Optimal Distributed Leader Election without Decoding

Dongxiao Yu[1], Yifei Zou[2,3,⋆], Yong Zhang[2], Feng Li[1], Shikun Shen[4], and Falko Dressler[5]

[1] School of Computer Science and Technology, Shandong University, Qingdao, P.R. China,
{dxyu,fli}@sdu.edu.cn
[2] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, P.R. China,
{yf.zou,zhangyong}@siat.ac.cn
[3] Department of Computer Science, The University of Hong Kong, Hong Kong, P.R. China,
yfzou@cs.hku.hk
[4] School of Computer Science, Wuhan University, Wuhan, P.R. China,
whussk@qq.com
[5] Heinz Nixdorf Institute and the Dept. of Computer Science, Paderborn University, Paderborn, Germany,
dressler@ccs-labs.org

Abstract. In the past decades, with the exponential implementation of large-scale wireless networks such as Internet-of-Things, an enormous demand on designing the relative algorithms and protocols has come into our sight. As an important primitive function in wireless communication, the problem of leader election has attracted much attention in research. In this paper, we pursue efficient distributed solutions for leader election in large-scale wireless networks. Specifically, a simple but optimal distributed leader election algorithm in wireless networks with asynchronous wakeup nodes is proposed, under the practical Signal-to-Interference-plus-Noise (SINR) interference-depicting model. Our algorithm is randomized and can elect a leader in an asymptotically optimal time bound of $\Theta(\log n)$ with high probability, where n is the number of nodes in network. The algorithm does not rely on any information about the network topology, needs only simple node capability to implement, and uses no encoding and decoding functions. Hence, the algorithm is efficient and energy-saving, which makes it possible to be implemented in a wide range of scenarios, and become a building block for many higher-level operations, like consensus, broadcast and so on.

Keywords: Leader election; Distributed algorithm; Physical carrier sensing.

1 Introduction

As one of the primitives in distributed computing, leader election has always been an important problem and hot topic since it was firstly proposed in 1970 [1]. The final goal is to select one process to be the leader in a multi-process scenario. With a leader, many higher-level functions in networks become operable, such as local broadcast, backbone construction, and consensus. So each time when a new efficient leader election algorithm is proposed, those functions sitting above would be benefited too.

In this work, we focus on the leader election problem under the SINR model in the large-scale wireless network. We are interested in distributed solutions which are more desirable in practice. Comparing with traditional protocol models, the SINR model depicts the interference in the network more accurately, which is a fading and accumulative phenomenon. The accumulation of interference implies global knowledge which makes distributed algorithm design difficult. In such a design, nodes can only acquire local information which is far from enough to accurately estimate the interference from all simultaneously transmitting nodes. However, we find that via the physical carrier sensing, the global interference in the SINR model can also be used to deliver some simple information even across distances much larger than the transmission range without using encoding and decoding in transmission. The observation gives rise to our efficient leader election algorithm.

⋆ The corresponding author is Yifei Zou

Contribution: We propose an efficient distributed leader election algorithm, which can select a node as the leader among all nodes in the network within $O(\log n)$ rounds w.h.p., with high probability in short. Note that this result is asymptotically optimal in light of the $\Omega(\log n)$ lower bound given in [15] which holds even without considering interference. With the algorithm, nodes make decisions by sensing the signals on the shared channel, rather than decoding the transmissions. Hence, the algorithm is significantly more energy-saving comparing with previous ones that rely on encoding and decoding transmissions. Furthermore, the algorithm requires very little knowledge and weak capability on the nodes. Hence, the algorithm should be quite implementable in realistic networks.

Related Work: In 1970, a randomized solution to the leader election problem was firstly proposed for radio networks [1]. A detailed survey of early works on this topic can be found in [7]. Since then, there has been a great deal of research done in the radio network model where however the interference is simplified to be a local and binary phenomenon [2,3,4,5,8,10,11,12,13]; these works differ in terms of the network topology, node capabilities and time complexity measurements they consider.

