SIAM J. COMPUT.
Vol. 44, No. 3, pp. 868–888

FAST NONADAPTIVE DETERMINISTIC ALGORITHM FOR CONFLICT RESOLUTION IN A DYNAMIC MULTIPLE-ACCESS CHANNEL*

GIANLUCA DE MARCO† AND DARIUSZ R. KOWALSKI‡

Abstract. A classical problem in addressing a decentralized multiple-access channel is resolving conflicts when a set of stations attempt to transmit at the same time on a shared communication channel. In a static scenario, i.e., when all stations are activated simultaneously, Komlós and Greenberg [*IEEE Trans. Inform. Theory*, 31 (1985), pp. 302–306] in their seminal work showed that it is possible to resolve the conflict among k stations from an ensemble of n, with a nonadaptive deterministic algorithm in time $O(k + k \log(n/k))$ in the worst case. In this paper we show that in a *dynamic scenario*, when the stations can join the channel at arbitrary rounds, there is a *nonadaptive* deterministic algorithm guaranteeing a successful transmission for each station in only a slightly bigger time: $O(k \log n \log \log n)$ in the worst case. This almost matches the $\Omega(k \log n/ \log k)$ lower bound by Greenberg and Winograd [*J. ACM*, 32 (1985), pp. 589–596] that holds even in much stronger settings: for *adaptive* algorithms, in the *static* scenario, and with additional *channel feedback–collision detection*. In terms of channel utilization, our result implies throughput, understood as the average number of successful transmissions per time unit, $\Omega(1/(\log n \log \log n))$ on the dynamic deterministic channel.

Key words. multiple-access channel, contention resolution, deterministic algorithms, distributed algorithms, latency, throughput

AMS subject classifications. 68W15, 68W40

DOI. 10.1137/140982763

1. Introduction.

1.1. Problem and previous work.

A set of stations labeled $1, 2, \ldots, n$ are connected to a multiple-access channel. A subset of $k \leq n$ stations have data packets and can transmit on the channel at rounds (also called time steps) numbered $1, 2, \ldots$ and measured by a global clock accessible by each station. If $m \leq k$ stations transmit at the same time, then the result of the transmission depends on m as follows:

- If $m = 0$, no station transmits and of course no packet is transmitted.
- If $m = 1$, the packet owned by the transmitting station is successfully sent to every station.
- If $m > 1$, a collision occurs (the simultaneous transmissions interfere with one another) and as a result no packet is successfully transmitted.

There is no central control: every station acts autonomously by means of a distributed algorithm. The aim is to let each of the k stations transmit successfully its packet. By *conflict resolution algorithm* in this paper we mean a distributed algorithm that schedules the transmissions for each of the k participating stations guaranteeing that every station eventually transmits individually (i.e., without interfering with others) on the channel.

[*] Received by the editors August 18, 2014; accepted for publication (in revised form) April 23, 2015; published electronically June 30, 2015.
http://www.siam.org/journals/sicomp/44-3/98276.html

[†] Dipartimento di Informatica, Università di Salerno, 84084 Fisciano (SA), Italy (demarco@dia.unisa.it).

[‡] Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK (d.kowalski@liverpool.ac.uk).

If the stations have a *collision detection* mechanism, which is the case not studied in this paper, then at each round they receive the feedback 0, 1, or 2+ from the channel, indicating, respectively, that m is 0, 1, or ≥ 2. This way it is possible to design an *adaptive* algorithm in which the behavior of any station at any given time may depend on the feedback received in previous rounds. Namely, a deterministic adaptive algorithm, at each step of its execution, specifies some subset of the n possible stations chosen as a function of the feedback obtained in previous steps. Any station that has not yet transmitted successfully, checks the chosen subset and transmits if and only if it belongs to it. For dynamic stochastic packet injection, Capetanakis [6], Hayes [24], and Tsybakov and Mikhailov [33] independently presented a deterministic tree algorithm for conflict resolution with worst-case time complexity $O(k+k\log(n/k))$ for every k and n. The worst case refers to the maximum number of rounds over all possible choices of the subset of k stations. Greenberg and Winograd [21] proved that the tree algorithm is close to optimal. They formalized a general framework of deterministic algorithms to resolve conflicts and showed that any deterministic algorithm needs $\Omega(k(\log n)/(\log k))$ rounds, in the worst case, to resolve conflicts among k stations out of n possible stations, with $2 \leq k \leq n$. The randomized version of the problem has also been studied and the current best randomized algorithm resolves conflicts without error in expected time $2.14k + O(\log k)$ [20, 19].

When collision detection is not available, which is the case considered in the present paper, the conflict resolution algorithm has even less power. However, it turns out that even simple nonadaptive solutions might be efficient. In nonadaptive algorithms, the only feedback a station must adapt to is when it actually transmits successfully, in which case it switches off. Precisely, a nonadaptive algorithm produces a list of transmission sets (queries), $Q_1, Q_2, \ldots, Q_s$ as a function of k and n only. At any round t, any station examines the list and transmits if and only if $x \in Q_t$ (provided that none of its previous transmissions succeeded). Among many advantages of nonadaptive algorithms are fast local processing (once the transmission schedules are implemented), higher resiliency, and independence on collision detection and many other physical parameters. Yet, as we show in this work, they can be very efficient.

Komlós and Greenberg [28] showed that there is a nonadaptive conflict resolution algorithm that, for all k and n such that $2 \leq k \leq n$, generates only $O(k + k\log(n/k))$ queries, surprisingly the same asymptotic number as the adaptive tree algorithm. The proof is nonconstructive and is obtained using probabilistic methods. Later Kowalski [29] showed a more constructive solution, based on selectors (cf., [11, 27]), reaching the same asymptotic bound. Recently, Chlebus, Kowalski, and Rokick [10] developed two deterministic conflict-resolution algorithms for a specific subclass of wake-up patterns, mainly, assuming that the number of awaken stations in time intervals of a given length is arbitrarily bounded. The first of them accomplishes the task in $O(k\log^2 n)$ rounds if the global clock is available, while the second one has time complexity independent on n that could be as large as $O(n\log^2 n)$ if the global clock is not available.

For recent results on randomized solutions one can refer to the paper by Fernandez Anta, Mosteiro, and Munoz [17].

1.2. Our contribution. All the above mentioned papers on conflict resolution assumed that either the k stations are all activated at the beginning and therefore can start simultaneously their transmitting schedules or the wake-up pattern is restricted yet they still could guarantee only $O(k\log^2 n)$ time complexity or worse. While the static scenario is particularly advantageous for designing algorithms, it is typically not attainable without a central control. We therefore focus on a *dynamic* scenario in which each station is totally independent, and consequently its activation time is locally determined and cannot be known or predicted by other stations. In such a situation, it would be desirable to have an algorithm that allows every station to transmit successfully, independently of its activation time. Since the activation times can be arbitrarily distant from each other, it is natural to measure the efficiency in terms of latency, i.e., the number of rounds necessary for a station to transmit successfully, measured since its wake-up time.

Our contribution is to show that there is a deterministic nonadaptive conflict resolution algorithm that, for all k and n, $2 \leq k \leq n$, and *for all possible activation times*, allows k stations from an ensemble of n to transmit successfully in only $O(k \log n \log\log n)$ rounds. The algorithm works without any a priori information about k. This is very close to the Komlós and Greenberg result, despite its more general setting, and it is nearly optimal in view of the $\Omega(k \log_k n)$ lower bound [21] established in a much more favorable (from the perspective of algorithms) framework: *adaptive* algorithms, *simultaneous* activations, and additional *channel feedback* about collisions. Our result also substantially outperforms the previous solutions for restricted wake-up pattern, which guarantee only $O(k \log^2 n)$ time complexity [10]. In terms of channel utilization, our algorithm guarantees an average number of successful transmissions per time unit of $\Omega(1/(\log n \log\log n))$. As in [28], our proof is nonconstructive and is obtained using the probabilistic method [2].

Apart from the result itself, this paper introduces a novel approach to design and analysis of contention resolution protocols, which can be summarized as follows. Our main algorithm is obtained by interleaving two algorithms serving different purposes: `contention-reduction` and `iterative-selection`. The goal of the first algorithm is to reduce the channel stale-contention, which is at most k at the beginning, to at most $k/\log n$; here, by *stale-contention*, we mean the number of stations that are "long enough" on the channel, i.e., that already reached the asymptotic latency that we want to achieve. This is done by analyzing properties of specific transmission schedules, mainly, those which schedule transmissions with decreasing frequencies, with additional transmissions occurring in consecutive periods of length $\Theta(\log\log n)$. We show that there are transmission schedules such that for any wake-up pattern of some k stations, starting from some point, the number of collisions will be limited, while the additional transmissions in $\Theta(\log\log n)$-length periods will increase chances of successful transmission to an appropriate level.

As we will see in the analysis, this is a big challenge with respect to the static case, as the algorithm does not have any control on newly awakened stations, which number can congest the channel at any moment. As announced earlier, we use probabilistic arguments over carefully defined classes of wake-up patterns to prove that such transmission schedules exist.

The goal of algorithm `iterative-selection` is to efficiently and distributedly schedule transmissions so that, under the reduced stale-contention, all stations are successful in the desired time $O(k \log n \log\log n)$. This is done by using the Komlós and Greenberg protocol [28], run in a logarithmic number of copies: the ith copy tries to solve the contention resolution problem for the (reduced) stale-contention 2^i. Stations join the ith copy of the protocol with some delay, in order to synchronize with the first `contention-reduction` protocol and to avoid a high contention (i.e., giving a chance to most of the stations to be successfully selected by the first protocol).

