even though the proposed leader election algorithm looks simple, its theoretical analysis is not. It will be shown that the algorithm can accomplish leader election in $O(\log n + \log R_c)$ rounds in the dynamic setting. Noting that $\Omega(\log n)$ is the lower bound for successful a message transmission with a high probability guarantee, our leader election algorithm is nearly optimal.

TABLE 2: Acronyms in model and algorithms

Acronym	Explanation	Acronym	Explanation
ack.	acknowledgement	prog.	progress
w.h.p.	with high probability	absMAC	abstract MAC
LE	Leader election	LB	Leader broadcast
TDMA	Time division multiple access		

0	1	...	c-1	0	1	...
c	c+1	...	2c-1	c	c+1	...
...	...	...	...	...	...	...
(c-1)c	...	...	c*c-1	(c-1)c	...	...
0	1	...	c-1	0	1	...
c	c+1	...	2c-1	c	c+1	...
...	...	...	...	...	...	...

Fig. 3: Coloring of cells

4.1 Algorithm Description

The leader election algorithm is given in Algorithm 1. The algorithm is very simple, but the analysis is non-trivial.

Before the algorithm execution, the cells are colored as follows: the cell g(i,j) gets the color $c * (i \mod c) + (j \mod c)$, where constant $c = \lceil ((\frac{\beta(32\frac{\alpha-1}{\alpha-2}+4)}{(1+\epsilon)^{-\alpha}-(1-\epsilon)^{\alpha}})^{\frac{1}{\alpha}} + 1 + \epsilon) * \frac{\sqrt{2}}{\epsilon} + 1\rceil$. It is easy to see that this coloring uses $c * c$ colors, as demonstrated in Fig. 3. For each node v, it has the same color as the cell it is in. The coloring generates a TDMA scheme for the algorithm's execution in each round and will be introduced later. The whole algorithm consists of $\Theta(\log n + \log R_c)$ rounds, each of which has $c * c$ slots. In particular, the active nodes of color j for $0 \le j \le c^2 - 1$ execute the algorithm in the j-th slot in each round. This TDMA scheme can help avoid interference from nodes in nearby cells. We next focus on the algorithm execution for nodes of a particular color.

In the algorithm, nodes can be in three states: state $\mathbb{A}$ means that the node is active for leader election; state $\mathbb{S}$ means that the node gives up becoming a leader and will remain silent in subsequent rounds; state $\mathbb{L}$ means that the node has become a leader. At the beginning, all nodes are in state $\mathbb{A}$. For parameters in Algorithm 1, $T_{LE} = k(\log n + \log R_c)$ and $p = (1 - (1-\epsilon)^{\alpha})(1 - 2^{1-\alpha/2})/(3 * 2^{\alpha+7}\beta)$,

Algorithm 1: $LE(n, R_c)$

```
Initialization: state_v = 𝔸;
-------------------------------------------
Each node v in color j does:
for T_LE rounds do
    slot = 0;
    for slot < c * c do
        if state_v = 𝔸 and slot = j then
            transmit with probability p;
            if receive a message from nodes in same cell then
                state_v = 𝕊;
        slot++;
if state_v = 𝔸 then
    state_v = 𝕃
```

TABLE 3: Parameters in algorithms and analysis

Parameter	Value	Parameter	Value
ϵ	$constant \in (0,1)$	c_1	$\frac{1-(1-\epsilon)^{\alpha}}{2^{\alpha+1}\beta}$
ε	$\frac{1-(2^{1-\alpha/2})}{2}$	c_{max}	$\frac{48}{1-2^{1-\alpha/2}}$
γ	$\frac{p(1-p)(e-1)}{8*(2s+5)^2e^2}$	γ_1	$1-\gamma$
τ	$\gamma_2/\gamma_1 > 1$		
p	$\frac{1-(1-\epsilon)^{\alpha}}{3*2^{\alpha+7}\beta}(1-2^{1-\frac{\alpha}{2}})$		
λ	$\frac{\gamma_1+\rho/(1-\rho)}{1-\rho}^{-\frac{1}{4\varsigma}}-1>0$		
c	$\lceil((\frac{\beta(32\frac{\alpha-1}{\alpha-2}+4)}{(1+\epsilon)^{-\alpha}-(1-\epsilon)^{\alpha}})^{\frac{1}{\alpha}}+1+\epsilon)*\frac{\sqrt{2}}{\epsilon}+1\rceil$		
s	$[\frac{(1-2^{1-\alpha/2})(1-(1-\epsilon)^{\alpha})}{3\cdot 2^{\alpha+5}\cdot\beta}]^{\frac{1}{1-\alpha/2}}$		
γ_2	$(\gamma_1+\rho/(1-\rho))(1+\lambda)^{2\varsigma}<1$		
ς	$\lceil\max\{(4\tau)/(a_1(1-\gamma_2)),4\tau\}\rceil$		
n	unknown but sufficiently large		
k	sufficiently large constant		
ρ	sufficiently small constant		
a_1	the constant hidden behind the Ω notation in the probability guarantee in Lemma 1		

where the constant k is set to be sufficiently large for the convenience of analysis.

In the corresponding slot of each round: active nodes transmit with probability p to compete for leadership or listen otherwise; When an active node receives a message from nodes in the same cell, it gives up the leader competition and joins state $\mathbb{S}$. After T_{LE} rounds, the nodes that are still active become the leaders of corresponding cells and join state $\mathbb{L}$.

Notice that a similar algorithmic idea is used in [14] to solve the leader election problem in a static single-hop network. But the analysis of our algorithm is different from that in [14], as we have to handle two factors that heavily affect transmissions and do not exist in static single-hop networks, including multi-hop interference, and dynamicity due to node churns and mobility.

We next show that under our dynamic setting and for any dynamic rate $\lambda < \frac{\gamma_1+\rho/(1-\rho)}{1-\rho}^{-\frac{1}{4\varsigma}} - 1$, where γ_1, ρ, and ς are constants given in Table 3. Algorithm 1 can correctly elect exactly one leader for each non-empty cell in $O(\log n + \log R_c)$ rounds w.h.p.

4.2 Algorithm Analysis

In the algorithm, the TDMA scheme makes active nodes of the same color execute the algorithm together. Hence, we next focus on the algorithm execution of nodes in cells of a fixed color $j \in [0, c*c-1]$. Here, let $color(g)$ be the color of cell g and $V_i(t) = \cup_{g\in\mathcal{G},color(g)=j} V_i^g(t)$, $\hat{V}_i(t) = \cup_{g\in\mathcal{G},color(g)=j} \hat{V}_i^g(t)$ and $n_i(t) = |V_i(t)|$, $\hat{n}_i(t) = |\hat{V}_i(t)|$.

Obviously, a leader will be elected in each non-empty cell g when all $\hat{V}_i$ for $i \in \{0, 1, \ldots, \log\epsilon R_c - 1\}$ are reduced to empty, since when this happens, there is exactly one active node left in each non-empty cell g that belongs to class $V_{\log\epsilon R_c}$.

Our analysis consists of two steps: we first prove that in each round r, a constant fraction of active nodes in network will become inactive with some probability; We then bound the time needed for the event that only one active node is left in each non-empty cell g to happen. For $i \in \{0, 1, \ldots, \log\epsilon R_c\}$, we use $V_{<i}(r)$ and $\hat{V}_{<i}(r)$ to denote the sets of active nodes in in classes V_js for $j < i$ at the beginning and the end of a round r. $n_{<i}(r)$ and $\hat{n}_{<i}(r)$ are defined correspondingly. Table 3 is used to detailedly present the values of notations in algorithm and analysis. The result for the first step is given in the following Lemma 1.

Lemma 1. For any round r, $i \in \{0, 1, \ldots, \log\epsilon R_c - 1\}$, if $n_{<i}(r) \le \varepsilon n_i(r)$, then γ fraction of nodes in V_i will become inactive with probability $1 - e^{\Omega(|V_i|)}$ at the end of round r, where $\varepsilon = \frac{1-(2^{1-\alpha/2})}{2}$, $\gamma = \frac{p(1-p)*e^{-1}*(1-e^{-1})}{8*(2s+5)^2}$, $s = [\frac{(1-2^{1-\alpha/2})(1-(1-\epsilon)^{\alpha})}{3\cdot 2^{\alpha+5}\cdot\beta}]^{\frac{1}{1-\alpha/2}}$.

Proof: We define a set S_i^g as following: for an active node u in cell g, *exponential annulus* $E_t^i(u) = A(u, 2^{t+1}2^i)\backslash A(u, 2^t 2^i)$, where $A(u,d)$ is the set of active nodes within distance d from u. If for every $t \in \{0, 1, ..., \log R_1 - 1\}$, $E_t^i(u) \le 48 * 2^{t(\alpha/2+1)}$, where R_1 is the maximum distance for any pair of nodes in the network, then u is called a *sparse node*. $S_i^g \subseteq V_i^g$ is the largest subset of sparse nodes in V_i^g that guarantees for any pair of nodes $u, v \in S_i^g$, $d(u,v) \ge (s+2)2^i$, where $s = [\frac{(1-2^{1-\alpha/2})(1-(1-\epsilon)^{\alpha})}{3\cdot 2^{\alpha+5}\cdot\beta}]^{\frac{1}{1-\alpha/2}}$. $S_i = \cup_{g\in\mathcal{G},color(g)=j} S_i^g$.

