Noname manuscript No.
(will be inserted by the editor)

Unbounded Contention Resolution in Multiple-Access Channels

Antonio Fernández Anta ·
Miguel A. Mosteiro ·
Jorge Ramón Muñoz

Received: date / Accepted: date

Abstract A frequent problem in settings where a unique resource must be shared among users is how to resolve the contention that arises when all of them must use it, but the resource allows only for one user each time. The application of efficient solutions for this problem spans a myriad of settings such as radio communication networks or databases. For the case where the number of users is unknown, recent work has yielded fruitful results for local area networks and radio networks, although either a (possibly loose) upper bound on the number of users needs to be known [9], or the solution is sub-optimal [4], or it is only implicit [14] or embedded [8] in other problems, with bounds proved only asymptotically. In this paper, under the assumption that collision detection or information on the number of contenders is not available, we present a novel protocol for contention resolution in radio networks, and we recreate a protocol previously used for other problems [8, 14], tailoring the constants for our needs. In contrast with previous work, both protocols are proved to be optimal up to a small constant factor and with high probability

A preliminary version of this work has appeared in [10]. This work was supported in part by the National Science Foundation (CCF-0937829, CCF-1114930), Comunidad de Madrid grant S2009TIC-1692, MINECO grant TEC2011-29688-C02-01, and National Natural Science Foundation of China grant 61020106002.

Antonio Fernández Anta
Institute IMDEA Networks, Madrid, Spain
E-mail: antonio.fernandez@imdea.org

Miguel A. Mosteiro
Department of Computer Science, Kean University, Union, NJ, USA &
LADyR, GSyC, Universidad Rey Juan Carlos, Madrid, Spain
Corresponding author. Phone: +1 908-737-4259, Fax: +1 908-737-4255
E-mail: mmosteir@kean.edu

Jorge Ramón Muñoz
Purdue University, West Lafayette, IN, USA
Work done while being at LADyR, GSyC, Universidad Rey Juan Carlos, Madrid, Spain
E-mail: jramonmu@purdue.edu

for big enough number of contenders. Additionally, the protocols are evaluated and contrasted with the previous work by extensive simulations. The evaluation shows that the complexity bounds obtained by the analysis are rather tight, and that both protocols proposed have small and predictable complexity for many system sizes (unlike previous protocols).

Keywords k-Selection · Radio Networks · Contention Resolution · Communication Protocols · Multiple-access Channel

1 Introduction

The topic of this work is the resolution of contention in settings where an unknown number of users must access a single shared resource, but multiple simultaneous accesses are not feasible. The scope of interest in this problem is wide, ranging from radio and local area networks to databases and transactional memory. (See [4] and the references therein.)

A common theme in protocols used for this problem is the adaptive adjustment of some user variable that reflects its eagerness in trying to access the shared resource. Examples of such variable are the probability of transmitting a message in a radio network or the frequency of packet transmission in a local area network. When such adjustment reduces (resp. increases) the contention, the technique is called *back-off* (resp. *back-on*). Combination of both methods are called *back-on/back-off*. Protocols used may be further characterized by the rate of adjustment. E.g., *exponential back-off*, *polynomial back-on*, etc. In particular, exponential back-off is widely used and it has proven to be efficient in practical applications where statistical arrival of contenders is expected. Nevertheless, worst case arrival patterns, such as bursty or *batched* arrivals, are frequent [15, 22].

A technique called Loglog-iterated Back-off was shown to be within a sublogarithmic factor from optimal with high probability in [4]. [1] The protocol was presented in the context of packet contention resolution in local area networks for batched arrivals. Later on, also for batched arrivals, we presented a back-on/back-off protocol in [9], instantiated in the k-selection problem in Radio Networks (defined in Section 2). The latter protocol, named here Log-fails Adaptive, is asymptotically optimal for any significant probability of error, but additionally requires that some upper bound (possibly loose) on the number of contenders is known. In the present paper, we remove such requirement. In particular, we present and analyze a protocol that we call One-fail Adaptive for k-selection in Radio Networks. We also recreate and analyze another protocol for k-selection, called here Exp Back-on/Back-off , which was previously embedded in protocols for other problems and analyzed only asymptotically [8, 14]. Our analysis shows that One-fail Adaptive and Exp Back-on/Back-off, both of independent interest, resolve contention among

[1] For k contenders, we define *with high probability* to mean with probability at least $1 - 1/k^c$ for some constant $c > 0$.

an unknown and unbounded [2] number of contenders with high probability in optimal time up to constants. Additionally, by means of simulations, we evaluate and contrast the average performance of all four protocols. The simulations show that the complexity bounds obtained in the analysis (with high probability) for these protocols are rather tight for the input sizes considered. Additionally, they show that they are faster that Loglog-iterated Back-off and more predictable for all network sizes than Log-fails Adaptive.