1.3. Related problems. In this paper we have used the definition of nonadaptive algorithm given by Komlós and Greenberg [28], in which the behavior of the stations has no dependence on feedback, but a station must adapt to the feedback produced when it successfully transmits (in the sense that it switches off). Although this implies that a collision at a given round t can be detected by the stations transmitting at round t, the model is different from the one where *collision detection* is a feature of the system allowing *all* participating stations to distinguish the noise heard in the case of two or more stations transmitting from the background noise in case of no transmission. There is an even more restricted definition of nonadaptiveness in the literature (*strong nonadaptiveness*), where transmissions are not influenced by any feedback at all. This means that stations cannot switch off but continue to interfere even after a successful transmission. This is known in combinatorial search theory as the problem of constructing minimum length superimposed codes [14, 25]. Formally, the problem is to produce, given k and n, with $2 \leq k \leq n$, the shortest sequence of queries $Q_1, Q_2, \ldots, Q_s$ such that for every subset $S \subseteq \{1, 2, \ldots . n\}$, with $|S| = k$, the following property holds: for all $x \in S$, there exists a query Q in the sequence such that $S \cap Q = \{x\}$. Of course, the sequence of all singletons $\{1\}, \{2\}, \ldots, \{n\}$ solves the problem for any k and n. A very interesting result by Bassalygo (cf. [14]) is that this is optimal when $k \geq \sqrt{2n} + 1$. In general, it is well known that $O\big(\min(n, k^2 + k^2 \log(n/k))\big)$ queries suffice for all k and n.

De Bonis, Gasieniec, and Vaccaro [15] considered the problem of *partial contention resolution*, i.e., when all but at most r contenders need to be successfully selected on the channel, for some given parameter $0 \leq r < k$. They proved a general lower bound $\Omega\big(\min\big(n, \frac{(r-1)^2}{k-r+1} \cdot \frac{\log(n/(k-r+1))}{\log((r-1)/(k-r+1))}\big)\big)$ for the number of rounds and showed the existence of a protocol accomplishing the task in $O\big(\frac{k^2}{k-r+1} \log(n/k)\big)$ rounds. Both results are for strongly nonadaptive algorithms. Indyk [27] showed a polynomial-time construction of such nonadaptive transmission schedules for $r = \Omega(k)$, with a slightly increased number of rounds by a polylogarithmic factor. This construction was later generalized in [9] to any $0 \leq r < k$, to match, up to a polylogarithmic factor, the lower bound in [15].

A broadcast from a synchronized start in a radio network was considered in [8, 11, 12, 13, 16, 29].

It is worth stressing that all the above mentioned results for strongly nonadaptive algorithms hold in the *static* model; therefore our paper is the first to study the contention resolution problem without any restriction on the activation times of the stations.

Backoff protocols. When some restrictions on packet arrivals are assumed, a very popular method of choice is the so-called backoff protocol. The idea of backoff is that when a collision occurs in a given time slot, the packets are retransmitted in the subsequent time slots with a diminished probability of transmission. When the probability decreases polynomially or exponentially the method is called *polynomial* or *exponential backoff*, respectively. This has been studied mainly under the statistical queuing-theory model (when packet arrivals are determined by a Poisson distribution) [31, 18, 26, 30]. For a worst-case adversarial approach the reader can consult the more recent work by Bender at al. [4] and the references therein.

Channel with jamming. Another related line of research is the one that contemplates the possibility that time slots can be *jammed*. Awerbuch, Richa, and Scheideler [3] studied jamming in multiple-access channels in an adversarial setting and gave an estimation of the saturation throughput for randomized protocols. For an account of the literature on adversarial models the interested reader can consult Richa and Scheideler [32]. For arbitrary jamming models we refer the reader to the works by Alistarh et al. [1] and Gilbert, Guerraoui, and Newport et al. [22]. Energy efficiency approaches can be found in [23].

2. Algorithm `conflict-resolution`.

Throughout the paper, a station that has been woken up and has not yet transmitted successfully will be called *active*. Since we are interested in the asymptotic bound of our algorithm, we omit all the floor and ceiling signs assuming that all derived numbers are integers. For the same reason, we will also simplify the analysis by assuming that n is sufficiently large.

The main algorithm, called `conflict-resolution`, is obtained by interleaving two algorithms, called `contention-reduction` and `iterative-selection`. They can be conceptually understood, without loss of generality for our asymptotic result, as run "in parallel," although the formal analysis should be done rigorously in the model when they are interleaved in odd-even rounds. The goal of the former algorithm is to reduce the stale-contention, which is at most k at the beginning, to at most $k/\log n$; here, informally, by stale-contention we mean the number of stations that are "long enough" on the channel. The goal of the latter algorithm is to efficiently and distributedly schedule transmissions so that, under the reduced stale-contention, all stations are successful in the desired time $O(k \log n \log\log n)$. We first describe each of these two algorithms separately, and then we conclude how they collaborate to achieve the goals mentioned above. The analysis will be given later in section 3.

Algorithm 1. `contention-reduction`(u, σ).

```
Require: a transmission matrix $\mathcal{M}$ (identical for every station)
    ▷ $t$ denotes the current round number measured by the global clock
$\ell_0 \leftarrow 0$
$\tau_0 \leftarrow \mu(\sigma)$
while $u$ is active do
    for $i = 1$ to $\log n$ do
        $\tau_i \leftarrow \tau_{i-1} + \ell_{i-1}$    ▷ $\tau_i$ is the time at which $u$ starts scanning row $i$ of $\mathcal{M}$
        for $t = \tau_i$ to $\tau_i + \ell_i - 1$ do
            $j \leftarrow t \mod \ell$    ▷ $\mathcal{M}$ is scanned circularly
            if $u \in M_{i,j}$ then
                transmit at time $t$    ▷ $u$ transmits at round $t$ iff its ID $\in M_{i,t \mod \ell}$
            end if
            if transmission is successful then
                switch-off
            end if
        end for
    end for
end while
```

2.1. Algorithm `contention-reduction`.

In this algorithm, every station u proceeds according to a *transmission schedule* defined by a matrix whose entries are subsets of stations.

DEFINITION 2.1 (transmission set). *A subset $S \subseteq [n] = \{1, 2, \dots, n\}$ of station IDs will be called a* transmission set.

We say that a station u transmits *according to a transmission set S* if and only if $u \in S$. Note that a sequence of transmission sets $S_1, \dots, S_t$ defines a transmission schedule for any station; indeed, any station checks whether it belongs to set S_j, for $1 \leq j \leq k$, and decides to transmit or stay silent accordingly.

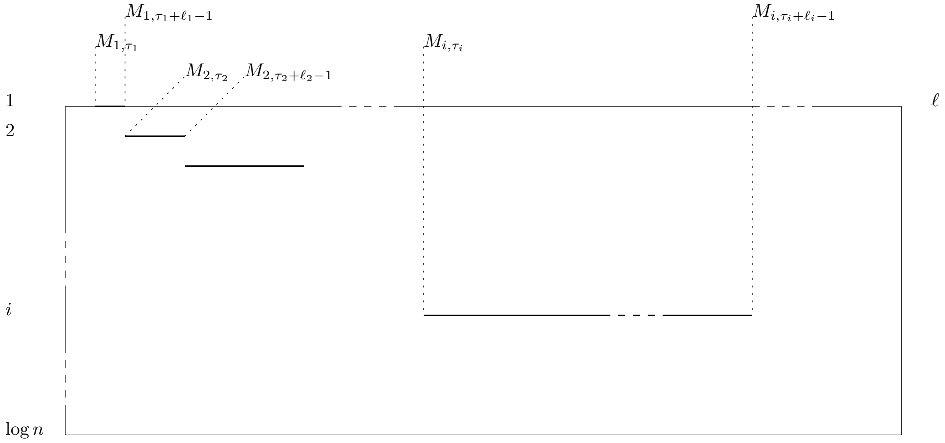

FIG. 1. *A graphical representation of the transmission sets of a* $(\log n \times \ell)$ *transmission matrix* $\mathcal{M}$*, according to which a station* u*, woken up at some time* σ*, transmits. The segments represent contiguous transmission sets on the same row. For the sake of simplicity, we have omitted the modulo* ℓ *on the column indices.*

DEFINITION 2.2 (transmission matrix). *A* transmission matrix $\mathcal{M}$ *is a* $(\log n \times \ell)$ *matrix, for some parameter* ℓ *called the* length of the transmission matrix*, whose entries* $M_{i,j}$*, for* $1 \leq i \leq \log n$ *and* $1 \leq j \leq \ell$*, are transmission sets.*

Every station executing the algorithm will be provided with the same transmission matrix. In what follows, we give a description of the protocol executed by each station. It will be based on a generic transmission matrix. In the next section we will introduce the *contention-reducing matrix*, a particular kind of transmission matrix that exhibits some combinatorial properties that guarantee the proper functioning of the protocol.

The task of the protocol for each station is simply to match rounds with transmission sets on the matrix. For each round t there will be a correspondent transmission set on the matrix according with which the station will transmit at round t. Let $c > 0$ be a sufficiently large constant.

For $i = 1, 2, \ldots, \log n$, let $\ell_i = c \cdot 2^{i+1} \log n \log\log n$ denote the number of consecutive rounds during which a station transmits according to transmission sets in row i of $\mathcal{M}$. Starting from its wake-up time, any station transmits first for ℓ_1 rounds (according to ℓ_1 consecutive transmission sets in row 1 of the matrix), then for ℓ_2 rounds (according to ℓ_2 consecutive transmission sets in row 2 of the matrix), and so on, scanning the matrix circularly.