We prove this lemma in two steps: In the first step, with the assumption that $i \in \{0, 1, ..., \log\epsilon R_c - 1\}$ and $n_{<i} \le \varepsilon n_i$, a constant fraction of nodes in V_i is shown to be in S_i; The second step proves that a constant fraction of nodes in S_i become inactive with probability $1 - e^{-\Omega(|S_i|)}$.

For any fixed $i \in \{0, 1, ..., \log\epsilon R_c - 1\}$, the first step is proved by the following Claim 1 and Claim 2. To maintain clarity in the whole analysis here, the proofs of claims in Section 4 are put in the appendix.

Claim 1. If $n_{<i} \le \varepsilon n_i$ with $\varepsilon = \frac{1-(2^{1-\alpha/2})}{2}$, a constant fraction of nodes in V_i are sparse nodes.

Claim 2. At least $\frac{1}{(2s+5)^2}$ fraction of sparse nodes in V_i are in set S_i.

From claim 1 and 2, we get the result that at least $\frac{1}{2(2s+5)}$ nodes in V_i are spares nodes in S_i for step one.

Then, we start to prove the second step by subsequent Claim 3 and Claim 4. For any fixed non-empty set S_i, $i \in \{0, 1, ..., \log\epsilon R_c - 1\}$, let T_i be the set of nearest neighbors for all nodes in S_i. For a node $u \in S_i$ who listens and with node v as the nearest neighbor, we divide the interference at u into two parts: interference from nodes in $S_i \cup T_i \setminus \{u, v\}$ and that from nodes outside $S_i \cup T_i$. For a node $v_1 \notin S_i \cup T_i$, define $\hat{\mathcal{I}}(v_1)$ to be the sum of interference at nodes in S_i that is generated by node v_1, and $\hat{\mathcal{I}}(v_1)$ is also recorded as the sum of interference on nodes in $E_t^i(v_1) \cap S_i$ over all annulus. Noting that the strength of all received signals follows an exponential distribution as is mentioned in the model section, we have $M(\hat{\mathcal{I}}(v_1))$ being the mean of the variable $\hat{\mathcal{I}}(v_1)$. Also with an area argument, we get $|E_t^i(v_1) \cap S_i| \leq 24 * 2^{2t}$. So

$$\begin{aligned} M(\hat{\mathcal{I}}(v_1)) &= \sum_{t=0}^{\log R_1 - 1} |E_t^i(v_1) \cap S_i| \frac{P}{(2^i 2^t)^\alpha} \\ &= \frac{P}{2^{i\alpha}} \sum_{t=0}^{\log R_1 - 1} \frac{|E_t^i(v_1) \cap S_i|}{2^{t\alpha}} \\ &\leq \frac{P}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{24 * 2^{2t}}{2^{t\alpha}} < \frac{24 * P}{2^{i\alpha}} \left(\frac{1}{1 - 2^{2-\alpha}}\right) \end{aligned}$$

$M(\hat{\mathcal{I}}(v_1)) < c_{max} P / 2^{i\alpha}$ by setting $c_{max} = \frac{48}{1 - 2^{1-\alpha/2}}$.

Claim 3. For any constant c_1 with $p = c_1/(4c_{max})$, with probability $1 - e^{-\frac{c_1^2}{24 c_{max}^2} |S_i|}$, at least half of the nodes in S_i have $M(\hat{\mathcal{I}}(v_1))$ no larger than $c_1 P / 2^{i\alpha}$.

Claim 4. For any fixed round, a constant fraction of nodes in S_i become inactive with probability $1 - e^{-\Omega(|S_i|)}$.

By now, we have completed the two-step proof. With the above two steps, we get results that for $i \in \{0, 1, 2, \ldots \log \epsilon R_c - 1\}$, (1) when $n_{<i} \leq \varepsilon n_i$, $\frac{1}{2(2s+5)^2}$ fraction of nodes in set V_i are in set S_i; (2) in each round, with probability at least $1 - e^{\Omega(|V_i|)}$, more than $p(1-p) * e^{-1}(1 - e^{-1}) * |S_i|/4$ nodes become inactive. These results complete the proof of Lemma 1.

□

Lemma 1 depicts the reduction process of V_i in each round. We still need to consider the change of V_i between rounds. There are three factors which may change V_i between rounds: first, churns of nodes with some active nodes joining or ending the algorithm execution; second, nodes move from cell to cell; third, some active nodes in $V_{<i}$ may join V_i because their nearest neighbors have become inactive; We need to prove that even with these factors, each V_i for $i \in \{0, 1, 2, \ldots \log \epsilon R_c - 1\}$ will finally become an empty set, which means that for each non-empty cell, there is exactly one active node left and elected as the leader.

Let $\gamma_1 = 1 - \gamma$ and $\gamma_2 = (\gamma_1 + \rho/(1-\rho))(1+\lambda)^{2\varsigma} < 1$ where ρ and ς are constants given in Table 3. Our requirement on dynamic rate $\lambda < \frac{\gamma_1 + \rho/(1-\rho)}{1-\rho}^{-\frac{1}{4\varsigma}} - 1$ ensures $(1+\lambda)^{2\varsigma} < \frac{1-\rho}{\gamma_2}$.

In the subsequent analysis, we next try to upper bound the number of active nodes in each class V_i by a series of vectors $m_t(i), \hat{m}_t(i)$ for $t \geq 0$ and $0 \leq i \leq \log \epsilon R_c - 1$ as follows.

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

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

Here $T_i = i * h$ and $h = \lceil \log_{\gamma_2} \rho \rceil$, ρ is a constant which is set to be sufficiently small for the convenience of subsequent analysis.

The subsequent analysis consists of two parts. We first give a time bound $\hat{T}$ when all $\hat{m}_i(\hat{T})$ become 0. By the definition of $\hat{m}_i(t)$, it is easy to check that $\hat{T} \in O(\log n + \log R_c)$.

We divide the time into consecutive intervals I_i for $i = 0, 1, \ldots$, and each interval consists of $\varsigma = \lceil \max\{(4\tau)/(a_1(1-\gamma_2)), 4\tau\} \rceil$ rounds, where $\tau = \gamma_2/\gamma_1 > 1$ and a_1 is the constant hidden behind the Ω notation in the probability guarantee in Lemma 1.

We define some notations to facilitate our analysis.

- Define random events $\mathcal{E}(j)$ and $\mathcal{F}(j)$ for $j \geq 0$: $\mathcal{E}(j)$ occurs in a round r if in round r, $\hat{n}_i(r) \leq m_i(j)$ for all $i \in \{0, 1, \ldots, \log \epsilon R_c - 1\}$. Similarly, $\mathcal{F}(j)$ occurs in round r if in round r, $\hat{n}_i(r) \leq \hat{m}_i(j)$ for all $i \in \{0, 1, \ldots, \log \epsilon R_c - 1\}$;
- If in all rounds during interval I_i, $\mathcal{E}(j)$ always occurs, we say $\mathcal{E}(j)$ always occurs in interval I_i. If $\mathcal{E}(j)$ occurs in at least one round during I_i, we say $\mathcal{E}(j)$ occurs once in I_i. $\mathcal{F}(j)$ has the similar definitions.

We next give three lemmas on $\mathcal{E}(j)$ and $\mathcal{F}(j)$. Lemma 2 and Lemma 3 can be easily obtained and we omit the proofs.

Lemma 2. Events $\mathcal{E}(0)$ and $\mathcal{F}(0)$ occur in every round.

Lemma 3. When $\mathcal{F}(\hat{T})$ occurs, all classes of V_i for $i \in \{0, 1, \ldots, \log \epsilon R_c - 1\}$ are reduced to empty, i.e., exactly one active node is left in each non-empty cell g and finally elected as the leader.

Lemma 4. In any round, if $\mathcal{F}(j)$ occurs, then $\mathcal{E}(j)$ occurs.

Proof: Since $\hat{m}_i(t) = m_i(t) * \frac{\gamma_1}{\gamma_2}$ and $\gamma_2 > \gamma_1$, when $\mathcal{F}(j)$ occurs at any round r_1, for all $i \in \{0, 1, \ldots, \log \epsilon R_c - 1\}$, $\hat{n}_i(r_1) \leq \hat{m}_i(j) \leq m_i(j)$. Thus, the Lemma holds. □

We next analyze the progress of $\mathcal{F}(j)$.

Lemma 5. If $\mathcal{F}(j)$ occurs once in I_a, then, in I_{a+1}:
(a) $\mathcal{F}(j)$ or $\mathcal{F}(j-1)$ always occurs;
(b) $\mathcal{E}(j)$ always occurs;
(c) With probability at least $\frac{3}{4}$, $\mathcal{F}(j+1)$ occurs once;

Proof: We prove (a) and (b) first. If $j = 0$, $\mathcal{F}(0)$ and $\mathcal{E}(0)$ always occur in I_{a+1}; When $j > 0$, let r_1, r_2 be a round in I_a and I_{a+1} respectively. Since $\mathcal{F}(j)$ occurs in r_1, i.e., for any $i \in \{0, 1, \ldots, \log \epsilon R_c - 1\}$, $\hat{n}_i(r_1) \leq \hat{m}_i(j)$, we have

$$\begin{aligned} \hat{n}_i(r_2) &\leq \hat{n}_i(r_1) * (1+\lambda)^{r_2 - r_1} + \sum_{s'=0}^{i-1} n_{s'}(r_1) * (1+\lambda)^{r_2 - r_1} \\ &\leq \hat{n}_i(r_1) * (1+\lambda)^{2\varsigma} + \sum_{s'=0}^{i-1} n_{s'}(r_1) * (1+\lambda)^{2\varsigma} \\ &\leq \hat{m}_i(j) * (1+\lambda)^{2\varsigma} + \sum_{s'=0}^{i-1} \hat{m}_{s'}(j)(1+\lambda)^{2\varsigma} \\ &\leq \hat{m}_i(j) * (1+\lambda)^{2\varsigma} + (1+\lambda)^{2\varsigma} \gamma_1 \sum_{s'=0}^{i-1} m_{s'}(j-1) \end{aligned} \quad (2)$$

In order to bound the value of $\sum_{s'=0}^{i-1} m_{s'}(j-1)$, we consider two cases that $m_{i-1} < n/\gamma_1$ and $m_{i-1} = n/\gamma_1$. Notice that it is impossible that $m_{i-1} > n/\gamma_1$.