Roadmap The rest of the paper is organized as follows. In the following section the problem, model, related work and results are detailed. In Section 3, we introduce One-fail Adaptive and its analysis. Exp Back-on/Back-off is detailed and analyzed in Section 4. The results of the empirical contrast of all four protocols is given in Section 5 and we finish with concluding remarks and open problems in Section 6.

2 Preliminaries

A well-studied example of unique-resource contention is the problem of broadcasting information in a multiple-access channel. A multiple-access channel is a synchronous system that allows a message to be delivered to many recipients at the same time using a channel of communication but, due to the shared nature of the channel, the simultaneous introduction of messages from multiple sources produce a conflict that precludes any message from being delivered to any recipient. The particular model of multiple-access channel we consider here is the Radio Network, a model of communication network where the channel is contended (even if radio communication is not actually used [6]). We first precise our model of Radio Network as follows.

The Model: We consider a Radio Network comprised of n stations called *nodes*. Each node is assumed to be potentially reachable from any other node in one communication step, hence, the network is characterized as *single-hop* or *one-hop* indistinctively. Before running the protocol, nodes have no information, not even the number of nodes n or their own label. Time is supposed to be slotted in *communication steps*. Assuming that the computation time-cost is negligible in comparison with the communication time-cost, time efficiency is studied in terms of communication steps only. The piece of information assigned to a node in order to deliver it to other nodes is called a *message*. We assume that a message is a data unit that cannot be divided for dissemination or any other purpose. The assignment of a message is due to an external agent and such an event is called a *message arrival*. Communication among nodes is carried out by means of radio broadcast on a shared channel. If exactly one

[2] We use the term unbounded to reflect that not even an upper bound on the number of contenders is known. This should not be confused with the infinitely-many users model where there are countably infinitely many stations. [6]

node transmits at a communication step, such a transmission is called *successful* or *non-colliding*, we say that the message was *delivered*, and all other nodes *receive* such a message. If more than one message is transmitted at the same time, a *collision* occurs, the messages are garbled, and nodes only receive *interference noise*. If no message is transmitted in a communication step, nodes receive only *background noise*. In this work, nodes can not distinguish between interference noise and background noise, thus, the channel is called *without collision detection*. Each node is in one of two states, *active* if it holds a message to deliver, or *idle* otherwise. As in [4, 14, 20], we assume that a node becomes idle upon delivering its message, for instance when an explicit acknowledgement is received (like in the IEEE 802.11 Medium Access Control protocol [2]). For settings where the channel does not provide such functionality, such as Sensor Networks, a hierarchical infrastructure may be predefined to achieve it [8], or a leader can be elected as the node responsible for acknowledging successful transmissions [27].

One of the problems that require contention resolution in Radio Networks is the problem known in the literature as *all-broadcast* [6], or *k-selection* [20]. In k-selection, a set of k out of n network nodes have to access a unique shared channel of communication, each of them at least once. As in [4,14,20], in this paper we study k-selection when all messages arrive simultaneously, or in a *batch*. Under this assumption the k-selection problem is called *static*. A *dynamic* counterpart where messages arrive at different times was also studied [20].

The Problem: Given a Radio Network where k network nodes are activated by a message that arrives simultaneously to all of them, the *static k-selection* problem is solved when each node has delivered its message.

Related Work: A number of fruitful results for contention resolution have been obtained assuming availability of collision detection. Martel presented in [23] a randomized adaptive protocol for k-Selection that works in $O(k+\log n)$ time in expectation[3]. As argued by Kowalski in [20], this protocol can be improved to $O(k + \log\log n)$ in expectation using Willard's expected $\log\log n + O(1)$ selection protocol of [28]. In the same paper, Willard shows that, for any given protocol, there exists a choice of $k \leq n$ such that selection takes at least $\log\log n - O(1)$ expected time for the class of fair selection protocols (i.e., protocols where all nodes use the same probability of transmission to transmit in any given time slot). For the case in which n is not known, in the same paper a $\log\log k + o(\log\log k)$ expected time selection protocol is described, again, making use of collision detection. If collision detection is not available, using the techniques of Kushilevitz and Mansour in [21], it can be shown that, for any given protocol, there exists a choice of $k \leq n$ such that $\Omega(\log n)$ is a lower bound in the expected time to get even the first message delivered. The classic *Decay* protocol [3] resolves conflicts among the transmitting neighbors

[3] Througout this paper, log means $\log_2$ unless otherwise stated.

of a receiver by randomly eliminating half of them in successive rounds. The protocol solves only selection (i.e. one non-coliding transmission) and requires the knowledge of an upper bound on the number of transmitters.