Let us now describe more formally the protocol executed by an arbitrary station u waking up at some time σ (see Protocol 1 and Figure 1). For any $j > 0$, define $\mu(j) = \min\{b \geq j : b \equiv 0 \mod 2 \log\log n\}$. Let $\ell_0 = 0$ and $\tau_0 = \mu(\sigma)$. Station u has the following transmitting behavior. It remains silent until round $\tau_1 = \tau_0 + \ell_0 = \mu(\sigma)$ and then starts transmitting according to transmission sets $M_{1,\tau_1 \mod \ell}, \ldots, M_{1,(\tau_1+\ell_1-1) \mod \ell}$ (row 1 of $\mathcal{M}$) for rounds $\tau_1, \ldots, \tau_1 + \ell_1 - 1$, respectively. This ends the first iteration of the for loop on line 4.

Then, from time $\tau_2 = \tau_1 + \ell_1 = \mu(\sigma) + \ell_1$ it starts transmitting according to row 2 of $\mathcal{M}$. Namely, it transmits according to $M_{2,\tau_2 \mod \ell}, \ldots, M_{2,(\tau_2+\ell_2-1) \mod \ell}$ for rounds $\tau_2, \ldots, \tau_2 + \ell_2 - 1$, respectively. This ends the second iteration of the for loop on line 4. And so on until a successful transmission. When a successful transmission occurs it shuts down (line 12) and exits the while loop.

2.2. Algorithm `iterative-selection`. The following algorithm is based on the Komlós–Greenberg algorithm [28]. For given parameters n and i, $1 \leq i \leq \log n$, the Komlós–Greenberg algorithm guarantees that if at most 2^i stations (out of n) start the protocol simultaneously at some round, all of them will transmit successfully (and therefore switch off) within $O(2^i + 2^i \log(n/2^i))$ rounds. (Repeating the execution of the algorithm for increasing values of i will guarantee the correctness for any *unknown* number of stations.)

As mentioned in the introduction, it was proved by probabilistic methods (a constructive approach can be found in [29]).

Let $(n, 2^i)$-KG denote the Komlós–Greenberg algorithm for parameters n and i and let m_i be its time complexity (i.e., the number of rounds in the worst case). In [28] the existence of a constant c' has been proved such that the time complexity $m_i = c'(2^i + 2^i \log(n/2^i)) = O(2^i + 2^i \log(n/2^i))$. This same hidden constant is used in our algorithm to express the exact number of rounds required by the Komlós–Greenberg algorithm for parameters n and i.

Let $\Gamma_k = \sum_{i \leq \log k} \ell_i = \Theta(k \log n \log\log n)$ be the time after which a station switches from row $\log k$ to row $\log k + 1$ in protocol `contention-reduction`. Any station u waking up at some round σ executes protocol `iterative-selection`.

Algorithm 2. `iterative-selection`(u, σ).

```
    ▷ $t$ denotes the current round number measured by the global clock
 1: while $u$ is active do
 2:     for $i = 1$ to $\log n$ do                                            ▷ thread $i$
 3:         $j \leftarrow \lfloor t/\log n \rfloor \mod m_i$          ▷ round $t$ corresponds to step $j$ in $(n, 2^i)$-KG
 4:         $t' \leftarrow$ smallest integer such that $t' \geq \Gamma_{2^i \log n} + \sigma$ and $t' = 0 \mod m_i \log n$
 5:         if $t \geq t'$ then
 6:             perform step $j$ of $(n, 2^i)$-KG                   ▷ $(n, 2^i)$-KG run circularly
 7:         end if
 8:         if transmission is successful then
 9:             switch-off
10:         end if
11:     end for
12: end while
```

Analogously to the previous protocol, the task here is to match rounds with steps of algorithm $(n, 2^i)$-KG for parameters n and $1 \leq i \leq \log n$. The for loop on line 2 cyclically runs $\log n$ threads until the station switches off. Each thread i, for $1 \leq i \leq \log n$, corresponds to a step of algorithm $(n, 2^i)$-KG. Namely, for each thread i and round t any active (i.e., not switched off) station runs the jth step, for $j = \lfloor t/\log n \rfloor \mod m_i$, of algorithm $(n, 2^i)$-KG only if the current round number $t \geq t'$ (line 5 of the pseudocode) for t' being the smallest integer such that

1. $t' \geq \Gamma_{2^i \log n} + \sigma$ and
2. $t' = 0 \mod m_i \log n$.

This condition on line 5 becomes true (and will continue to hold till the end, by monotonicity) at the first round $t' = 0 \mod m_i \log n$ such that t' is at least $\Gamma_{2^i \log n}$ rounds after time σ at which u has been woken up. For any station u executing the protocol, conditions 1 and 2 ensure, respectively, that u has switched to some row $r > 2^i \log n$ in protocol `contention-reduction` (run in parallel) and every execution of algorithm $(n, 2^i)$-KG starts from the beginning, i.e., at step 0 of algorithm $(n, 2^i)$-KG (note that $(n, 2^i)$-KG consists of m_i steps, each of them executed every $\log n$ rounds). If the condition on line 5 is satisfied we say that *the station participates in thread i of algorithm* `iterative-selection`. This terminology will be used in the analysis of the algorithm (subsection 3.3).

3. Analysis of the main algorithm. The analysis of our main algorithm `conflict-resolution` is carried out as follows. In the next subsection we present an outline of the analysis that will also highlight, at a high level, some properties of the two interleaved algorithms before their rigorous study. In the subsequent two subsections we present properties of the two interleaved algorithms `contention-reduction` and `iterative-selection`. In the final subsection we use these properties to derive our main result.

3.1. Our approach. Throughout the paper we will say that a station is isolated to mean that it can transmit singly (and therefore successfully) to the channel. Before starting our high-level overview of the analysis of the main algorithm, observe that we could restrict the analysis to the case when $k < n/2$; otherwise, a simple interleaving of algorithm `conflict-resolution` with the round-robin transmission schedule could guarantee isolation of all stations in $O(n) = O(k)$ rounds.

The goal of the first ingredient of the main algorithm `conflict-resolution`, protocol `contention-reduction`, is to reduce the contention of stations that stay on the channel for a "long" time by a logarithmic factor, i.e., to $O(k/\log n)$. Note that the algorithm does not have any control on the number of newly awakened stations; therefore it is impossible to ensure such a logarithmic reduction for *all* awakened stations. Fortunately, stations that are on the channel for a "short" time do not burst the worst-case latency (at least not shortly after awakening), and therefore we can afford to wait until they become "mature." The key concept in the analysis of protocol `contention-reduction` is the notion of saturated intervals. In short, an interval containing a large enough number, i.e., $\Omega(k \log n)$, of segments of lengths $2 \log\log n$ in the beginning of which there are many stations (i.e., more than $k/\log n$) executing row $\log k$ of the matrix and not so many stations (i.e., at most $2k/\log n$) that already passed that row (cf. Definitions 3.4 and 3.5). Intuitively, what we prove is that many stations in row $\log k$ of the matrix create a chance of successful isolation of some station, while we still control the number of stations that are "long" on the channel, that is, that are executing rows bigger than $\log k$.

The latter stations are simultaneously dealt with by algorithm `iterative-selection` mainly, by thread $\log(k/\log n)+1$ in which algorithm $(n, 2k/\log n)$-KG of length $O(k)$ is circularly executed. We show in Lemma 3.9 of section 3.3 that none of these stations stays longer than $O(k \log n \log\log n)$ on the channel after passing row $\log k$ in the matrix and shortly after starting participation in thread $\log(k/\log n) + 1$ of protocol `iterative-selection`.

It is important to emphasise that neither of the two algorithmic ingredients of the main algorithm `conflict-resolution` could do its desired task on its own; instead, a strong implicit collaboration between them takes place. On one hand, protocol `contention-reduction` reduces the stale-contention, i.e., the number of "old" stations to allow the other protocol `iterative-selection` to isolate the remaining ones quickly. On the other hand, by doing so, protocol `iterative-selection` guarantees the second condition of the saturation property of protocol `contention-reduction`, i.e., that there are at most $2k/\log n$ stations that already passed row $\log k$ of the matrix (cf. Definition 3.4). This synergy is crucial for obtaining the final result.

Let us briefly say why saturated segments and intervals are so important. Mainly, because as we showed in the sequence of Lemmas 3.6, 3.2, and 3.8 in section 3.2, in each of them we can find a round with constant probability of successful transmission, and thus we obtain $\Omega(k \log n)$ chances, each with a constant probability, to successfully isolate a station. With this conclusion, we are ready to apply probabilistic arguments.

More precisely, in order to conclude the analysis, in section 3.4 we prove that because of the large number of saturated intervals, for fixed parameter k, the wake-up pattern, and sets $A(j)$ of stations active in rounds j, the probability that the latency will be asymptotically above $k \log n \log\log n$ is exponentially small in the number of saturated segments; cf., Lemma 3.10. Since the number of possible configurations of parameter k, the wake-up pattern, and sets of active stations $A(j)$ in rounds j is exponential in $O(k \log n)$, by using the union bound we obtain that the probability of a configuration with large latency existing is small; cf. Lemma 3.11. Using the probabilistic method and properties of protocol `iterative-selection` expressed in Lemma 3.9, we derive the final theorem about a matrix existing such that using it for instantiation of `contention-reduction` in the main algorithm `conflict-resolution` guarantees latency $O(k \log n \log\log n)$.