If $m_{i-1}(j-1) < n/\gamma_1$, we have the following claim.

Claim 5. If $m_{i-1} < n/\gamma_1$, $\sum_{s'=0}^{i-1} m_{s'}(j-1) \le \frac{\rho}{(1-\rho)} m_i(j-1)$.

Applying Claim 5 in Eqt. 2, we get that

$$\begin{aligned}\hat{n}_i(r_2) &\le \hat{m}_i(j) * (1+\lambda)^{2\varsigma} + (1+\lambda)^{2\varsigma}\gamma_1 \sum_{s'=0}^{i-1} m_{s'}(j-1)\\ &\le \hat{m}_i(j) * (1+\lambda)^{2\varsigma} + (1+\lambda)^{2\varsigma}\gamma_1 m_i(j-1)\rho/(1-\rho)\\ &= \frac{(1+\lambda)^{2\varsigma}}{1-\rho}\hat{m}_i(j) = \frac{(1+\lambda)^{2\varsigma}\gamma_2}{1-\rho}\hat{m}_i(j-1)\\ &= \frac{(1+\lambda)^{2\varsigma}\gamma_1}{(1-\rho)\gamma_2} m_i(j)\end{aligned}$$

Since $\frac{(1+\lambda)^{2\varsigma}\gamma_2}{1-\rho} \le 1$ and $\frac{(1+\lambda)^{2\varsigma}\gamma_1}{(1-\rho)\gamma_2} \le 1$, $\mathcal{F}(j-1)$ and $\mathcal{E}(j)$ occur in any round r_2 during I_{a+1}.

If $m_{i-1}(j-1) = n/\gamma_1$, we can see that $m_i(j-1) = m_i(j) = n/\gamma_1$, $\hat{m}_i(j) = n$, then $n_i^g(r_2) \le \hat{m}_i(j) < m_i(j)$. $\mathcal{F}(j)$ and $\mathcal{E}(j)$ always occur in I_{a+1}.

Combined all above together, we can get that $\mathcal{F}(j)$ or $\mathcal{F}(j-1)$ always occur in I_{a+1}, and $\mathcal{E}(j)$ always occurs in I_{a+1}, i.e., (a) and (b) are proved.

Then, we consider (c), which can be proved by the subsequent Claim 6 and Claim 7. Let r and $r+1$ be rounds in I_{a+1}.

Claim 6. For any $i \in \{0, 1, \ldots, \log \epsilon R_c - 1\}$, with probability at least $1 - e^{-\Omega(n_i^g(r))}$, $\hat{n}_i(r+1) \le \hat{m}_i(j+1)$.

Claim 7. With probability at least $3/4$, $\mathcal{F}(j+1)$ occurs once in interval I_{a+1}.

□

Lemma 6. $\mathcal{F}(\hat{T})$ occurs by the time $T_{LE} = \Theta(\log n + \log R_c)$ with high probability.

Proof: By Lemma 2 and Lemma 5, we know that: 1) $\mathcal{F}(0)$ always occurs in all intervals; 2) when $\mathcal{F}(j)$ occurs once in I_a, $\mathcal{F}(j-1)$ or $\mathcal{F}(j)$ always occurs in I_{a+1} and with probability at least $3/4$, $\mathcal{F}(j+1)$ occurs once in I_{a+1}. Thus, using the Chernoff bound, $\mathcal{F}(\hat{T})$ occurs w.h.p. within time complexity of $O(\log n + \log R_c)$ and the Lemma is proved. □

Based on all above analysis, we can get the final result.

Theorem 1. With high probability, the leader election algorithm can elect a leader in each non-empty cell in $O(\log n + \log R_c)$ rounds, if the dynamic rate $\lambda < \frac{\gamma_1 + \rho/(1-\rho)}{1-\rho}^{-\frac{1}{4\varsigma}} - 1$.[2]

5 IMPLEMENTATION OF ABSTRACT MAC LAYER

In this section, we present the implementation algorithm of the abstract MAC layer, as shown in Algorithm 2, also called algorithm AML. The algorithm can efficiently implement both acknowledgement and progress primitives under the same constant constraints on the dynamic rate as that in Section 4, i.e., $\lambda < (\frac{\gamma_1 + \rho/(1-\rho)}{1-\rho}^{-\frac{1}{4\varsigma}} - 1)$.

2. The detail value of γ_1, ρ, and ς is given in Table 3

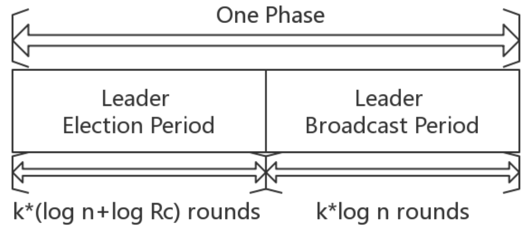


Fig. 4: Two periods in one phase

The algorithm execution is divided into successive phases, each of which consists of $T_P = T_{LE} + T_{LB}$ rounds, where T_{LE} and T_{LB} are the time complexity of leader election period and leader broadcast period respectively, $T_{LE} = k(\log n + \log R_c)$ and $T_{LB} = k \log n$. Basically, in each phase, the algorithm first invokes the leader election algorithm to elect a leader in each non-empty cell, and then makes the leaders successfully disseminate their messages using a TDMA scheduling generated by the cell coloring. Hence, in each phase, there are two periods: Leader Election (LE) Period and Leader Broadcast (LB) Period, as is illustrated in Fig. 4

Compared with the states in Algorithms 1 and 2 has one more state $\mathbb{S}_1$, which means that nodes do not have a message to disseminate and always keep listening. The other states of nodes in Algorithm AML are the same as those in Algorithm 1.

In the first period, for any node v with message m_v to transmit, it is in state $\mathbb{A}$ at the beginning. Otherwise, $state_v = \mathbb{S}_1$, which means v has no message to transmit and always keeps listening. Then, all nodes in state $\mathbb{A}$ will run $LE(n, R_c)$ to elect a leader in each non-empty cell. Notice that the nodes are allowed to move among cells during the election process. At the end of this period, for any cell g, which is non-empty at the beginning of this phase, there will be exactly one leader in g elected with high probability. By the constraint on dynamic rate, when an active node is elected as a leader in g, it will not leave g until the end of the current phase. Otherwise, $V^g_{\log \epsilon R_c}$ will change from 1 to 0, which violates the constraint on dynamic rate. This constraint is necessary, as otherwise, all leaders in current phase can move to a small area, where leaders are close to each other such that the interference is large enough to disturb all transmissions.

In the second period, the elected leaders will disseminate their messages in $k \log n$ rounds by executing the TDMA scheduling generated according to their colors. The TDMA scheduling ensures that each leader can disseminate its message to all neighbors within distance $(1+\epsilon)R_c$ in one period. After this period, leaders join state $\mathbb{S}_1$. Those nodes that still have messages to transmit but failed to become leader in previous phases, i.e. in state $\mathbb{S}$, will join state $\mathbb{A}$ again, to compete for becoming a leader in the subsequent phase.

Analysis. Basically, we first show some curical lemmas for the correctness of leader election and leader broadcasting, and then we will give time bounds for accomplishing the acknowledgement and progress primitives.

We consider a phase L of the algorithm execution, and

Algorithm 2: $AML(n, R_c)$

In each phase, for each node v in color j:

Leader election period:

if *v has message m_s to transmit* **then**
  $state_v = \mathbb{A}$;
else
  $state_v = \mathbb{S}_1$;
if $state_v = \mathbb{A}$ **then**
  $LE(n, R_c)$;

Leader broadcast period:

if $state_v = \mathbb{L}$ **then**
  for T_{LB} *rounds* **do**
    $slot = 0$;
    for $slot < c * c$ **do**
      if $slot = j$ **then**
        transmit m_v;
      $slot + +$;
  $state_v = \mathbb{S}_1$;
if $state_v = \mathbb{S}$ **then**
  $state_v = \mathbb{A}$;

have the following results.

Lemma 7. At the end of the leader election period in L, there is exactly one leader elected in each non-empty cell g with high probability.

Proof: The lemma is a direct corollary of Theorem 1. □

We next analyze the message disseminations of leaders. Consider a non-empty cell g. Let v be the leader elected in g during the leader election period of phase L. As discussed before, the dynamic rate constraint guarantees that the leader in each non-empty cell will not leave the cell during the leader broadcast period. The following lemma shows how leaders successfully disseminate their messages to their neighbors

Lemma 8. After a leader broadcast period, all nodes within distance R_c from the cell of v can successfully receive the message of v at least with high probability.

Proof: Let g' be the cell of v and u be a node within distance R_c from g' in round r_1, and r_1 be a round in the leader broadcast period of phase L.. By the triangle inquality, we get that $d(v, u) \leq \epsilon R_c + d(g', u) \leq (1 + \epsilon)R_c$.