There has also been some recent research considering leader election in SINR model. A randomized algorithm in SINR model is proposed in [6] to elect a leader with time complexity of $O(\log n + \log R)$ w.h.p., where R is the transmission range. However, this work assumes a single-hop network with synchronous wakeup nodes and uniform transmission power (i.e., all nodes have the same transmission power). Another work on the SINR model, which treats a multi-hop network with asynchronous wakeup nodes, is proposed in [9] with time complexity of $O(D \log^2 n + \log^3 n)$. Comparing to these two pieces of work in the SINR model, our algorithm elects a leader regardless of the network topology and is faster (improved to the asymptotically optimal time complexity).

2 Model

We model a wireless network in a 2-dimensional Euclidean space, where n nodes are arbitrarily deployed. No requirement is imposed on the topology of network, i.e., it can be a single-hop, multi-hop, connected, or unconnected network, etc. The time is divided into rounds, each of which contains two slots. A slot is the time unit for nodes to send a one-bit signal. The transmissions between nodes in a slot are synchronized.

Communication Model. In each slot, every transmitter (transmitting node) contributes to the signal in the network. The signal from a transmitter fades with distance. It is possible that the signals received by two nodes differ because of their different distances to the transmitter. The SINR model is used to depict the signal received from the transmitter. For any node v, let $Signal(v)$ be the value of signal at v, and $SINR(u, v)$ be the SINR rate at a node v that listens from a transmitter u:

$$\begin{aligned} Signal(v) &= \sum_{w \in S} P_w \cdot d(w, v)^{-\alpha} + \mathcal{N}, \\ SINR(u, v) &= \frac{P_u \cdot d(u, v)^{-\alpha}}{\sum_{w \in S \setminus \{u\}} P_w \cdot d(w, v)^{-\alpha} + \mathcal{N}}. \end{aligned} \tag{1}$$

In the above equation, S is the set of transmitters in current slot; $d(w, v)$ is the Euclidean distance between nodes w and v; P_w is the transmission power of node w; $\alpha \in (2, 6]$ is the path-loss exponent and $\mathcal{N}$ is the ambient noise, both of which are constants determined by the environment. When $SINR(u, v) \geq \beta$, v can decode the message from u, where β is a threshold determined by hardware and larger than 1.

Considering that the ambient noise $\mathcal{N}$ is not always a fixed constant but changing in a slight way in reality, we take N as a close upper bound of ambient noise $\mathcal{N}$ in our model. A good tuning on the value of the upper bound N makes sure that N is always larger than $\mathcal{N}$, and when there are transmissions in network, the accumulation of signals from transmissions and ambient noise on any node is larger than N.

knowledge and Capability of nodes. Asynchronous wakeup mode is assumed for nodes. Each node can wake up at the beginning of any round. The only knowledge each node needs to know is the value of the ambient noise upper bound N. It is not required that the transmission power be uniform, i.e., nodes can have different transmission powers. Also, every node is equipped with a half-duplex transceiver, i.e., a node can transmit or listen on each slot but cannot do both. Physical carrier sensing on each node is needed, but the encode/decode functions are not required. Physical carrier sensing is part of the IEEE 802.11 standard, and is provided by a Clear Channel Assessment (CCA) circuit, which monitors the signal in the channel. A detailed description and usage of physical carrier sensing can be found in [14].

Our constraints on the network topology, nodes' knowledge, and nodes' capability are so simple that the proposed leader election algorithm can be widely applied in many different scenarios.