3.2. Properties of algorithm `contention-reduction`. Given a parameter $1 \leq i \leq \log n$ and a round j, we denote by $A_{i,j}$ the set of active stations that at time j transmit according to transmission set $M_{i,j \mod \ell}$. Let $A(t)$ be the set of active stations at round t. The main effect of the nonsimultaneous starting of the k stations is that each station in $A(t)$, when following protocol `contention-reduction`, may transmit according to transmission sets located in different rows of $\mathcal{M}$, depending on the time at which it was woken up. Indeed, set $A(t)$ can be partitioned using sets $A_{i,j}$ in the following way:

$$\begin{cases} A(t) = \bigcup_{i=1}^{\log n} A_{i,t}, & \\ A_{i,t} \cap A_{i',t} = \emptyset & \text{for every pair } (i,i') \text{ with } i \neq i'. \end{cases}$$

In other words, at every round t all active stations transmit according to transmission sets that may be in different rows of $\mathcal{M}$, but that are *vertically aligned* on $\mathcal{M}$, i.e., they are in the same column $j = t \mod \ell$ of matrix $\mathcal{M}$.

We say that a station $w \in A_{i,j} \subseteq A(t)$ is *isolated* at time j if and only if

$$\bigcup_{i=1}^{\log n} (A_{i,j} \cap M_{i,j \mod \ell}) = \{w\}.$$

In order to simplify the notation, in the following we will assume, without loss of generality, that the clock starts counting when the first subset of stations is activated. Moreover, we will avoid specifying the modulo ℓ on the columns of the matrix: it is understood that the matrix is scanned circularly.

Definition 3.1 (contention-reducing matrix). *A contention-reducing matrix $\mathcal{M}$ is a transmission matrix randomly constructed as follows. Let $\rho(i,j) = \max\{1, i + [j \mod (2\log\log n)] - \log\log n\}$ for any $1 \leq i \leq \log n$ and $j > 0$. Any entry $M_{i,j}$ is formed by letting $\text{Prob}[x \in M_{i,j}] = \frac{1}{2^{\rho(i,j)}}$ for every station's ID $x \in [n]$. All decisions on whether $x \in M_{i,j}$ are made* independently.

In the following, we assume that every station executing `contention-reduction` is equipped with a contention-reducing matrix.

LEMMA 3.2. *Let j be an arbitrary round. The probability that there exists a station $w \in A(j)$ isolated at round j is at least*

$$\sum_{i=1}^{\log n} |A_{i,j}| \cdot \mathrm{Prob}[x \in M_{i,j}] \left(\frac{1}{4}\right)^{\sum_{i=1}^{\log n} |A_{i,j}| \cdot \mathrm{Prob}[x \in M_{i,j}]} .$$

Proof. Let $E_1(i,j)$ be the event "there exists $w \in A_{i,j}$ such that $A_{i,j} \cap M_{i,j} = \{w\}$", and let $E_2(i,j)$ be the event "for all l with $l \neq i$, $A_{l,j} \cap M_{l,j} = \emptyset$." Clearly $A(t)$ is isolated at time j if and only if the event $\bigcup_{i=1}^{\log n} E_1(i,j) \cap E_2(i,j)$ arises.

Event $E_1(i,j)$ arises if and only if there exists at least $w \in A_{i,j}$ such that $w \in M_{i,j}$ and $y \notin M_{i,j}$ for every $y \in A_{i,j} \setminus \{w\}$. Hence,

$$\begin{aligned}\mathrm{Prob}[E_1(i,j)] &\geq |A_{i,j}| \cdot \mathrm{Prob}[x \in M_{i,j}] \left(1 - \mathrm{Prob}[x \in M_{i,j}]\right)^{|A_{i,j}|-1} \\ &\geq |A_{i,j}| \cdot \mathrm{Prob}[x \in M_{i,j}] \left(1 - \mathrm{Prob}[x \in M_{i,j}]\right)^{|A_{i,j}|} .\end{aligned}$$

On the other hand, event $E_2(i,j)$ corresponds to the condition that for all $l \neq i$, $y \notin M_{l,j}$ for any $y \in A_{l,j}$. Hence,

$$\mathrm{Prob}[E_2(i,j)] = \prod_{l=1, l \neq i}^{\log n} \left(1 - \mathrm{Prob}[x \in M_{l,j}]\right)^{|A_{l,j}|} .$$

Since $E_1(i,j)$ and $E_2(i,j)$ are statistically independent, it follows that

$$\begin{aligned}&\mathrm{Prob}[E_1(i,j) \cap E_2(i,j)] \\ &\geq |A_{i,j}| \cdot \mathrm{Prob}[x \in M_{i,j}] \prod_{i=1}^{\log n} \left(1 - \mathrm{Prob}[x \in M_{i,j}]\right)^{|A_{i,j}|} \\ &= |A_{i,j}| \cdot \mathrm{Prob}[x \in M_{i,j}] \prod_{i=1}^{\log n} \left(1 - \mathrm{Prob}[x \in M_{i,j}]\right)^{\frac{|A_{i,j}| \cdot \mathrm{Prob}[x \in M_{i,j}]}{\mathrm{Prob}[x \in M_{i,j}]}} \\ &\geq |A_{i,j}| \cdot \mathrm{Prob}[x \in M_{i,j}] \cdot 4^{-\sum_{i=1}^{\log n} |A_{i,j}| \cdot \mathrm{Prob}[x \in M_{i,j}]},\end{aligned}$$

where in the last inequality we have used the fact that for $1 \leq i \leq \log n$,

$$\left(1 - \mathrm{Prob}[x \in M_{i,j}]\right)^{\frac{1}{\mathrm{Prob}[x \in M_{i,j}]}} \geq \frac{1}{4} .$$

By observing that for any fixed j and $1 \leq i \leq \log n$ the events $E_1(i,j) \cap E_2(i,j)$ are mutually exclusive, the result follows. □

In the rest of this subsection, our goal will be to show that there are sufficiently many rounds in which the probability of isolating a station is constant (cf. Lemma 3.8). In view of Lemma 3.2, in order to estimate such a probability it will suffice to give upper and lower bounds to the sum $\sum_{i=1}^{\log n} |A_{i,j}| \cdot \mathrm{Prob}[x \in M_{i,j}]$ (cf. Lemma 3.7). First we need to introduce some additional notation.

We can partition the global time in consecutive segments $T_0, T_1, \ldots$, where each *time segment* is a time interval that starts with a round j such that $j \mod (2 \log\log n) = 0$ and ends with a round j such that $j \mod (2 \log\log n) = 2\log\log n - 1$. Namely, we have the following definition.

DEFINITION 3.3 (time segment). *For $r \geq 0$, the rth time segment is the following interval of $2 \log\log n$ consecutive rounds: $T = [2r \log\log n, 2r \log\log n + 1, \ldots, (2r + 2) \log\log n - 1]$.*

Given a time segment $T = [2r \log\log n, 2r \log\log n + 1, \ldots, (2r + 2) \log\log n - 1]$, for some $r \geq 0$, we let $A_i(T) = A_{i,2r \log\log n}$, for $1 \leq i \leq \log n$. That is, $A_i(T)$ is the set of active stations running on row i at the first round of segment T.

Definition 3.4 (saturated time segment). *A saturated time segment T is a time segment such that the following conditions hold:*

(a) $|A_{\log k}(T)| > \frac{k}{\log n}$*;*

(b) *for* $i > \log k$*,* $|A_i(T)| \leq \frac{2k}{\log n}$.

Definition 3.5 (saturated time interval). *A time interval $[t_1, t_2]$ is said to be saturated if it includes at least $\eta = c \cdot (k/2) \log n$ saturated time segments $T_0, T_1, \ldots, T_{\eta-1}$.*

Given a time segment $T_r = [2r \log\log n, 2r \log\log n + 1, \ldots, (2r + 2) \log\log n - 1]$, for some $r \geq 0$, we let $A_i(T_r) = A_{i,2r \log\log n}$, for $1 \leq i \leq \log n$. That is, $A_i(T_r)$ denotes the set of active stations running on row i at the first round of T_r.

Following the algorithm, we can observe that any station starts transmitting on row i, for $i = 1, 2, \ldots, \log n$, at time $\tau_i \equiv 0 \mod (2 \log\log n)$. Therefore, an active station can join a row only at the beginning of a time segment (i.e., on rounds j such that $j \mod (2 \log\log n) = 0$).

As a result, the number of active stations in each row of the matrix cannot increase during a time segment. It could decrease as some active station could successfully transmit (and consequently switch off). A time segment T_r such that there is at least one station that transmits successfully during T_r, will be called a *successful time segment*; otherwise it will be called *unsuccessful*. Clearly, within any unsuccessful time segment the number of active stations remains the same for all rounds of the segment. Namely, for any fixed unsuccessful time segment T_r, for every round $j \in T_r$, we have

$$|A_{i,j}| = |A_i(T_r)|, \tag{3.1}$$

for every $1 \leq i \leq \log n$.

Now we are ready to give upper and lower bounds to the sum $\sum_{i=1}^{\log n} |A_{i,j}| \cdot \text{Prob}[x \in M_{i,j}]$. Namely, the following two lemmas aim at showing that a saturated interval contains sufficiently many rounds in which the above sum is constant.