Claim 8. If $d(u, v) \leq (1+\epsilon)R_c$, when v transmits in r_1, u can receive the message of v with probability $e^{-1} * (1 - e^{-1})$.

Proof: Assume that v transmits in slot s of round r_1. We divide the whole space into annuluses $\{C_b : b \geq 1\}$, where C_b denotes the annulus with distance from u between $(b-1)(c-1) * (\frac{\sqrt{2}\epsilon}{2} R_c)$ and $b * (c-1) * (\frac{\sqrt{2}\epsilon}{2} R_c)$. Let L_b be the set of leaders that also transmit in slot s and locate in C_b for $b \geq 2$. The TDMA scheduling ensures that any two leaders transmitting simultaneously are seperated by a distance at least $(c-1) * (\frac{\sqrt{2}\epsilon}{2} R_c)$. Hence, disks centered at leaders in L_b with radius $(c-1) * (\frac{\sqrt{2}\epsilon}{4} R_c)$ are disjoint, and these disks are in the annulus with distance from u between $(b - \frac{3}{2})(c-1) * (\frac{\sqrt{2}\epsilon}{2} R_c)$ and $(b + \frac{1}{2}) * (c-1) * (\frac{\sqrt{2}\epsilon}{2} R_c)$. Then the number of leaders transmitting simultaneously with v is upper bounded as follows.

$$\frac{\pi(\frac{\sqrt{2}\epsilon}{2} R_c)^2 (c-1)^2 ((b + \frac{1}{2})^2 - (b - \frac{3}{2})^2)}{\pi((c-1) * (\frac{\sqrt{2}\epsilon}{4} R_c))^2} \leq 16 * b$$

Furthermore, it is easy to get that the number of broadcasters in C_1 that simultaneously transmit with v is at most 4, and the mean of interference on u caused by these broadcasters is at most $M(\mathcal{I}_{C_1}) = 4P * (((\frac{\sqrt{2}\epsilon(c-1)}{2} - 1 - \epsilon)R_c))^{-\alpha}$. Then the mean of interference $\mathcal{I}$ at node u from other nodes that simultaneously transmit with v in slot s is bounded by:

$$\sum_{b=2}^{\infty} 16b * P * ((b-1)(c-1) * \frac{\sqrt{2}\epsilon}{2} R_c))^{-\alpha} + M(\mathcal{I}_{C_1})$$
$$\leq (32 * \frac{\alpha - 1}{\alpha - 2} + 4) * P * (\frac{\sqrt{2}\epsilon(c-1)}{2} - 1 - \epsilon)^{-\alpha} * R_c^{-\alpha}$$
$$= (32 * \frac{\alpha - 1}{\alpha - 2} + 4) * \frac{\beta N}{(1-\epsilon)^{\alpha}} * (\frac{\sqrt{2}\epsilon(c-1)}{2} - 1 - \epsilon)^{-\alpha}.$$

Setting $c = \lceil ((\frac{\beta(32\frac{\alpha-1}{\alpha-2}+4)}{(1+\epsilon)^{-\alpha} - (1-\epsilon)^{\alpha}})^{\frac{1}{\alpha}} + 1 + \epsilon) * \frac{\sqrt{2}}{\epsilon} + 1 \rceil$, we get

$$Pr(S_{v,u} \geq P/((1+\epsilon)R_c)^{\alpha}) = exp(-\frac{P/((1+\epsilon)R_c)^{\alpha}}{\bar{S}_{v,u}}) \geq e^{-1},$$

$$Pr(\mathcal{I} \leq ((1-\epsilon)^{-\alpha}(1+\epsilon)^{-\alpha} - 1) * \beta N)$$
$$= 1 - exp(-\frac{((1-\epsilon)^{-\alpha}(1+\epsilon)^{-\alpha} - 1) * \beta N)}{\bar{S}_{v,u}}) \geq \quad 1 - e^{-1}.$$

Thus, at least with probability $e^{-1} * (1 - e^{-1})$ at round r_1, when v transmits, u can receive the message of v by the SINR condition

$$SINR(u, v, S) \geq \frac{P * ((1+\epsilon)R_c)^{-\alpha}}{N + \mathcal{I}} \geq \beta$$

□

By applying a Chernoff bound on the above Claim, it can be proved that the message dissemination from leaders to their neighbors succeed with high probability after T_{LB} rounds. □

Theorem 2. With high probability, the AML algorithm can implement the abstarct MAC layer with time bounds
(i) $f_{ack} = |I|$, where I is the interval from the beginning (round 0) to a round t such that $(\Delta^g(I) + 1) * T_P \leq t$ holds for all non-empty cells g;
(ii) $f_{prog} = T_P = O(\log n + \log R_c)$.

Proof: (i) Consider a node v in a cell g. By Lemma 7 and Lemma 8, if cell g is non-empty, after each phase, there is an active node in g elected as leader and disseminate its message to all neighbors with high probability. Hence, during the interval I, after at most $\Delta^g(I)$ phases, either v has been elected as a leader in a phase and disseminated its message, or all active nodes except v appearing in cell g during the interval I have disseminated their messages. Hence, in the subsequent phase, there are no other active nodes in g competing with v, and v will accomplish local broadcast in this phase as analyzed above. By tuning the constant parameters carefully, it is easy to check that the result holds for all nodes.

(ii) Consider a node v. By the assumption, v stably connects to non-empty cells in at least one entire phase. Assume the phase as L, L_1 as the leader broadcasting period in L, v connects to cell g at period L_1, and let u be the leader elected in g during phase L. Then by the definition of stable connection, $d(v,u) \leq (1+\epsilon)R_c$ at period L_1. Then by Lemma 8, v can receive a message from u in L with high probability. This completes the proof. □

6 Necessity of Dynamic Rate Restriction

We next discuss whether a significantly larger dynamic rate can be handled by an efficient or asymptotically optimal AbsMAC layer implementation algorithm. However, if the algorithm involves a leader election procedure, the result is negative. Specifically, we have the following result.

Theorem 3. If the dynamic rate $\lambda \in \omega(1)$, there is no algorithm that can solve the leader election in $O(\log n)$ time.

Proof: The result is proved by contradiction. Assume that there is an algorithm $\mathcal{A}$ that can accomplish leader election in $O(\log n)$ rounds, which means that $\sum_{i=0}^{\log \epsilon R_c - 1} n_i$ can be reduced to empty within $O(\log n)$ rounds. We claim that during the algorithm execution, at least $\frac{\lambda}{1+\lambda}$ fraction of active nodes in $\mathcal{A}$ become inactive, i.e. $\frac{n(t)-\hat{n}(t)}{n(t)} \geq \frac{\lambda}{1+\lambda}$.

Otherwise, note that $n_i(t+1)$ can be recorded as $(1+\lambda)\hat{n}_i(t)$ for $i \in \{0,1,...,\log \epsilon R_c\}$, if $\frac{n(t)-\hat{n}(t)}{n(t)} < \frac{\lambda}{1+\lambda}$, we get $\hat{n}(t) > \frac{n(t)}{1+\lambda}$. Hence

$$n(t+1) = (1+\lambda)\hat{n}(t) > (1+\lambda) \cdot \frac{n(t)}{1+\lambda} = n(t)$$

This means that the set of active nodes will never be reduced to empty, i.e., the leader cannot be elected.

Then if implementing algorithm $\mathcal{A}$ in a static network with n active nodes, the above result implies that the leader election procedure can be completed in $O(\log_{\frac{\lambda}{1+\lambda}} n)$ rounds, which is $o(\log n)$ if $\lambda \in \omega(1)$. But in [36], it has been proved that $\Omega(\log n)$ is the minimum time needed for leader election, even if without interference. This contradiction completes the proof. □

7 Simulation Results

In this section, we investigate the empirical performances of our abstract MAC layer implementation algorithm. Specifically, we investigate (i) the performance of the algorithm under different dynamic rates and the largest dynamic rate that our algorithm can handle in reality; (ii) the acknowledgement bound f_{ack} and the progress bound f_{prog} our algorithm can attain when the dynamic rate changes; and (iii) the impact of SINR parameters on the algorithm performance. Because our algorithm is the first one for implementing abstract MAC layer in dynamic networks, there are no comparisons with previous work.

Structure and Parameter setting in simulation. Our entire simulation is developed as a C++ program with multiple functions, some of which are from standard libraries while the others are written by ourselves. Fig. 5 illustrates the flow chart of our simulation. For the simulation, n_0 nodes are randomly and uniformly distributed in a network with size $300m \times 300m$. Each node has a uniform transmission range of $30m$. The values of other parameters are given in Table 4. At least 20 runs of the simulation were 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 g++ compiler.