3 Algorithm description

Algorithm 1: Leader Election

Initialization: $round_v = \mathbb{A}$;

In each round, each node v does:

Slot 1: Candidate Control Slot ( );
Slot 2: Leader Election Slot ( );

Candidate Control Slot ( ):

if $state_v = \mathbb{C}$ **then**
    Transmit message $\mathcal{M}_v$;
if $state_v = \mathbb{A}$ **then**
    Listen;
    if *Received signal is larger than* N **then**
        $state_v = \mathbb{S}$;
    else
        $state_v = \mathbb{C}$;

Leader Election Slot ( ):

if $state_v = \mathbb{C}$ **then**
    Let $X \leftarrow 1$ or 0 with probability p_v and $1 - p_v$, respectively;
    if $X = 1$ **then**
        Transmit message $\mathcal{M}_v$;
    else
        Listen;
        if *Received signal is larger than* N **then**
            $state_v = \mathbb{S}$;

Our simple and optimal leader election (LE) algorithm designed for wireless networks with asynchronous wakeup nodes is laid out in Algorithm 1. Basically, in the algorithm, nodes wake up and try to compete for the leadership; the leader is elected when only one node is left in the leader competition. There are two challenges in designing the LE algorithm. The first one is how to deal with the asynchronous wakeup nodes, the wakeup time of which cannot be predicted. If we do not prevent the nodes that wake up later from the leader competition, the running time of the algorithm will be highly influenced by the final wakeup nodes and it is impossible to get non-trivial results. For example, consider a network with n nodes in total, in which only one node initially wakes up. Assume that at the beginning of each round, there is a node waking up. The final node wakes up at the $(n-1)$-th round. It can be seen that there are always at least two nodes competing for leader during the first $(n-1)$ rounds. Then, it takes $\Omega(n)$ rounds to finish the leader election. Another challenge is how to efficiently reduce the number of nodes in the leader competition to get an optimal result on time complexity.

Each round during algorithm execution contains two slots: the Candidate Control (CC) slot and the Leader Election (LE) slot, which deal with the the two challenges mentioned above, respectively. To handle the nodes that wake up later, we partition the nodes into two categories: the *Candidate* for leader competition and the other nodes. Candidates are the nodes that compete for the leadership. The setting in Candidate Control slot guarantees that only nodes that firstly wake up in network become candidates that will compete for the leadership, to overcome the difficulty of asynchronous wakeups. In the Leader Election slot, when a candidate listens and finds that there are other candidates transmitting, it stops the leader competition. This leader competition will be proved to be asymptotically optimal in terms of time complexity in the analysis.

To be more specific, nodes can be in one of three states: $\mathbb{C}$, $\mathbb{A}$, and $\mathbb{S}$, corresponding to the candidate state, the awake state, and the silent state. The message $\mathcal{M}_v$ transmitted by any node v in the algorithm is a one-bit message. When a node listens, it monitors the signal in network via physical carrier sensing. For any node waking up at the beginning of a round, it is regarded as an awake node in state $\mathbb{A}$.

In the CC slot of every round, each node v in state $\mathbb{C}$ transmits message $\mathcal{M}_v$; and each node u in state $\mathbb{A}$ listens. If the signal sensed by u is larger than N, which means that there are already some candidates, node u in state $\mathbb{A}$ changes its state to $\mathbb{S}$ and keeps silent in subsequent rounds. Otherwise u becomes a candidate in state $\mathbb{C}$. In this way, the CC slot guarantees that only the nodes that firstly wake up become the candidates. Only candidates compete for the leadership in the Leader Election slot of each round.

In the LE slot, each candidate v transmits $\mathcal{M}_v$ with a constant probability p_v, where p_v can be any constant in $(0, 1)$ and can be different for different nodes. The different transmission probabilities will not affect the asymptotical running time bound. If a candidate v listens and senses a signal larger than N, it stops the leader competition, changes its state to $\mathbb{S}$, and keeps silent in subsequent rounds. We use the following description to briefly present the correctness and efficiency of our leader election scheme in LE slot. In reality, it is nearly impossible for nodes in wireless network to distinguish between the interference caused by ambient noise, single transmission and multiple transmissions if without decoding, and we also have this assumption in our work to make it close to reality. But fortunately, we can adopt an alternative way in our scheme for nodes to detect whether there are some nodes transmitting in network. Specifically, according to our assumption, the ambient noise from environment is smaller than the threshold N. Only when there are one or more nodes transmitting, the accumulative interference at any node is larger than the threshold. So, when a listening node finds that its interference is larger than N, it will quit the leader competition, because there must be some nodes who compete for the leader and transmit in the current round. Thus, in each LE slot, the nodes who listen quit the leader competition if there are some nodes transmitting until only one node is left in leader competition.