Lemma 3.6. *Let $[t_1, t_2]$ be a saturated interval. There are at least $\delta = c \cdot (k/8) \log n$ unsuccessful time segments $T_0, T_1, \ldots, T_{\delta-1}$ such that for every T_m, $0 \leq m \leq \delta - 1$,*

$$\frac{1}{\log n} \leq \sum_{i=1}^{\log n} \frac{|A_i(T_m)|}{2^i} \leq 8 \cdot \log n. \tag{3.2}$$

Proof. Let $\mathcal{F}$ be the family of saturated time segments included in $[t_1, t_2]$. By Definition 3.5, we know that $|\mathcal{F}| = \eta = c \cdot (k/2) \log n$. Let $\mathcal{F}' \subseteq \mathcal{F}$ be the subset of unsuccessful segments. Since there are at most k successful time segments, we must have

$$|\mathcal{F}'| \geq |\mathcal{F}| - k = \eta - k > \eta - \delta = 2\delta, \tag{3.3}$$

where the last inequality follows for n sufficiently large (namely, $n > 2^{8/c}$). In view of (3.3), our aim is to show that at least half the elements in $\mathcal{F}'$ satisfy condition (3.2).

By condition (a) of Definition 3.4 we must have that $|A_{\log k}(T_m)| \geq k/ \log n$ for every time segment $T_m \in \mathcal{F}'$. Hence, for every $T_m \in \mathcal{F}'$, it holds that

$$\sum_{i=1}^{\log n} \frac{|A_i(T_m)|}{2^i} \geq \frac{|A_{\log k}(T_m)|}{k} \geq \frac{1}{\log n}.$$

It remains to prove the upper bound of (3.2).

Assume by contradiction that $\mathcal{F}'$ contains less than $\delta = c \cdot (k/8) \log n$ segments that satisfy the rightmost inequality of (3.2). So, for at least $|\mathcal{F}'| - c \cdot (k/8) \log n \geq c \cdot (k/8) \log n$ saturated segments T_m, we must have

$$\sum_{i=1}^{\log n} \frac{|A_i(T_m)|}{2^i} > 8 \cdot \log n. \tag{3.4}$$

Let W be the set of all rounds included in time segments T_m that satisfies (3.4). We know by (3.1) that $|A_{i,j}| = |A_i(T_m)|$ for every $j \in T_m$. Therefore, for every round $j \in T_m$, we have

$$\sum_{i=1}^{\log n} \frac{|A_{i,j}|}{2^i} = \sum_{i=1}^{\log n} \frac{|A_i(T_m)|}{2^i} > 8 \cdot \log n. \tag{3.5}$$

Since there are $2 \log \log n$ rounds in each time segment, the contradiction hypothesis implies that

$$|W| > c \cdot (k/8) \log n (2 \log \log n) = c \cdot (k/4) \log n \log \log n. \tag{3.6}$$

Since the total number of stations that can be active at any round is upper bounded by k, for every round $j \in [t_1, t_2]$ we must have

$$\sum_{i=1}^{\log n} |A_{i,j}| \leq k. \tag{3.7}$$

For any $j \in [t_1, t_2]$, let $U(j) = \bigcup_{i=1}^{\log n} A_{i,j}$. Following the algorithm, any station lies on row i, $1 \leq i \leq \log n$, for ℓ_i rounds. Therefore, for every row $1 \leq i \leq \log n$

$$\ell_i \max_{t_1 \leq j \leq t_2} |U(j)| \geq \sum_{j=t_1}^{t_2} |A_{i,j}| \geq \sum_{j \in W} |A_{i,j}|.$$

Hence, we have

$$\begin{aligned}
\sum_{i=1}^{\log n} \max_{t_1 \leq j \leq t_2} |U(j)| &\geq \sum_{i=1}^{\log n} \sum_{j \in W} \frac{|A_{i,j}|}{\ell_i} \\
&= \sum_{j \in W} \sum_{i=1}^{\log n} \frac{|A_{i,j}|}{\ell_i} \\
&= \frac{1}{2c \log n \log \log n} \sum_{j \in W} \sum_{i=1}^{\log n} \frac{|A_{i,j}|}{2^i} \quad \text{(substituting the value of } \ell_i\text{)} \\
&> \frac{1}{2c \log n \log \log n} \sum_{j \in W} 8 \cdot \log n \quad \text{(by (3.5))} \\
&\geq \frac{|W| \cdot 4}{c \log \log n}.
\end{aligned}$$

Therefore,

$$\max_{t_1 \leq j \leq t_2} |U(j)| > \frac{|W| \cdot 4}{c \log n \log \log n}.$$

And by using (3.6) we get

$$\max_{t_1\le j\le t_2} |U(j)| > \frac{c(k/4)\log n\log\log n\cdot 4}{c\log n\log\log n} = k.$$

This implies that there exists $j'\in[t_1,t_2]$ such that

$$\left|\bigcup_{i=1}^{\log n} A_{i,j'}\right| > k,$$

which contradicts (3.7). □

LEMMA 3.7. *Let $[t_1,t_2]$ be a saturated interval. At least $\delta = c\cdot(k/8)\log n\log\log n$ rounds $j\in[t_1,t_2]$ are such that*

$$\frac{1}{2} \le \sum_{i=1}^{\log n}\frac{|A_{i,j}|}{2^{\rho(i,j)}} \le 8.$$

Proof. By Lemma 3.6, there are at least $\delta = c\cdot(k/8)\log n$ unsuccessful time segments $T_0, T_1, \ldots, T_{\delta-1}$ such that for every T_m, $0\le m\le\delta-1$,

$$\frac{1}{\log n} \le \sum_{i=1}^{\log n}\frac{|A_i(T_m)|}{2^i} \le 8\cdot\log n. \tag{3.8}$$

Hence, by (3.1), we have that for every round $j\in T_m$, with $0\le m\le\delta-1$,

$$\frac{1}{\log n} \le \sum_{i=1}^{\log n}\frac{|A_{i,j}|}{2^i} \le 8\cdot\log n,$$

with $|A_{i,j}|$ that does not change for every i, $1\le i\le\log n$, and $j\in T_m$. Recalling that $\rho(i,j) = \max\{1, i+[j \mod (2\log\log n)] - \log\log n\}$, we want to show that for any time segment T_m, for $0\le m\le\delta-1$, there exists a $j\in T_m$ such that

$$\frac{1}{2} \le \sum_{i=1}^{\log n}\frac{|A_{i,j}|}{2^{\rho(i,j)}} \le 8. \tag{3.9}$$

Fix $x = \sum_{i=1}^{\log n}\frac{|A_{i,j}|}{2^i}$ and $\xi(j) = \sum_{i=1}^{\log n}\frac{|A_{i,j}|}{2^{\rho(i,j)}}$. Observe that $\xi(j) = x$ for $j = \log\log n$. We distinguish two cases.

Suppose first that $1/2\le x\le 8\cdot\log n$. Since x can be as large as $8\cdot\log n$, our aim is to increase the value of j, starting from $j=\log\log n$, until (3.9) is satisfied. In fact, for every fixed i there is a one-to-one correspondence between $\log\log n < j \le 2\log\log n$ and $1\le\rho(i,j)\le i+\log\log n$. Observe that for $\log\log n < j\le 2\log\log n$ the value of $\rho(i,j)$ is equal to $i+[j \mod (2\log\log n)] - \log\log n$, and thus increasing j by 1 within the considered range decreases $\xi(j)$ by factor 2, i.e., $\xi(j) = 2\xi(j+1)$. It also follows that $\xi(2\log\log n) = x/2^{\log\log n} = x/\log n$. Therefore, there must exist j, with $\log\log n < j\le 2\log\log n$, such that $\xi(j)$ falls within the bounds of inequality (3.9).

Suppose now that $\frac{1}{\log n}\le x < 1/2$. Since for $j=\log\log n$, $\xi(j) = x < 1/2$, our aim is now to *decrease* the value of j, starting from $j=\log\log n$, until $\xi(j)$ rises to within the bounds of inequality (3.9). In order to do so, we will show that there exists $1\le t<\log\log n$ such that $1/2\le\xi(\log\log n - t)\le 8$. By the definition of

$\rho(i,j)$ it follows that $\rho(i, \log\log n - t) = 1$ for $i = 1, 2, \ldots, t+1$. Hence, for any fixed $1 \le t < \log\log n$,

$$\xi(\log\log n - t) = \sum_{i=1}^{t+1} \frac{|A_{i,\log\log n - t}|}{2} + \sum_{i=t+2}^{\log n} \frac{|A_{i,\log\log n-t}|}{2^{i-t}} \tag{3.10}$$

$$\le x + 2x + \cdots + 2^t x + 2^t x < 2^{t+2} x, \tag{3.11}$$

where in the second-, last inequality we have used the fact that for $i = 1, 2, \ldots, t+1$

$$\frac{|A_{i,\log\log n-t}|}{2} = 2^{i-1} \cdot \frac{|A_{i,\log\log n-t}|}{2^i} \le 2^{i-1} x.$$

Now we want to bound $\xi(\log\log n - t)$ from below for $1 \le t < \log\log n$. Consider equation (3.10) and let $S_1(t) = \sum_{i=1}^{t+1} \frac{|A_{i,\log\log n-t}|}{2}$ and $S_2(t) = \sum_{i=t+2}^{\log n} \frac{|A_{i,\log\log n-t}|}{2^{i-t}}$.

Suppose first that $S_1(1) \neq 0$. In such a case at least one summand in $S_1(1)$ is larger than zero. Consequently, $\xi(\log\log n - t) \ge 1/2$, which satisfies the lower bound in inequality (3.9). Moreover, for $t = 1$, inequality (3.11) guarantees an upper bound of $8x \le 4$. Therefore, both lower and upper bounds of inequality (3.9) are satisfied.