TABLE 4: Parameters in simulation

Notation	Definition	Value
n_0	Number of nodes	$[1000, 10000]$
R	Transmission range	30m
ϵ	Parameter in model	0.3
c	Parameter in coloring	9
p	Transmission probability	0.2
α	Parameter in SINR	$\{3,4\}$
β	Parameter in SINR	$\{1.5, 2\}$
λ	Dynamic rate	$[0, 0.3]$

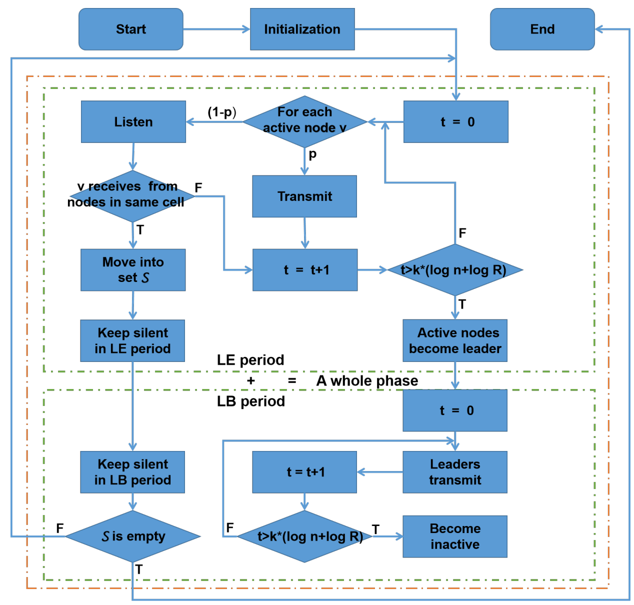


Fig. 5: Flow chart of our simulation

7.1 Algorithm Performance

In the simulation, n_0 active nodes, which have a message to broadcast, are randomly and uniformly distributed in the network initially. As the algorithm executes, we count the number of incomplete nodes (those that have not completed the acknowledgement or progress primitive). If a dynamic rate cannot be handled by our algorithm, the incomplete nodes will increase constantly, as there are new nodes joining the network. Then, for each fixed $n_0 \in \{1,2,3,4,5,6,7,8,9,10\} * 10^3$, we get the largest dynamic rate that can be handled by our algorithm, which is shown in Table 5. From the table, we can see that except the very sparse case of $n_0 = 1000$, the dynamic rate that can be handled is roughly the same in different cases.

Figure 6 shows the total number of nodes, the number of incomplete nodes in network, and the ratio of nodes that have completed the acknowledgement and progress

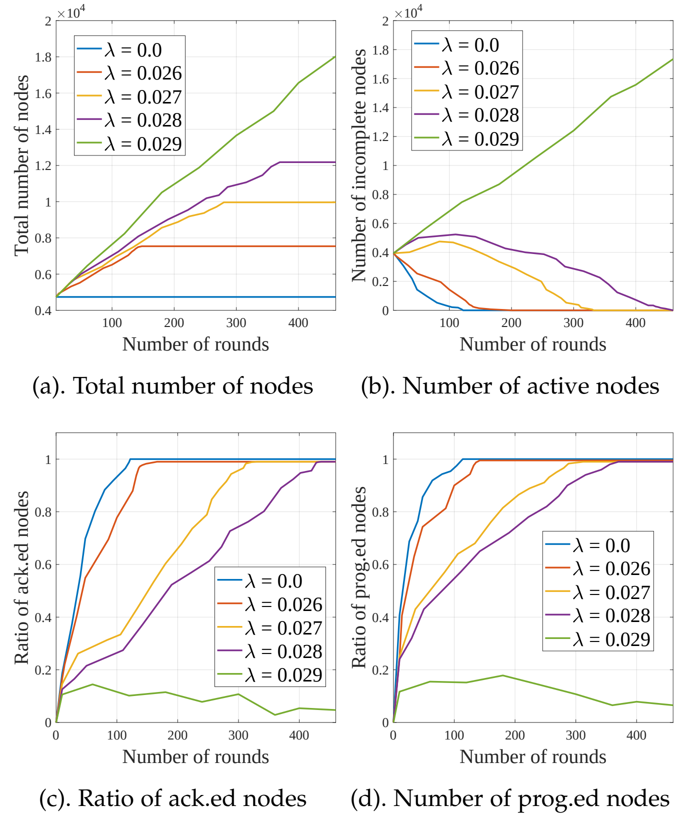
TABLE 5: Largest dynamic rate that can be handled

n_0	Dynamic upper bound	n_0	Dynamic upper bound
1000	0.14	2000	0.035
3000	0.029	4000	0.029
5000	0.028	6000	0.028
7000	0.028	8000	0.027
9000	0.027	10000	0.026

operation, for the case of $n_0 = 5000$ under different dynamic rates as the algorithm executes. If a node completes the acknowledgement/progress operation, we say it is ack.ed/prog.ed for short. In Fig. 6, the x-axes represent the number of rounds, and the y-axes represent the total number of nodes, the number of incomplete nodes, and ratio of ack.ed/prog.ed nodes, respectively. Also, the dynamic parameter is set as $\lambda \in \{0, 0.026, 0.027, 0.028, 0.029\}$.

The curves with $\lambda = 0$ correspond to the situation of static network. From the curves with $\lambda = 0$ in Fig. 6 (a)–(d), we can see that (i) the total number of nodes in the network is always unchanged because the network is static; (ii) the number of active nodes decreases to 0 sharply within 150 rounds; (iii) the ratio of ack.ed/prog.ed nodes increases from 0 to 1 rapidly. From the curves with $\lambda = 0.029$, we can see that the dynamic scenario of $\lambda = 0.029$ cannot be handled by our algorithm. When $\lambda = 0.029$, the dynamicity in network is too strong, which results in the total number of nodes and the number of incomplete nodes always increasing (as shown Fig. 6 (a) and (b)), and the ratio of ack.ed/prog.ed nodes (as shown in Fig. 6 (c) and (d)) becomes very small. To have a direct understanding of the dynamic level with $\lambda = 0.029$, we can see that the network size at round 120 is already twice as large as the network size at the beginning in Fig. 6 (a). In other cases with $\lambda \in \{0.026, 0.027, 0.028\}$, it can be seen that the number of incomplete nodes keeps stable at a very low value after the algorithm has executed for a period; and the ratio of ack.ed/prog.ed nodes gradually gets close to 1. This means that although there are new nodes joining the network due to the dynamic setting, the nodes can complete the local broadcast primitives as required in time, such that the total number of incomplete nodes stays at a low level.

Figure 7 shows how the acknowledgement bound f_{ack} and the progress bound f_{prog} change as the algorithm execution when initially $n_0 = 5000$ and λ varies from 0.026 to 0.029. In Figure 7, the x-axes represent the number of rounds in the algorithm execution, while the left y-axes represent the number of rounds for the bounds f_{ack} and f_{prog}, and the right y-axes represent the number of incomplete nodes. There are three curves in these figures, which represent the bounds f_{ack} and f_{prog} for the nodes finishing the local broadcast primitives in corresponding rounds and the number of incomplete nodes in corresponding rounds, respectively. From the figures, it can be seen that for $\lambda = 0.029$, the number of incomplete nodes and the bound f_{ack} keep increasing, which means that our algorithm cannot handle this dynamic scenario. But for the progress bound f_{prog}, it keeps decreasing in this situation. So our algorithm can handle a larger dynamic rate in terms of the progress primitive. For $\lambda \leq 0.028$, it can be found that the acknowledgement bound f_{ack} first increase due to



(a). Total number of nodes (b). Number of active nodes

(c). Ratio of ack.ed nodes (d). Number of prog.ed nodes

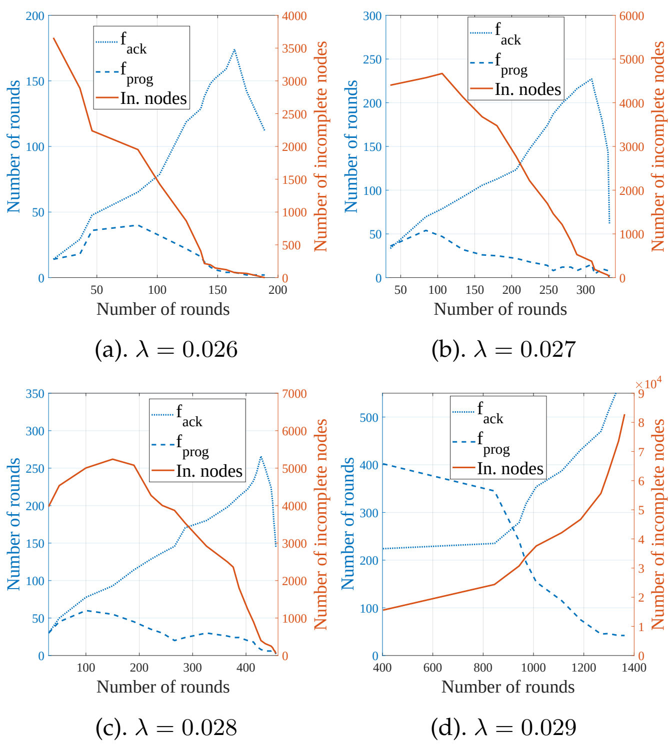
Fig. 6: Total number of nodes and the number of incomplete nodes as algorithm execution

the newly joining nodes and then keep decreasing or stable subsequently. Furthermore, we can see that our algorithm is very efficient, with f_{ack} and f_{prog} smaller than 280 and 60 in cases of $\lambda \leq 0.028$.

To show the efficiency of our algorithm in dynamic networks, we compare our dynamic abstract MAC layer algorithm (written as D-absMAC for short) with two state-of-the-art abstract MAC layer protocols [17], [41] (written as S-LBL and S-absMAC for short) that are proposed for static networks under the SINR model. Fig. 8 illustrates the comparison results in networks with various dynamic levels, i.e. $\lambda = 0.01, 0.02$. It can be seen that the ack. and prog. bound of our algorithm is significantly smaller than those of the algorithms in [17], [41]. And, the performance of S-absMAC is better than that of S-LBL. This is because with S-LBL, nodes decide to transmit or not by computing a special node set, which is heavily influenced by the dynamic nodes. Whereas, the adaptive technique used in [41] can inherently tolerate some slight dynamic behaviors. In conclusion, the comparison results indicate that our algorithm is much more efficient when facing dynamic behaviors in networks.