4 Algorithm analysis

In this section, we analyze the correctness and time complexity of our algorithm. At first, we give a basic lemma which will be frequently used in the analysis.

Lemma 1. *For any node v listening in the network, when there are nodes transmitting simultaneously, the signal sensed by v is larger than N; Otherwise, the signal sensed by v is no larger than N;*

Proof. This is a direct conclusion from Equation 1 and our definition of N in the communication model.

Let C denote the set of nodes that firstly wake up in network, and assume that nodes in C wake up at the beginning of round t. For the candidates that compete for the leadership, we have the following two lemmas.

Lemma 2. *All nodes in C become the candidates.*

Proof. Let v be a node in C. Waking up in state $\mathbb{A}$, v listens in the CC slot of round t. Considering that no nodes wake up earlier than v, and the other nodes in C also listen, no nodes transmit in this slot. The signal sensed by v is no larger than N according to Lemma 1, and v changes its state to $\mathbb{C}$. So, v becomes a candidate in the first slot when it wakes up.

Lemma 3. *Nodes outside C never become a candidate.*

Proof. Let u be a node outside C, waking up at the beginning of round t'. According to the definition of set C, $t < t'$. After u wakes up in state $\mathbb{A}$, it listens in the CC slot of round t'. According to Lemma 2, there are already some candidates transmitting in the CC slot. Hence, the signal sensed by u is larger than N according to Lemma 1. So, u changes its state to $\mathbb{S}$ in the first CC slot after it wakes up, and keeps silent in the subsequent rounds.

The above two lemmas show that all nodes in C become the candidates in the first CC slot when they wake up at round t and no other nodes become candidates after that. Now we only need to consider the reduction of candidates in the Leader Election slots beginning at round t.

The analysis for candidates' reduction in the LE slots is divided into three cases: (1) $|C| = 1$; (2) $|C|$ is a constant larger than 1; (3) $|C|$ is sufficiently large, not a constant any more. If $|C| = 1$, the leader election problem is already solved. For the other two cases, assume that $\{v_1, v_2, \ldots, v_{|C|}\}$ are the candidates in C. Each candidate v has a constant probability p_v to transmit message $\mathcal{M}_c$ in the LE slot. Let $p_{min} = \min_{v\in C}\{p_v\}$, and $p_{max} = \max_{v\in C}\{p_v\}$. Obviously, p_{min} and p_{max} are also constants.

Lemma 4. *When $|C|$ is a constant larger than 1, it takes $O(\log n)$ rounds to elect a leader with high probability.*

Proof. Define $\mathcal{E}_1$ to be the event that all candidates in the network transmit or listen in the current LE slot. It is easy to obtain the probability that $\mathcal{E}_1$ does not occur is no smaller than $1 - p_{max}^{|C|} - (1 - p_{min})^{|C|}$. When $\mathcal{E}_1$ does not occur, according to Lemma 1, at least one node would stop the leader competition. Using the Chernoff bound, after $O(\log n)$ rounds w.h.p., only one candidate will be left in the leader competition, and it becomes the leader.

We next consider the case that $|C|$ is sufficiently large, not a constant any more.

Lemma 5. *For a sufficiently large $|C|$, in each LE slot, a constant fraction of the candidates stop the leader competition with a probability of $1 - e^{-\Omega(|C|)}$.*

Proof. We focus on the number of candidates that listen in the LE slot and define a random variable x_v as follows.

$$x_v = \begin{cases} 0 & \text{when v listens} \\ 1 & \text{when v transmits} \end{cases}$$

Let μ be the expectation of $\sum_{v\in C} x_v$, $\mu_0 = |C| * p_{min}$, $\mu_1 = |C| * p_{max}$. We can get that $\mu = \mathbb{E}\Big[\sum_{v\in C} x_v\Big] = \sum_{v\in C} p_v$, and $\mu_0 \le \mu \le \mu_1$.