Suppose now that $S_1(1) = 0$. Let t^* be the largest integer $1 \le t^* < \log\log n$ such that $S_1(t) = 0$ for $t = 1, 2, \ldots, t^*$. Then, for $1 \le t \le t^*$, we must have

$$\xi(\log\log n - t) = S_2(t) = 2^t x. \tag{3.12}$$

There are two cases to analyze. Suppose first that

$$\text{for all } t \le t^*, \xi(\log\log n - t) = 2^t x < 1/2. \tag{3.13}$$

This implies that $t^* < \log\log n - 1$ (indeed for $t^* = \log\log n - 1$ we would have $\xi(\log\log n - t^*) \ge 1/2$, since $x \ge 1/\log n$). Hence, by definition of t^*, $S_1(t^*+1)$ will have at least one nonzero summant, which implies $\xi(\log\log n - (t^*+1)) \ge S_1(t^*+1) \ge 1/2$. By (3.13) it follows that $x < 1/(2^{t^*+1})$. Plugging this into (3.11), we get that $\xi(\log\log n - (t^*+1)) < 2^{t^*+3} x < 4$. Therefore, there exists a $1 \le t < \log\log n$, mainly $t = t^* + 1$, such that $1/2 \le \xi(\log\log n - t) < 4$.

Suppose now that (3.13) does not hold and let t' be the smallest $t \le t^*$ such that $\xi(\log\log n - t) = 2^t x \ge 1/2$. It follows that for $t' - 1 \ge 0$, we have $2^{t'-1} x < 1/2$. Applying this to the upper bound on $\xi(\log\log n - t')$ in inequality (11), we obtain

$$2^{t'+2} x = 2^3 \cdot 2^{t'-1} x < 2^3 \cdot \frac{1}{2} = 4.$$

Therefore, there exists a $0 \le t \le \log\log n - 1$, mainly $t = t' - 1$, such that $1/2 \le \xi(\log\log n - t) < 4$. In both cases $\xi(\log\log n - t)$ is within the bounds defined in inequality (3.9) and the proof is now completed. □

Using Lemmas 3.7 and 3.2 we can finally estimate the number of rounds in a saturated interval such that the probability of isolating a station is a constant.

LEMMA 3.8. *Let* $[t_1, t_2]$ *be a saturated interval. There are at least* $\delta = c \cdot (k/8) \log n$ *rounds* $j \in [t_1, t_2]$ *such that the probability that there exists a station* $w \in A(j)$ *isolated at time* j *is at least* $(1/2)^{17}$.

Proof. Combining Lemmas 3.7 and 3.2, we have that there are at least $c \cdot (k/8) \log n$ rounds $j \in [t_1, t_2]$ such that the probability of isolating a station at round j is at least

$$\sum_{i=1}^{\log n} \frac{|A_{i,j}|}{2^{i+\rho(j)}} \left(\frac{1}{4}\right)^{\sum_{i=1}^{\log n} \frac{|A_{i,j}|}{2^{i+\rho(j)}}} \ge \frac{1}{2} \cdot \left(\frac{1}{4}\right)^8 = (1/2)^{17}. \qquad □$$

3.3. Properties of algorithm `iterative-selection`. Assume that c is sufficiently large to guarantee that $(c/32)\cdot(2^i + 2^i\log(n/2^i)) > m_i$, for any integer $0 \le i \le \log n$, where $m_i = \Theta(2^i + 2^i\log(n/2^i))$ is the length of algorithm $(n, 2^i)$-KG used in the thread i of algorithm `iterative-selection`. Recall that $\Gamma_k = \sum_{i\le\log k}\ell_i = \Theta(k\log n\log\log n)$ and that an active station is called "participating in thread i of algorithm `iterative-selection`" if it is actively running algorithm $(n, 2^i)$-KG, which starts at the first round t' such that $t' = 0 \mod m_i\log n$ and t' is at least $\Gamma_{2^i\log n}$ rounds after awakening. To avoid rounding, assume that $k/\log n$ is a power of 2.

It follows from the specification of algorithm `iterative-selection` that it can isolate k stations arriving in time interval of length $O((k + k\log(n/k))\log n)$ in $O((k + k\log(n/k))\log n)$ rounds, simply by applying the selection property of algorithm $(n, 2^{\log k})$-KG run in thread $\log k$. The $O((k + k\log(n/k))\log n)$ bound comes from the length of algorithm $(n, 2^{\log k})$-KG and an additional logarithmic factor deriving from the number of threads in algorithm `iterative-selection`. By a simple inductive argument applied to consecutive sufficiently "saturated" time intervals, we could generalize this result to obtain maximum latency $O((k + k\log(n/k))\log n)$ for k being the total number of awakened stations. However, if some other mechanism ensures that we could decrease the number of stations participating in threads by a logarithmic factor, algorithm `iterative-selection` would guarantee smaller latency.

LEMMA 3.9. *Consider an execution of algorithm* `iterative-selection`*. Assume that in each interval of length* $(c/8)k\log n\log\log n$*, no more than* $k/\log n$ *stations start participating in thread* $\log(k/\log n) + 1$ *for some* $k \le n$*. Then the following three conditions hold:*

(a) *at each time of this execution there are no more than* $2k/\log n$ *active stations participating in thread* $\log(k/\log n) + 1$*;*

(b) *no station participates in a thread bigger than* $\log(k/\log n) + 1$*;*

(c) *each station has latency* $O(k\log n\log\log n)$*.*

Proof. First we prove part (a). Let $i = \log(k/\log n)$. Suppose, to the contrary, that part (a) of the lemma is not true, and let t be the first time when the number of active stations participating in thread $i+1 = \log(k/\log n)+1$ is bigger than $2k/\log n$. It implies that the number of active stations participating in thread $i+1$ in any round smaller than t is at most $2k/\log n$. Let $\tau = t - (c/8)k\log n\log\log n + 1$.

By the assumptions on the execution, at most $k/\log n$ stations have started participating in thread $i + 1$ in the period $[\tau, t]$. Hence, there are more than $k/\log n$ stations active and participating in thread $i + 1$ at round τ; otherwise the contradictory assumption on the number of active stations participating in thread $i + 1$ at round t would not hold.

On the other hand, we argue that all stations active at round τ and participating in thread $i+1$ become isolated by round t. To see this, consider the first round $\tau' \ge \tau$ such that $\tau' = 0 \mod m_{i+1}\log n$, i.e., the first round not smaller than τ in which algorithm $(n, 2^{i+1})$-KG restarts in thread $i + 1$ of algorithm `iterative-selection`. We have

$$m_{i+1} < (c/32)\cdot(2^{i+1} + 2^{i+1}\log(n/2^{i+1})) \le (c/8)\cdot\frac{k}{\log n}\log\frac{n\log n}{2k},$$

and thus

$$m_{i+1}\log n < (c/8)\cdot k\log\frac{n\log n}{2k} < (1/2)\cdot(c/8)k\log n\log\log n = (1/2)\cdot(t-\tau+1)\ ,$$

where the second inequality holds for sufficiently large n. We get that $\tau' \leq t - m_{i+1}\log n$ and consequently $\tau' + m_{i+1}\log n \leq t$. This means that algorithm $(n, 2^{i+1})$-KG (run in thread $i+1$ of algorithm `iterative-selection`) restarted at time τ' finishes by time $\tau' + m_{i+1}\log n \leq t$. Note that

- there were less than $2^{i+1} = \frac{2k}{\log n}$ active stations participating in thread $i+1$ at round τ' (round t defined as the first round with more than $\frac{2k}{\log n}$ active stations participating in thread $i+1$ and the fact that $\tau' < t$), and
- only these stations can participate in the restarted run of algorithm $(n, 2^{i+1})$-KG (by the specification of algorithm `iterative-selection`), and
- all these stations become isolated during the execution of this run of algorithm $(n, 2^{i+1})$-KG, because there are no more than $\frac{2k}{\log n} = 2^{i+1}$ of them (this is guaranteed by the property of algorithm $(n, 2^{i+1})$-KG).

Hence, all stations active and participating in thread $i+1$ at round τ' are isolated before t and so also those active and participating in thread $i+1$ at round $\tau \leq \tau'$: as each station active and participating in thread $i+1$ at round τ is either isolated before τ' or active and participating in thread $i+1$ at τ'—in both cases it becomes isolated by round t.

To summarize, in the period $[\tau, t]$ at most $k/\log n$ stations have started participating in thread $i+1$ and more than $k/\log n$ stations (all active and participating in thread $i+1$ at round τ, and perhaps some more) have been isolated. These two facts, combined with the property that at round τ there were at most $2k/\log n$ stations that were active and participated in thread $i+1$ (as $\tau < t$), imply that at round t there are less active stations participating in thread $i+1$ than at round τ, i.e., less than $2k/\log n$. This is a contradiction (with the choice of t), which completes the proof of part (a).

The proof of part (b) follows directly from part (a) and from the property of algorithm $(n, 2^{i+1})$-KG run in thread $i+1$: since at each starting point of this protocol there are at most $2k/\log n$ stations that are active and participating in thread $i+1$ (by part (a)), algorithm $(n, 2^{i+1})$-KG will isolate them all. The additional observation that needs to be done to support the above argument is that after starting participation in thread $i+1$, it takes more than $2m_{i+1}\log n$ rounds to a station to start participating in thread $i+2$ (because the difference between $\Gamma_{2^{i+1}\log n}$ and $\Gamma_{2^{i+2}\log n}$ is at least $\ell_{\log(2^{i+1}\log n)} \geq \ell_{k+1}$ that is bigger than $2m_{i+1}\log n$), which means that the station has at least one chance to participate in a full run of algorithm $(n, 2^{i+1})$-KG before starting participation in the next thread $i+2$, and thus during this run it will be successfully isolated.