7.2 Impact of SINR parameters

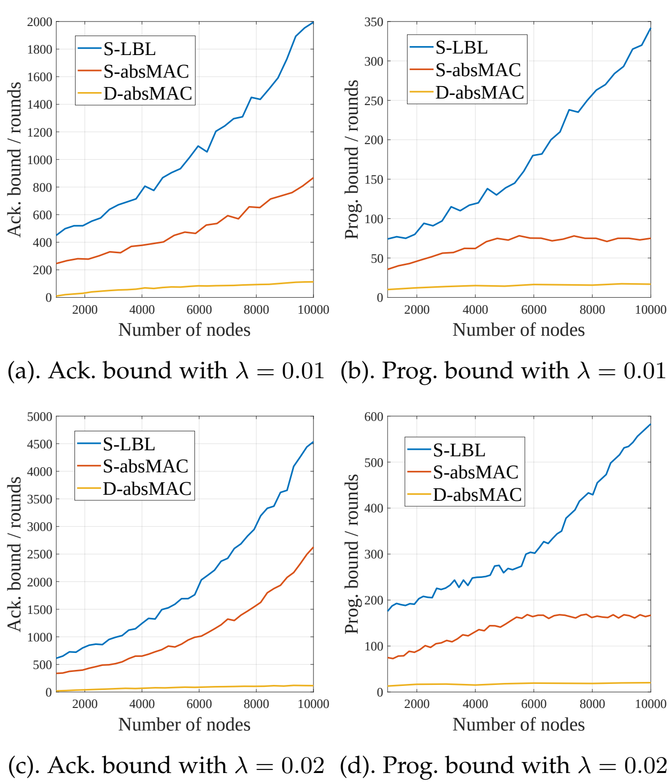
We illustrate the impact of SINR parameters α and β in Figure 9, where it sets that $n_0 = 5000$ and $\lambda = 0.028$. In the figure, the curves represents the bounds f_{ack} and f_{prog} under different setting of α and β as the algorithm execution, respectively. From the figures, it can be seen that acknowledgement/progress bounds for different α and β



(a). $\lambda = 0.026$ (b). $\lambda = 0.027$

(c). $\lambda = 0.028$ (d). $\lambda = 0.029$

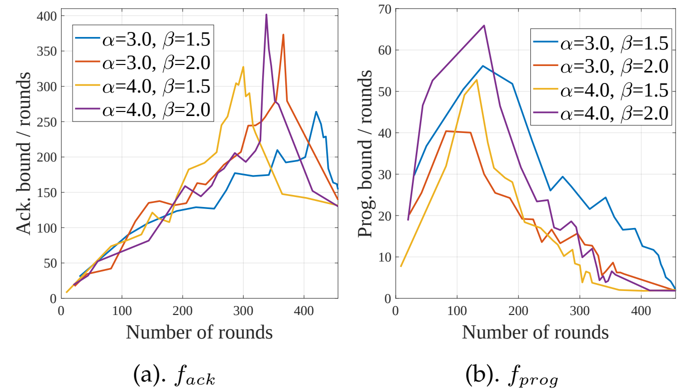
Fig. 7: Changes of f_{ack}, f_{prog} and the number of incomplete nodes as algorithm execution



(a). Ack. bound with $\lambda = 0.01$ (b). Prog. bound with $\lambda = 0.01$

(c). Ack. bound with $\lambda = 0.02$ (d). Prog. bound with $\lambda = 0.02$

Fig. 8: Comparison of our algorithm with others



(a). f_{ack} (b). f_{prog}

Fig. 9: Ack. and prog. bounds under different α and β

are similar on values and tendency. Hence, our algorithm is insensitive to SINR parameters.

7.3 Summary

The simulation results show that our algorithm can handle a constant dynamic rate. Because the dynamic 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. In the dynamic environment, our algorithm can efficiently accomplish the functions of local broadcast primitives in the abstract MAC layer, and it has been shown that the algorithm is insensitive to the SINR parameters.

8 Conclusion

In this paper, we studied the problem of implementing local broadcast primitives in the abstract MAC layer in dynamic networks. We proposed a dynamic network model based on SINR interference, which incorporates both churns and node mobility. Under this comprehensive model, we proposed an efficient implementation algorithm the performance of which is guaranteed with high probability, under the reasonable constraint of constant dynamic rate. The obtained acknowledgement bound is inferior to the optimal solution by at most a logarithmic factor and the progress bound is asymptotically optimal. Furthermore, we show that the constraint of constant dynamic rate is necessary for getting efficient or asymptotically optimal solutions. Our algorithm provides a base for designing new efficient algorithms for high-level communication problems. Extensive simulations indicate the good performance of our algorithm in realistic settings.

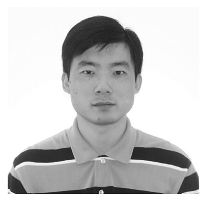
It should be noted that our algorithm relies on reliable communication channels. It is unknown whether our algorithm is resilient to channel jamming for instance, which could be an interesting future research direction.

References

[1] M.H. Bodlaender, M.M. Halldórsson and P. Mitra. Connectivity and Aggregation in Multihop Wireless Networks. In *PODC*, 2013.

[2] S. Cheng, Z. Cai, J. Li. Curve Query Processing in Wireless Sensor Networks. In *IEEE Trans. Vehicular Technology*, 64(11): 5198-5209, 2015.

[3] S. Cheng, Z. Cai, J. Li, X. Fang. Drawing dominant dataset from big sensory data in wireless sensor networks. In *INFOCOM* 2015.
[4] S. Cheng, Z. Cai, J. Li, H. Gao. Extracting Kernel Dataset from Big Sensory Data in Wireless Sensor Networks. In *IEEE Trans. Knowl. Data Eng.*, 29(4): 813-827, 2017.
[5] B.S. Chlebus, D.R. Kowalski and M. Strojnowski. Fast scalable deterministic consensus for crash failures. In *PODC*, 2009.
[6] A.E.F. Clementi, A. Monti and R. Silvestri. Round robin is optimal for fault-tolerant broadcasting on wireless networks. In *Journal of Parallel and Distributed Computing*, 64(1): 89-96, 2004.
[7] A. Cornejo, S. Gilbert and C. Newport. Aggregation in dynamic networks. In *PODC*, 2012.
[8] A. Cornejo, N. Lynch, S. Viqar, and J.L. Welch. Neighbor discovery in mobile ad hoc networks using an abstract mac layer. In *Allerton*, 2009.
[9] A. Cornejo, S. Viqar, and J.L. Welch. Reliable neighbor discovery for mobile ad hoc networks. In *Ad Hoc Networks*, 12: 259-277, 2014.
[10] B. Dappuri, T.G. Venkatesh. Design and Performance Analysis of Multichannel MAC Protocol for Cognitive WLAN. In *IEEE Trans. Vehicular Technology*, 67(6): 5317-5330, 2018.
[11] S. Daum, S. Gilbert, F. Kuhn, and C. Newport. Broadcast in the Ad Hoc SINR Model. In *DISC*, 2013.
[12] M. Dinitz, J.T. Fineman, S. Gilbert and C. Newport. Smoothed analysis of dynamic networks. In *DISC*, 2015.
[13] P. Fazio, F.D. Rango, C. Sottile. A Predictive Cross-layered Interference Management in a Multichannel MAC with Reactive Routing in VANET.In *IEEE Trans. Mob. Comput.*, 15(8): 1850-1862, 2016.
[14] J.T. Fineman, S. Gilbert, F. Kuhn, C. Newport. Contention Resolution on a Fading Channel. In *PODC* 2016.
[15] M. Ghaffari, E. Kantor, N. Lynch, and C. Newport. Multi-message broadcast with abstract mac layers and unreliable links. In *PODC*, 2014.
[16] O. Goussevskaia, T. Moscibroda, and R. Wattenhofer. Local broadcasting in the physical interference model. In *DIALM-POMC*, 2008.
[17] M.M. Halldórsson, S. Holzer, and N. Lynch. A local broadcast layer for the sinr network model. In *PODC*, 2015.
[18] Z. He, Z. Cai, S. Cheng, X. Wang. Approximate aggregation for tracking quantiles and range countings in wireless sensor networks. In *Theor. Comput. Sci.*, 607: 381-390, 2015.
[19] T. Jurdzinski, D.R. Kowalski, M.Rozanski and G. Stachowiak. On Setting-Up Asynchronous Ad Hoc Wireless Networks. In *INFOCOM*, 2015.
[20] M. Khabbazian, D.R. Kowalski, F. Kuhn, and N.A. Lynch. Decomposing broadcast algorithms using abstract MAC layers. In *Ad Hoc Networks*, 12:219-242, 2014.
[21] M. Khabbazian, F. Kuhn, D.R. Kowalski, N.A. Lynch. Decomposing broadcast algorithms using abstract MAC layers. In *DIALM-PODC*, 2010.
[22] M. Khabbazian, F. Kuhn, N.A. Lynch, M. Médard, and A. ParandehGheibi. MAC design for analog network coding. In *FOMC*, 2011.
[23] F. Kuhn, N.A. Lynch and C.C. Newport, Brief announcement: hardness of broadcasting in wireless networks with unreliable communication. In *PODC*, 2009.
[24] F. Kuhn, N.A. Lynch, and C.C. Newport. The abstract MAC layer. In DISC 2009.
[25] F. Kuhn, N.A. Lynch, C.C. Newport. The abstract MAC layer. In *Distributed Computing*, 24(3-4): 187-206, 2011.
[26] F. Kuhn, N.A. Lynch and R. Oshman, Distributed computation in dynamic networks. In *STOC*, 2010.
[27] F. Kuhn, T. Moscibroda, and R. Wattenhofer, Initializing newly deployed ad hoc and sensor networks. In *MOBICOM*, 2004.
[28] F. Kuhn and R. Oshman. Dynamic networks: models and algorithms. In *SUGACT*, 42(1): 82-96, 2011.
[29] J. Li, S. Cheng, Z. Cai, J. Yu, C. Wang, Y. Li. Approximate Holistic Aggregation in Wireless Sensor Networks. In *TOSN*, 13(2): 11:1-11:24, 2017.
[30] C.H. Lin, K. Lin, W. Chen. Channel-Aware Polling-Based MAC Protocol for Body Area Networks: Design and Analysis, In *IEEE Sensors Journal*, 17(9): 2936-2948, 2017.
[31] N.A. Lynch and C. Newport. A (truly) local broadcast layer for unreliable radio networks. In *PODC*, 2015.
[32] N.A. Lynch, T. Radeva, S. Sastry. Asynchronous leader election and MIS using abstract MAC layer. In *FOMC*, 2012.
[33] C.C. Newport. Consensus with an abstract MAC layer. In *PODC*, 2014.
[34] C.C. Newport, P. Robinson. Fault-Tolerant Consensus with an Abstract MAC Layer. In *DISC*, 2018.
[35] J. Schneider and R. Wattenhofer. Coloring unstructured wireless multihop networks. In *PODC*, 2009.
[36] J. Schneider and R. Wattenhofer. What Is the Use of Collision Detection (in Wireless Networks)? In *DISC*, 2010.
[37] A. Sinha, L. Tassiulas and E. Modiano. Throughput-optimal broadcast in wireless networks with dynamic topology. In *MobiHoc*, 2016.
[38] 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.
[39] D. Yu, Y. Wang, M.M. Halldórsson and T. Tonoyan. Dynamic Adaptation in Wireless Networks Under Comprehensive Interference Via Carrier Sense. In *IPDPS*, 2017.
[40] 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. In *INFOCOM*, 2015.
[41] D. Yu, Y. Zhang, Y. Huang, H. Jin, J. Yu, Q.S. Hua. Exact Implementation of Abstract MAC Layer via Carrier Sensing. In *INFOCOM*, 2018.
[42] 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.
[43] D. Yu, Y. Zou, Y. Zhang, F. Li, J. Yu, Y. Wu, X. Cheng, F.C.M. Lau. Distributed Dominating Set and Connected Dominating Set Construction Under the Dynamic SINR Model. In *IPDPS*, 2019.
[44] 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.