Applying the Chernoff bound with a constant $\delta \in (0, \min\{1/p_{max} - 1, 1\})$:

$$Pr(\sum_{v\in C} x_v \geq (1+\delta)\mu_1) \leq Pr(\sum_{v\in C} x_v \geq (1+\delta)\mu) \leq e^{-\frac{\delta^2\mu}{3}},$$

$$Pr(\sum_{v\in C} x_v \leq (1-\delta)\mu_0) \leq Pr(\sum_{v\in C} x_v \leq (1-\delta)\mu) \leq e^{-\frac{\delta^2\mu}{3}}.$$

So, with probability $1-2\cdot e^{-\frac{\delta^2\mu}{3}}$, $(1-\delta)p_{min}$ fraction of the candidates transmit and $1-(1+\delta)p_{max}$ fraction of the candidates listen. According to Lemma 1, $1-(1+\delta)p_{max}$ fraction of the candidates would stop the leader competition with a probability of $1-2\cdot e^{-\frac{\delta^2\mu}{3}}$.

Lemma 6. *If $|C|$ is sufficiently large, it takes $O(\log n)$ rounds to elect a leader with high probability.*

Proof. According to Lemma 5, a constant fraction of the candidates would stop the leader competition in each round with probability of $1-e^{-\Omega(|C|)}$. Using the Chernoff bound, after $O(\log n)$ rounds, only one candidate will be left in the leader competition w.h.p., and it then becomes the leader.

Combining the three cases above, we can get the time complexity of our leader election algorithm.

Theorem 1. *After nodes wake up for $\Theta(\log n)$ rounds, a leader can be selected with high probability.*

5 Simulation result

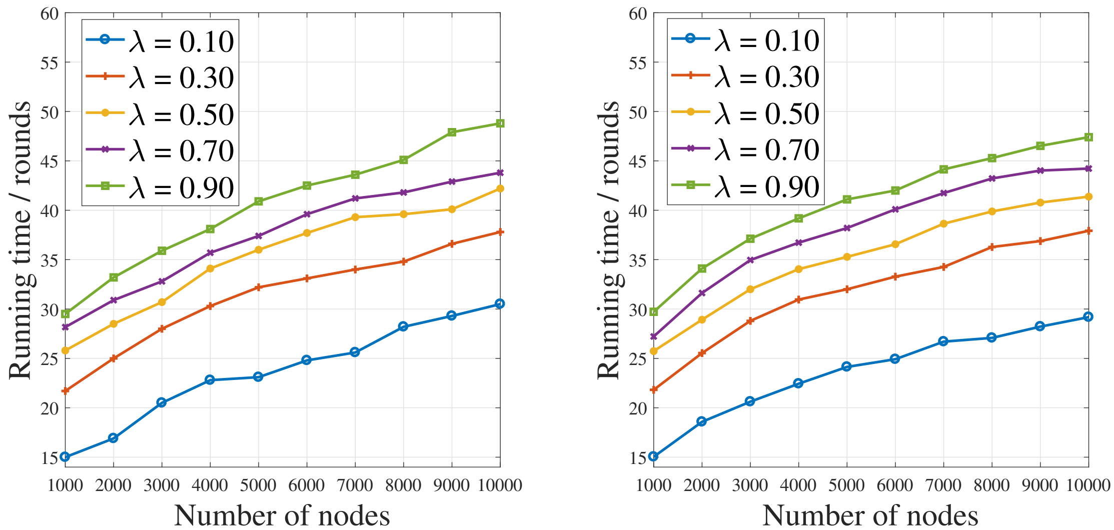
In this section, we present the empirical performance of our leader election algorithm. Specifically, we focus on the time used for leader election (i) in networks with different sizes, and (ii) in networks with different topologies, i.e. single-/multi-hop networks. Also, we report on the observed number of candidates in the leader election process.