In order to prove part (c), observe that a station starts participating in thread $i+1$ after at most $\Gamma_k + m_{i+1}\log n$ rounds from its wake-up. After joining, it becomes isolated in this run, which is within another at most $m_{i+1}\log n$ rounds. This is because together with it at most $(2k/\log n) - 1$ other stations could start this run of algorithm $(n, 2^{i+1})$-KG, by (already proved) parts (a) and (b) of the lemma, and thus algorithm $(n, 2^{i+1})$-KG guarantees isolation of all participating at most $2^{i+1} = 2k/\log n$ stations (by the property of the algorithm). Finally, recall that $\Gamma_k + 2m_{i+1}\log n = \Theta(2^{i+1}\log(n/2^{i+1})\log n) \leq O(k\log n\log\log n)$, since $2^{i+1} = 2k/\log n$ by definition of i. On the other hand, if a station does not participate in thread $i+1$, it means that it has already been isolated before time $\Gamma_k + m_{i+1}\log n = O(k\log n\log\log n)$. This completes the proof of part (c). □

3.4. Analysis of algorithm `conflict-resolution`.

In the following, the random contention-reducing matrix defined in Definition 3.1 will be referred to simply as the *random matrix*. Our aim is to show the existence of a *deterministic matrix* that guarantees a fast execution of algorithm `conflict-resolution` (Theorem 3.12). We first need two preparatory results. Let c^* be a constant such that $c/8 < c^* < c/4$.

LEMMA 3.10. *The random matrix guarantees the following property. Fix the following parameters: the total number k of awakened stations, a wake-up pattern σ for some k stations, and the sets $A(j)$ of active stations in rounds j. Then in algorithm* `contention-reduction`*, for every time interval of length $2c^*k\log n\log\log n$ at most $k/\log n$ stations switch from row $\log k$ to row $\log k+1$ of the matrix during the interval, with probability at least $1-\exp(-(c/2^{23})k\log n)$.*

Proof. Let us fix the following parameters:

- the number of awaken stations $k \le n$;
- a wake-up schedule σ of some k stations; without loss of generality assume that the first station is awakened in time interval $[1, 2\log\log n]$ (since the definition of the matrix applies operation modulo $2\log\log n$ whenever refers to global time);
- sets $A(j)$ of active stations in rounds j;
- a time interval $[\tau_1,\tau_2]$ of length $2c^*k\log n\log\log n$ contained in $[\Gamma_k, k\cdot(\Gamma_n + 2m_{\log n}\log n)]$.

Note that $\Gamma_k = \ell_1 + \cdots + \ell_{\log k}$ is the earliest possible time when a station can switch from its row $\log k$ to row $\log k+1$ of the matrix, while $k\cdot(\Gamma_n + 2m_{\log n}\log n)$ is an upper bound on the time period in which there is always some of k stations actively running the algorithm (if there is a round between when there are no active stations, the two sides of this round could be analyzed independently). Indeed, the latter argument is based on the property of the $(n, 2^{\log n})$-KG procedure guaranteeing isolation of all participating stations in thread $\log n$ of algorithm `iterative-selection` in $m_{\log n}$ rounds, while it also has to be taken into account that starting participation in this thread may take up to $\Gamma_n + m_{\log n}$ rounds.

In the first part of the proof of the lemma we will calculate the probability of the following event:

> (*) $[\tau_1,\tau_2]$ is the first time interval of length $2c^*k\log n\log\log n$ during which at most $k/\log n$ stations switch from row $\log k$ to $\log k+1$ of the matrix.

Let us consider time interval $[\tau_1 - 2(c-c^*)k\log n\log\log n, \tau_1)$, and set up $t_1 = \tau_1 - 2(c-c^*)k\log n\log\log n$ and $t_2 = \tau_1 - 1$. We prove the following claim.

CLAIM 1. *With probability* 1*, if the condition* (*) *is satisfied, then $[t_1,t_2]$ is a saturated interval.*

Proof. Indeed, each station that switches from row $\log k$ to row $\log k+1$ in the interval $[\tau_1,\tau_2]$ has been in row $\log k$ for $\ell_{\log k}$ preceding rounds, that is, it entered row $\log k$ during the interval $[\tau_1-\ell_{\log k}, \tau_2-\ell_{\log k}]$. Since $\tau_2 - \ell_{\log k} \le \tau_1 - 2(c-c^*)k\log n\log\log n - 1 = t_1 - 1$, there are more than $k/\log n$ stations that are in row $\log k$ of the matrix during the whole interval $[t_1,t_2]$, for instance, all stations that switch to row $\log k+1$ during time interval $[\tau_1,\tau_2]$, the number of which is bigger than $k/\log n$ by the assumption about time interval $[\tau_1,\tau_2]$. This implies that in all these rounds condition (a) of Definition 3.4 of a saturated segment is satisfied. Next, observe that by round τ_2 the assumptions of Lemma 3.9 hold—by the choice of $[\tau_1,\tau_2]$, which is the first interval of length $2c^*k\log n\log\log n > (c/2)k\log n\log\log n$ violating these assumptions, and the specification of the algorithm `iterative-selection` saying that a station starts participating in thread $i + 1 = \log(k/\log n) + 1$ at round at most $\Gamma_k + m_{i+1}\log n$, i.e., at most $m_{i+1}\log n < (c/16) \cdot 2k\log n\log\log n$ rounds after the station switches to row $\log k + 1$. Hence, by parts (a) and (b) of Lemma 3.9, the parallel run of protocol `iterative-selection` guarantees that the number of active stations that in parallel execution of the matrix are below row $\log k$ is at most $2k/\log n$ in any round $t \le \tau_2$. This implies that in all these rounds condition (b) of Definition 3.4 of a saturated segment is fulfilled. Finally, observe that the length of $[t_1, t_2]$ is $2(c - c^*)k\log n\log\log n \ge (3c/4)k\log n \cdot (2\log\log n)$, which means that the interval contains at least $(c/4)k\log n$ consecutive saturated time segments, and thus is saturated itself; cf. Definitions 3.4 and 3.5. This concludes the proof of Claim 1.

To conclude the first part of the proof it is then sufficient to prove that $[t_1, t_2]$ is saturated with small probability. If $[t_1, t_2]$ is saturated, then, by Lemma 3.8, there are at least $(c/8)k\log n$ rounds in which the probability of isolating a station is at least $1/2^{17}$. Since the random choices made in different rounds are independent, by applying the Chernoff bound to such rounds we obtain that the total number of isolated stations in these rounds is valid, in particular it is at most k (as no more than all k stations could be isolated in the whole execution), with probability at most $\exp(-(c/2^{22})k\log n)$. Indeed, the expected number of such rounds is at least $(c/2^{20})k\log n$, and deviating from this number even by a sufficiently small constant factor occurs with probability at most $\exp(-(c/2^{22})k\log n)$. Hence, by applying Claim 1, we get that (*) holds with probability at most $\exp(-(c/2^{22})k\log n)$.

In the second part of the proof we conclude the lemma by observing that there are at most $k \cdot (\Gamma_n + 2m_{\log n}\log n)$ different intervals $[\tau_1, \tau_2]$ that could be fixed in the first part of the proof. Therefore, after applying the union bound over all such possibilities, the sought probability of violating the statement of the lemma for *some* interval $[\tau_1, \tau_2]$ of length $2c^*k\log n\log\log n$ contained in $[\Gamma_k, k \cdot (\Gamma_n + 2m_{\log n}\log n)]$ is at most $\exp(-(c/2^{22})k\log n) \cdot k \cdot (\Gamma_n + 2m_{\log n}\log n) \le \exp(-(c/2^{23})k\log n)$ for sufficiently large constant $c > 0$. ◻

In the next result, we essentially generalize the statement of Lemma 3.10 to any number of at most k awakened stations and any possible wake-up schedule and sets of active stations within an interval of at most $k \cdot (\Gamma_n + 2m_{\log n}\log n)$ rounds.

LEMMA 3.11. *The random matrix guarantees the following property. For any $1 \le k \le n$, if at most k stations awaken, then the algorithm* `contention-reduction` *guarantees that for every time interval of length $2c^*k\log n\log\log n$ at most $k/\log n$ stations switch from row $\log k$ to row $\log k + 1$ of the matrix during the interval, with probability at least $1 - \exp(-\Omega(k\log n))$.*

Proof. We remove the assumptions about fixed parameters from the statement of Lemma 3.10: parameter k, wake-up schedule of some k stations, and sets of active stations $A(j)$ in rounds j, all within at most $k \cdot (\Gamma_n + 2m_{\log n}\log n)$ rounds. There are at most n possible values of k and at most $\binom{n}{k}(k \cdot (\Gamma_n + 2m_{\log n}\log n))^k$ of wake-up schedules and also rounds of isolation of some k stations (taken out of n) within at most $2k\Gamma_n$ rounds. Here we used the fact that instead of fixing the wake-up pattern and the sets $A(j)$ of active stations in rounds j we could equivalently take the same wake-up pattern and the isolation rounds of k stations corresponding to the sets $A(j)$. Thus, the number of different configurations of these objects is upper bounded by

$$n \cdot \left(\binom{n}{k}(k \cdot (\Gamma_n + 2m_{\log n}\log n))^k\right)^2 \le \left(\frac{ne}{k}\right)^{2k} (n\log^2 n)^{4k} \le 2^{5k\log n}$$

for sufficiently large n. This multiplied by the probability $\exp(-(c/2^{22})k\log n)$ that a given configuration of parameters violates the statement of Lemma 3.10 is still

$\exp(-\Omega(k \log n))$ for sufficiently large constant $c > 0$. This means, by the union bound over all such configurations, that the probability that any configuration of parameters of execution violates the statement of this lemma is $\exp(-\Omega(k \log n))$. □

We are now ready for the final result.