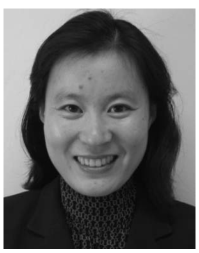
Dongxiao Yu received the BSc degree in 2006 from the School of Mathematics, Shandong University and the PhD degree in 2014 from the Department of Computer Science, The University of Hong Kong. He became an associate professor in the School of Computer Science and Technology, Huazhong University of Science 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.

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 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).

Yong Zhang received his PhD degrees in Computer Science from Fudan University, China, in 2007. Currently he is a professor in Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. Prior to joining SIAT, he has worked as Post-Doctoral Fellow in TU-Berlin and senior research associate in the University of Hong Kong. His research interests include design and analysis of algorithms, graph algorithms, online algorithms, distributed computing, etc.

Feng Li received his BS and MS degrees in Computer Science from Shandong Normal University, China, in 2007, and Shandong University, China, in 2010, respectively. He got his PhD degree (also in Computer Science) from Nanyang Technological University, Singapore, in 2015. From 2014 to 2015, he worked as a research fellow in National University of Singapore, Singapore. He is currently an assistant professor at School of Computer Science and Technology, Shandong University, China. His research interests include distributed algorithms and systems, wireless networking, mobile sensing and computing, and Internet of Things.



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.

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. He is a senior member of the IEEE and the IEEE Computer Society.

Appendix

Proof for Claim 1

Proof: We use a stronger condition to determine sparse nodes: for a node $u \in V_i$, for every $t \in \{0, 1, ..., \log \epsilon R_c - 1\}$, it satisfies that $|E_t^i(u) \cap V_{\geq i}| \leq 24 * 2^{t(\alpha/2+1)}$ and $|E_t^i(u) \cap V_{<i}| \leq 24 * 2^{t(\alpha/2+1)}$. Let J_i be the set of nodes which satisfy the above condition. Then, we try to figure out the ratio of $\frac{|J_i|}{|V_i|}$.

We first show the condition that $|E_t^i(u) \cap V_{\geq i}| \leq 24 * 2^{t(\alpha/2+1)}$. Since nodes in $V_{\geq i}$ keep distance at least 2^i with each other, the disks centered at nodes in $V_{\geq i}$ and with radius 2^{i-1} are disjoint. Considering any given annulus $E_t^i(u)$, an area argument in the following Eqt. 3 shows that for each node $u \in V_i$, $|E_t^i(u) \cap V_{\geq i}| \leq 24 * 2^{2t} \leq 24 * 2^{t(\alpha/2+1)}$.

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

Under the condition that $|E_t^i(u) \cap V_{<i}| \leq 24 * 2^{t(\alpha/2+1)}$, for a fixed class V_i, i, t, let Γ_t^i be the sum of nodes in $E_t^i(u) \cap V_{<i}$ for all nodes u in V_i. Then we have

$$\begin{aligned} \Gamma_t^i = \sum_{u \in V_i} |E_t^i(u) \cap V_{<i}| &= \sum_{u' \in V_{<i}} |E_t^i(u') \cap V_i| \\ \leq n_{<i} * 24 * 2^{2t} &\leq \varepsilon * n_i * 24 * 2^{2t} \end{aligned}$$

i.e. at most $\varepsilon * 2^{t(1-\alpha/2)} * n_i$ nodes do not satisfy the condition that $|E_t^i(u) \cap V_{<i}| \leq 24 * 2^{t(\alpha/2+1)}$. Thus for a fixed t, at most $\varepsilon * 2^{t(1-\alpha/2)} * n_i$ nodes are not sparse ones in $E_t^i(u)$ for each node $u \in V_i$. Summing up the number of non-sparse nodes for all $t \in \{0, 1, ..., \log R_1 - 1\}$, we get an upper bound on the number of non-sparse nodes in V_i:

$$\begin{aligned} \sum_{t=0}^{\log R_1 - 1} n_i * \varepsilon * 2^{t(1-\alpha/2)} &= n_i * \varepsilon * \sum_{t=0}^{\log R_1 - 1} (2^{1-\alpha/2})^t \\ &\leq n_i * \varepsilon * \frac{1}{1 - (2^{1-\alpha/2})} \\ &= \frac{1}{2} n_i. \end{aligned}$$

So, $\frac{|J_i|}{|v_i|} \geq \frac{1}{2}$ and we get the claim that when $n_{<i} \leq \varepsilon n_i$, more than half of nodes in V_i are sparse nodes. □

Proof for Claim 2

Proof: Since S_i is defined to be the largest subset of good nodes that have distance at least $(s+2)2^i$ pairwise, all spares nodes in V_i are covered by the disks centered at nodes in S_i and with radii $(s+2)2^i$. To get $|S_i|/|V_i|$, we bound the maximal number of spares nodes which could be covered by a disk centered at nodes in S_i and with radius of $(s+2)2^i$ via the following area argument.

Considering any node $v \in S_i$ and all the spares nodes in V_i within distance $(s+2)2^i$ from v. Let D_v and D'_v be the disk centered at v, with radius $(s+2)2^i$ and $(s+\frac{5}{2})2^i$ respectively. Since the spares nodes in D_v have distance at least 2^i with each other, the disks D''_vs centered at those spares nodes with radii 2^{i-1} are disjoint, and all D''_vs are covered by D'_v. Then we can get the number of spares nodes in D_v is at most

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

The claim then proved with the result that at least $\frac{1}{(2s+5)^2}$ fraction of spares nodes are in S_i □

Proof for Claim 3

Proof: We prove the claim in two cases.

Case 1. $c_1 \geq c_{max}$.

For any node $u \in S_i$, $\mathcal{I}(u)$ is used to record the interference at u that is caused by nodes outside $S_i \cup T_i$. Let $M(\mathcal{I}(u))$ be the mean of variable $\mathcal{I}(u)$, then

$$\begin{aligned} M(\mathcal{I}(u)) &\leq \sum_{t=0}^{\log R_1 - 1} |E_t^i(u)| \frac{P}{(2^t 2^i)^\alpha} \\ &= \frac{P}{2^{i\alpha}} \sum_{t=0}^{\log R_1 - 1} |E_t^i(u)|/2^{t\alpha} \\ &\leq \frac{P}{2^{i\alpha}} \sum_{t=0}^{\log R_1 - 1} \frac{48 * 2^{t(\alpha/2+1)}}{2^{t\alpha}} \\ &= \frac{48P}{2^{i\alpha}} \sum_{t=0}^{\log R_1 - 1} \frac{1}{2^{t(\alpha/2-1)}} \\ &< \frac{48P}{2^{i\alpha}} (\frac{1}{1-2^{1-\alpha/2}}) \\ &\leq c_{max} P/2^{i\alpha} \\ &\leq c_1 P/2^{i\alpha} \end{aligned}$$

Case 2. $c_1 < c_{max}$.