Parameter setting. In the simulation, n nodes are randomly and uniformly distributed in a network with size of $300m \times 300m$, the minimum distance between nodes is $1m$. The constant λ is used to depict nodes' asynchronous waking up. It is assumed that λ fraction of nodes initially wake up, and the other nodes randomly wake up in the following 50 rounds, as we have observed that our algorithm always ends within 50 rounds. The ambient noise upper bound N is normalized as 1.0. P_v and p_v are the power and probability of node v in transmission, both of which are random variables uniformly selected from their respective intervals. P_v has two intervals, the first (denoted as P_v^s) and the second (denoted as P_v^m) of which (in Tab. 1) correspond to the single-hop and the multi-hop networks respectively. Tab 1 summarizes the parameters given above.

Table 1: Parameters in simulation

Parameter	Value	Parameter	Value
N	1.0	n	$\in \{1,2,...,10\} * 10^3$
α	3.0	λ	$\in \{1,3,5,7,9\} * 10^{-1}$
β	2.0	P_v^s	$\in [(300\sqrt{2})^\alpha * \beta N, 600^\alpha * \beta N]$
p_v	$\in [0.10, 0.90]$	P_v^m	$\in [\beta N, (300\sqrt{2})^\alpha * \beta N)$

Algorithm Performance. The time our algorithm used for leader election is given in Fig. 1, in which the x-axis and y-axis represent the network size n and number of rounds used for leader election respectively. From Fig. 1 (a), which depicts the scenario in a single-hop network, we can see that when n is fixed and λ gets larger or λ is fixed and n gets larger, the time for leader election increases. This phenomenon indicates



(a) Single-hop network

(b) multi-hop network

Fig. 1: Time used for leader election

that it is the number of nodes that firstly wake up in network determines the running time of our leader election algorithm. Even in the case that $n = 10000$ and $\lambda = 0.90$, which means 9000 nodes initially wake up, our algorithm can elect a leader within 50 rounds. In Fig. 1 (b), we get a similar conclusion in multi-hop networks. Besides, the performances of our algorithm are similar in single-hop and multi-hop networks, which confirms that our algorithm works well regardless of the network topology.

In our algorithm, the firstly wake up nodes become the candidates, and compete for the leadership. To illustrate the leader competition process in details, fixing $n = 5000$, we show the number of candidates in each round in Fig. 2, in which the x-axis and the y-axis represent the number of rounds and candidates respectively. Fig. 2 shows that in single-/multi-hop networks, a constant fraction of candidates are reduced in each rounds, which verifies our analysis.