Theorem 3.12. *There exists a deterministic matrix* $\mathcal{M}$ *such that the corresponding (deterministic) algorithm* `conflict-resolution` *that uses* $\mathcal{M}$ *guarantees packet latency* $O(k \log n \log\log n)$ *if at most* k *stations are awaken with packets. This also implies throughput* $\Omega(1/\log n \log\log n)$.

Proof. It is enough to prove the latency formula, as the throughput follows immediately. The existence of a matrix that satisfies the property of Lemma 3.11 follows by a straightforward application of the probabilistic method [2]. Namely, if we consider the probability space of random matrices from Definition 3.1, it follows that a randomly chosen element in this space has the property of Lemma 3.11 with probability at least $1 - \exp(-\Omega(k \log n)) > 0$. Consequently, a matrix with such a property must exist. This property guarantees that in every time interval of $2c^* k \log n \log\log n$ rounds, at most $k/\log n$ stations switch from row $\log k$ to row $\log k + 1$. This switch takes place $\Gamma_k = \Gamma_{(k/\log n)\log n}$ rounds after the awakening of these nodes, which is a necessary condition to participate in thread $\log(k/\log n) + 1$. Hence, by Lemma 3.9(c), under such an assumption every station becomes isolated in $O(k \log n \log\log n)$ rounds during the parallel run of algorithm `iterative-selection` after switching row $\log k + 1$ in the contention-reduction matrix. Hence, its latency is at most $\Gamma_k + O(k \log n \log\log n) = O(k \log n \log\log n)$. Obviously a station that never switches row $\log k{+}1$ in the run of the contention-reduction matrix must have been isolated by the time it could have reached that row, i.e., before $\Gamma_k = O(k \log n \log\log n)$ rounds after its wake-up. This completes the proof. □

4. Open problems.

Surprisingly, there are still a number of important open problems in this area. The first is to close the gaps for latency of deterministic contention-resolution protocols, practically in all the settings (included the case of strongly nonadaptive protocols). The second interesting twist is to deliver an efficient implementation of our protocol or even an entirely new solution, where by "efficient" we understand polynomially computable (of all used transmission sequences) and using small constants. Further, a vibrant question is how to improve latency of nonadaptive *randomized* protocols, in particular, identifying the most restricted settings for which $O(k)$ expected latency is achievable. Considering worst-case scenarios with dynamic packet arrivals is also a challenging topic, recently started in [10] and [5]. Finally, we conjecture that our nonadaptive protocol can also tolerate random channel failures, at the cost of only slightly increasing the time complexity (and thus, decreasing channel utilization) by a constant factor depending on the (fixed) probability of failures.

Acknowledgment. The authors express sincere gratitude to the referees for their competent and careful reading. Their comments helped to improve the presentation of the paper and to correct some issues present in the preliminary version.

REFERENCES

[1] D. Alistarh, S. Gilbert, R. Guerraoui, Z. Milosevic, and C. Newport, *Securing every bit: Authenticated broadcast in radio networks*, in Proceedings of the 22nd Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), 2010, pp. 50–59.

[2] N. Alon and J. Spencer, *The Probabilistic Method*, John Wiley, New York, 1992.

[3] B. Awerbuch, A. Richa, and C. Scheideler, *A jamming-resistant MAC protocol for single-hop wireless networks*, in Proceedings of the 27th ACM Symposium on Principles of Distributed Computing (PODC), 2008, pp. 45–54.

[4] M. A. Bender, M. Farach-Colton, S. He, B. C. Kuszmaul, and C. E. Leiserson, *Adversarial contention resolution for simple channels*, in Proceedings of the 17th Annual ACM Symposium on Parallel Algorithms (SPAA), 2005, pp. 325–332.

[5] M. Bienkowski, T. Jurdzinski, M. Korzeniowski, and D. R. Kowalski, *Distributed online and stochastic queuing on a multiple access channel*, in Proceedings of DISC, 2012, pp. 121–135.

[6] J. Capetanakis, *Tree algorithms for packet broadcast channels*, IEEE Trans. Inform. Theory, 25 (1979), pp. 505–515.

[7] B. Chlebus, *Randomized Communication in Radio Networks*, in Handbook on Randomized Computing, Vol. I, P. M. Pardalos, S. Rajasekaran, J. Reif, and J. D. P. Rolim, eds., Kluwer Academic Publishers, Norwell, MA, 2001, pp. 401–456.

[8] B. S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, and W. Rytter, *Deterministic broadcasting in unknown radio networks*, Distributed Comput., 15 (2002), pp. 27–38.

[9] B. S. Chlebus and D. Kowalski, *Almost Optimal Explicit Selectors*, in FCT 2005, Lecture Notes in Comput. Sci. 3623, M. Likiewicz and R. Reischuk, eds., Springer, Heidelberg, 2005, pp. 270–280.

[10] B. S. Chlebus, D. R. Kowalski, and M. A. Rokicki, *Adversarial queuing on the multiple access channel*, ACM Trans. Algorithms, 8 (2012).

[11] M. Chrobak, L. Gasieniec, and W. Rytter, *Fast broadcasting and gossiping in radio networks*, J. Algorithms, 43 (2002), pp. 177–189.

[12] A. E. F. Clementi, A. Monti, and R. Silvestri, *Distributed broadcast in radio networks of unknown topology*, Theoret. Comput. Sci., 302 (2003), pp. 337–364.

[13] A. Czumaj and W. Rytter, *broadcasting algorithms in radio networks with unknown topology*, in Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FOCS), 2003, pp. 492–501.

[14] A. Dyachkov and V. Rykov, *A survey of superimposed code theory*, Problems Control Inform. Theory, 12 (1983), pp. 229–242.

[15] A. De Bonis, L. Gasieniec, and U. Vaccaro, *Optimal two-stage algorithms for group testing problems*, SIAM J. Comput., 34 (2005), pp. 1253–1270.

[16] G. De Marco, *Distributed broadcast in unknown radio networks*, SIAM J. Comput., 39 (2010), pp. 2162–2175.

[17] A. Fernandez Anta, M. A. Mosteiro, and J. R. Munoz, *Unbounded contention resolution in multiple-access channels*, Algorithmica, 67 (2013), pp. 295–314.

[18] L. A. Goldberg, P. D. MacKenzie, M. Paterson, and A. Srinivasan, *Contention resolution with constant expected delay*, J. ACM, 47 (2000), pp. 1048–1096.

[19] A. G. Greenberg, P. Flajolet, and R. E. Ladner, *Estimating the multiplicities of conflicts to speed their resolution in multiple access channels*, J. ACM, 34 (1987), pp. 289–325.

[20] A. G. Greenberg and R. E. Ladner, *Estimating the multiplicities of conflicts in multiple access*, in Proceedings of the 24th Annual Symposium on Foundations of Computer Science (FOCS), Tucson, AZ, IEEE, New York, 1983, pp. 383–392.

[21] A. G. Greenberg and S. Winograd, *A lower bound on the time needed in the worst case to resolve conflicts deterministically in multiple access channels*, J. ACM, 32 (1985), pp. 589–596.

[22] S. Gilbert, R. Guerraoui, and C. C. Newport, *Of malicious motes and suspicious sensors: On the efficiency of malicious interference in wireless networks*, Theoret. Comput. Sci., 410 (2009), pp. 546–569.

[23] S. Gilbert, V. King, S. Pettie, E. Porat, J. Saia, and M. Young, *(Near) Optimal resource-competitive broadcast with jamming*, in Proceedings of the 26th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), 2014, pp. 257–266.

[24] J. F. Hayes, *An adaptive technique for local distribution*, IEEE Trans. Commun., 26 (1978), pp. 1178–1186.

[25] W. H. Kautz and R. C. Singleton, *Nonrandom binary superimposed codes*, IEEE Trans. Inform. Theory, 10 (1964), pp. 363–377.

[26] P. Kumar and L. Merakos, *Distributed control of broadcast channels with akcnowledgement feedback: Stability and performance*, in Proceedings of CDC, 1984.

[27] P. Indyk, *Explicit constructions of selectors and related combinatorial structures, with applications*, in Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2002, pp. 697–704.

[28] J. Komlós and A. G. Greenberg, *An asymptotically optimal nonadaptive algorithm for conflict resolution in multiple-access channels*, IEEE Trans. Inform. Theory, 31(1985), pp. 302–306.

[29] D. Kowalski, *On selection problem in radio networks*, in Proceedings of the 24th ACM Symposium on principles of distributed computing (PODC), 2005, pp. 158–166.

[30] R. M. Metcalfe and D. R. Boggs, *Ethernet: Distributed packet switching for local computer networks*, Comm. ACM, 19 (1976), pp. 395–404.

[31] P. Raghavan and E. Upfal, *Stochastic contention resolution with short delays*, SIAM J. Comput., 28 (1999), pp. 709–719.

[32] A. Richa and C. Scheideler, *Jamming-Resistant MAC Protocols for Wireless Networks*, in Encyclopedia of Algorithms, Springer, New York, 2014, pp. 1–5.

[33] B. S. Tsybakov and V. A. Mikhailov, *Free synchronous packet access in a broadcast channel with feedback*, Probl. Inf. Transm., 14 (1978), pp. 259–280.