We define a random variable x_{v_1} for a node $v_1 \notin S_i^g \cup T_i^g$ as

$$x_{v_1} = \begin{cases} M(\hat{\mathcal{I}}(v_1))2^{i\alpha}/(c_{max}P) & \text{when node } v_1 \text{ transmits} \\ 0 & \text{when node } v_1 \text{ listens} \end{cases}$$

Then we can get

$$\begin{aligned} \mathbb{E}\Bigg[\sum_{v_1 \notin S_i \cup T_i} x_{v_1} \Bigg] &= \sum_{v_1 \notin S_i \cup T_i} p * M(\hat{\mathcal{I}}(v_1))2^{i\alpha}/(c_{max}P) \\ &= p \sum_{v_1 \notin S_i \cup T_i} M(\hat{\mathcal{I}}(v_1))2^{i\alpha}/(c_{max}P) \end{aligned}$$

Because $\frac{|S_i|}{2} * c_1 * P/2^{i\alpha} \leq \sum_{v_1 \notin S_i \cup T_i} M(\hat{\mathcal{I}}(v_1)) \leq |S_i| * c_{max} * P/2^{i\alpha}$, we get $(c_1^2/8c_{max}^2)|S_i| \leq \mathbb{E}\Big[\sum_{v_1 \notin S_i \cup T_i^g} x_{v_1}\Big] \leq c_1|S_i|/(4c_{max})$. Notice that $x_{v_1} \in [0,1)$. Then using standard Chernoff bound for the set of independent random variable $\{x_{v_1} : v_1 \notin S_i \cup T_i\}$ with $\mu = \mathbb{E}[\sum_{v_1 \notin S_i \cup T_i}(x_{v_1})]$, we get

$$\begin{aligned} &Pr\Big(\sum_{v_1 \notin S_i \cup T_i} x_{v_1} \geq 2 * (c_1|S_i|/(4c_{max})) \Big) \\ &\leq Pr\Big(\sum_{v_1 \notin S_i \cup T_i} x_{v_1} \geq 2\mu \Big) \leq e^{-\mu/3} \leq e^{-\frac{c_1^2}{24c_{max}^2}|S_i|} \end{aligned}$$

Then, with probability at least $1 - e^{-\frac{c_1^2}{24c_{max}^2}|S_i|}$,

$$\begin{aligned} \sum_{v_1 \notin S_i \cup T_i} M(\hat{\mathcal{I}}(v_1)) &= \sum_{v_1 \notin S_i \cup T_i} x_v * c_{max} P/2^{i\alpha} \\ &\leq (2c_1|S_i|/(4c_{max})) * c_{max}P/2^{i\alpha} \\ &= c_1|S_i|P/2^{i\alpha+1} \end{aligned}$$

Since $\frac{c_1|S_i|P/2^{i\alpha+1}}{c_1 P/2^{i\alpha}} = \frac{|S_i|}{2}$, it is impossible for more than half of nodes in S_i having the mean of interference from nodes outside $S_i \cup T_i$ larger than $c_1 P/2^{i\alpha}$. □

Proof for Claim 4

Proof: We consider the case that $u \in S_i$ receives a message from its nearest neighbor v in the same cell. Let $\mathcal{E}$ be the event that u listens and v transmits, then $Pr(\mathcal{E}) = p(1-p)$. Under the assumption that $\mathcal{E}$ occurs, we consider the probability that u can receive message from v. According to the SINR model, we still need to bound the interference at $u \in S_i$, and the strength of signal from v to u. We first bound the interference from nodes in $S_i \cup T_i \setminus \{u, v\}$. By the definition of S_i, each pair of nodes in S_i have distance at least $(s+2)2^i$ from each other. And each node in S_i has the nearest neighbor within the range of $[2^i, 2^{i+1})$, since $S_i \subseteq V_i$. Then the distance between u and nodes in $(S_i \cup T_i) \setminus \{u, v\}$ is at least $s * 2^i$. Let $\mathcal{I}_1$ denote the interference at u caused by nodes in $(S_i \cup T_i) \setminus \{u, v\}$. Then, $\mathcal{I}_1$ is also a random variable which is exponentially distributed with mean $M(\mathcal{I}_1)$, which can be bounded as follows.

$$M(\mathcal{I}_1) = \sum_{t=\log s} \frac{|E_t^i(u)|P}{(2^i 2^t)^\alpha} \leq \frac{48P}{2^{i\alpha}} \cdot \frac{1}{s^{\alpha/2-1}} \cdot \frac{1}{1-2^{1-\alpha/2}}. \quad (4)$$

Noting that $s = [\frac{(1-2^{1-\alpha/2})(1-(1-\epsilon)^\alpha)}{3 \cdot 2^{\alpha+5} \cdot \beta}]^{\frac{1}{1-\alpha/2}}$ and setting $c_1 = \frac{1-(1-\epsilon)^\alpha}{2^{\alpha+1}\beta}$, we get $M(\mathcal{I}_1)$, i.e. the mean of interference at u caused by nodes in $(S_i \cup T_i) \setminus \{u, v\}$, is at most $c_1 P/2^{i\alpha}$.

Combining Claim 3 with the above result for interference from nodes in $S_i \cup T_i \setminus \{u, v\}$, with probability at least $1 - e^{-\frac{c_1^2}{24c_{max}^2}|S_i|}$, at least half of nodes in S_i experience an interference the mean of which is at most $2c_1 P/2^{i\alpha}$. Then, let $\mathcal{E}_1$ be the event that $S_{u,v} \geq P/2^{\alpha(i+1)}$ and $\mathcal{E}_2$ be the event that interference experienced by u smaller than $2c_1 P/2^{i\alpha}$. Then, we get

$$\begin{aligned} Pr(\mathcal{E}_1) &= exp(-\frac{P/2^{\alpha(i+1)}}{\bar{S}_{uv}}) \geq e^{-1}, \\ Pr(\mathcal{E}_2) &= 1 - exp(-\frac{2c_1 P/2^{i\alpha}}{M(\mathcal{I}_1) + M(\hat{\mathcal{I}}(v_1))}) \\ &\geq 1 - exp(-\frac{2c_1 P/2^{i\alpha}}{2c_1 P/2^{i\alpha}}) = 1 - e^{-1} \end{aligned}$$

Under the assumption that $\mathcal{E}_1$ and $\mathcal{E}_2$ occur, by the SINR condition, we can show that if u experiences an interference that is at most $2c_1P/2^{i\alpha}$, it can receive a message from its nearest neighbor v as follows.

$$SINR(v,u) > \frac{P/2^{\alpha(i+1)}}{2c_1P/2^{i\alpha}+N} \geq \beta$$

Combining the successful transmission result with the assumption that $\mathcal{E}$, $\mathcal{E}_1$ and $\mathcal{E}_2$ occur, which occurs with probability $p(1-p)*e^{-1}*(1-e^{-1})$, we get that in expectation, there are $p(1-p)*e^{-1}*(1-e^{-1})*|S_i|/2$ nodes become inactive. Claim 4 is then proved by applying Chernoff bound. □

Proof for Claim 5

Proof: By the definition of $m_i(t)$, if $m_{i-1}(j-1) < n/\gamma_1$, then for $\forall s' \in \{0,1,\ldots,i-1\}$, $m_{s'}(j-1) = \rho m_{s'+1}(j-1)$, and $\sum_{s'=0}^{i-1} m_{s'}(j-1) \leq m_i(j-1)\rho/(1-\rho)$ □

Proof for Claim 6

Proof: The prove can be divided into three cases.

Case 1. $m_{i-1}(j-1) = n/\gamma_1$. In this case, $m_i(j-1) = m_i(j) = n/\gamma_1$, then $\hat{m}_i(j+1) = n$ and $\hat{n}_i(r+1) \leq \hat{m}_i(j+1)$

Case 2. $n_i(r+1) \leq \hat{m}_i(j+1)$. In this case, $\hat{n}_i(r+1) \leq n_i(r+1) \leq \hat{m}_i(j+1)$.

Case 3. $m_{i-1}(j-1) < n/\gamma_1$ and $n_i(r+1) \geq \hat{m}_i(j+1)$. Because $\mathcal{E}(j)$ always occurs in I_{a+1}, $\hat{n}_{<i}(r) \leq m_{<i}(j)$. By $m_{i-1}(j-1) < n/\gamma_1$ and Claim 5, $m_{<i}(j) \leq m_i(j)\rho/(1-\rho)$. By the definition of dynamic rate, $n_{<i}(r+1) \leq (1+\lambda)\hat{n}_{<i}(r)$. Combining all above together, we have

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

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

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

□

Proof for Claim 7

Proof: We prove the Claim by contradiction. Assume that $\mathcal{F}(j+1)$ does not occur in any round of I_{i+1}. By Claim 6, for any $i \in \{0,1,\ldots,\log\epsilon R_c - 1\}$, the error probability, i,e, the probability that $\hat{n}_i$ is larger than $\hat{m}_i(j+1)$ when interval I_{a+1} ends is

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

By applying an union bound on the error probabilities for all is, the probability that at least one $\hat{n}_i$ is larger than $\hat{m}_i(j+1)$ at the end of I_{a+1} is at most

$$\sum_{i=0}^{\log\epsilon R_c - 1} (1-\gamma_2)/(4m_i(j+1)) \leq \frac{1-\gamma_2}{4}\sum_{i=0}^{+\infty}\gamma_2^i \leq \frac{1}{4}$$

Hence, with probability at least $3/4$, $\mathcal{F}(j+1)$ occurs once in I_{a+1}. □