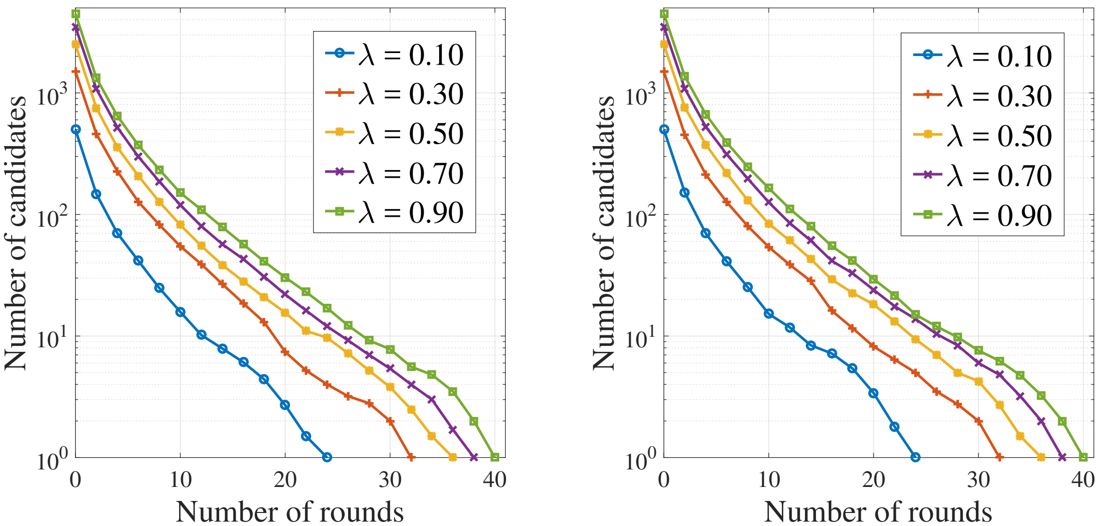
6 Conclusion

In this paper, we present a simple randomized and distributed algorithm which elects a leader in time that is asymptotically optimal. The algorithm fully makes use of physical carrier sensing, a standard function in most wireless devices, and it avoids transmission encoding and decoding to achieve efficiency and save energy. Our algorithm can be readily implementable in a wide range of real networks, as it requires very little knowledge and weak capability on the nodes.

Our algorithm states that interference sometimes is not necessarily always a negative factor, and can actually be used to facilitate efficient algorithm design. It will be interesting to further explore such usage in designing distributed algorithms for other fundamental problems, such as consensus and backbone network construction.

Acknowledgement

This work is supported by National Key R&D Program of China (No. 2018YFB0204005), NSFC (No. 61433012, 61971269, 61832012, 61672321, 61771289, 61702304), Shandong Provincial Natural Science Foun-



(a) Single-hop network (b) multi-hop network

Fig. 2: Reduction of candidates in leader election

dation (No. ZR2017QF005), Shenzhen research grant (KQJSCX20180330170311901, JCYJ20180305180840138 and GGFW2017073114031767), Hong Kong GRF 17210017 and Shenzhen Discipline Construction Project for Urban Computing and Data Intelligence.

References

1. N. Abramson. The ALOHA System: Another Alternative for Computer Communications. In *AFIPS FJCC*, 1970.
2. R. Bar-Yehuda, O. Goldreich, and A. Itai. Efficient emulation of single-hop radio network with collision on multi-hop radio network with no collision detection. In *Distributed Computing*, 5: 67-71, 1991.
3. B.S. Chlebus, D.R. Kowalski, and A. Pelc. Electing a leader in multi-hop radio networks. In *OPODIS*, 2012.
4. A. Czumaj, and P. Davies. Brief Announcement: Optimal Leader Election in Multi-Hop Radio Networks. In *PODC*, 2016.
5. A. Czumaj and W. Rytter. Broadcasting algorithms in radio networks with unknown topology. In *FOCS*, 2003.
6. J.T. Fineman, S. Gilbert, F. Kuhn, and C.C. Newport. Contention Resolution on a Fading Channel. In *PODC*, 2016.
7. R. Gallager. A Perspective on Multiaccess Channels. In *IEEE Trans. Information Theory*, 31(2): 124-142, 1985.
8. M. Ghaffari and B. Haeupler. Near optimal leader election in multi-hop radio networks. In *SODA*, 2013.
9. T. Jurdzinski, D.R. Kowalski, M. Rozanski, and G. Stachowiak. On setting-up asynchronous ad hoc wireless networks. In *INFOCOM*, 2015.
10. T. Jurdzinski and G. Stachowiak. Probabilistic algorithms for the wake-up problem in single-hop radio networks. In *Theory of Computing Systems*, 38(3): 347-367, 2005.
11. D. Kowalski and A. Pelc. Broadcasting in undirected ad hoc radio networks. In *PODC*, 2003.
12. E. Kushilevitz and Y. Mansour. An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks. In *SIAM Journal on Computing*, 27(3): 702-712, 1998.
13. C. Newport. Radio Network Lower Bounds Made Easy. In *DISC*, 2014.
14. C. Scheideler, A.W. Richa, and P. Santi, An $O(\log n)$ dominating set protocol for wireless ad-hoc networks under the physical interference model. In *MobiHoc*, 2008.
15. J. Schneider and R. Wattenhofer. What Is the Use of Collision Detection (in Wireless Networks)? In *DISC*, 2010.