Regarding deterministic solutions, the k-Selection problem was shown to be in $O(k \log(n/k))$ already in the 70's by giving adaptive protocols that make use of collision detection [5, 16, 24]. In all these results the algorithmic technique, known as *tree algorithms*, relies on modeling the protocol as a complete binary tree where the messages are placed at the leaves. Later, Greenberg and Winograd [13] showed a lower bound for that class of protocols of $\Omega(k \log_k n)$. Regarding oblivious algorithms, Komlòs and Greenberg [19] showed the existence of $O(k \log(n/k))$ solutions even without collision detection but requiring knowledge of k and n. More recently, Clementi, Monti, and Silvestri [7] showed a lower bound of $\Omega(k \log(n/k))$, which also holds for adaptive algorithms if collision detection is not available. In [20], Kowalski presented the construction of an oblivious deterministic protocol that, using the explicit selectors of Indyk [17], gives a $O(k \operatorname{polylog} n)$ upper bound without collision detection. More recently, in [1] the authors studied deterministic broadcast in multiple access channels. In their model the data arrivals are adversarial and continuous. The protocols studied that do not require collision detection (RRW, MBTF, and OFRRW) do require knowledge of n.

About related problems, in [12], Gerèb-Graus and Tsantilas presented an algorithm that solves the problem of realizing arbitrary h-relations in an n-node network, with probability at least $1-1/n^c, c > 0$, in $\Theta(h+\log n \log\log n)$ steps. In an h-relation, each processor is the source as well as the destination of at most h messages. Making $h = k$ this protocol can be used to solve static k-selection. However, it requires that nodes know h. Extending previous work on tree algorithms, Greenberg and Leiserson [14] presented randomized routing strategies in fat-trees for bounded number of messages. Underlying their algorithm lies a sawtooth technique used to "guess" the appropriate value for some critical parameter (load factor), that can be used to "guess" the number of contenders in static k-selection. Furthermore, modulo choosing the appropriate constants, Exp Back-on/Back-off uses the same sawtooth technique. Their algorithm uses n and it is analyzed only asymptotically.

Another well-studied problem in Radio Networks is that of *wake-up* [11,18], where nodes are activated either spontaneously or by receiving a transmission from another node. The goal is to activate all nodes in the network. Applied to our setting, the wake-up problem is solved after the first non-colliding transmission is achieved (selection), whereas k-selection is more difficult because k different non-colliding transmissions are needed. On the other hand, the data-arrival model used to study wake-up is more general in that nodes may be activated (in our terms: messages arrive) at different times. Nevertheless, it might be interesting to study the overhead cost of solving k-selection with respect to wake-up. After all, a batched arrival is a particular case of an arbitrary arrival schedule. However, for a model like ours, where n is unknown and collisions cannot be detected, the protocols proposed in [11] are super-quadratic on n. That is, they do not perform better than the trivial

k-selection lower bound of k matched here asymptotically. A better upper bound for wake up of $O(n/\log n)$ was shown in [18], but it requires knowledge of n because the parametric error ϵ is bounded by n. In the same paper it was proved a $\Omega(n/\log n)$ lower bound that seemingly contradicts our results. However, such lower bound applies only to non-adaptive protocols modeled by a fixed sequence of probabilities for each node. Our protocols are execution dependent.

Monotonic back-off strategies for contention resolution of batched arrivals of k packets on simple multiple access channels, a problem that can be seen as static k-selection, have been analyzed in [4]. In that paper, it is shown that *r-exponential back-off*, a monotonic technique used widely that has proven to be efficient for many practical applications is in $\Theta(k \log^{\log r} k)$ for batched arrivals. The best strategy shown is the so-called LOGLOG-ITERATED BACK-OFF, that is based on each node choosing one slot in a window to transmit. The algorithm uses widows whose length increases by doubling its length, and for each window of length w, there is a sequence of $\lg \lg w$ consecutive windows of that length. It is shown that the LOGLOG-ITERATED BACK-OFF protocol has a makespan in $\Theta(k \log\log k/\log\log\log k)$ with probability at least $1-1/k^c, c>0$, which does not use any knowledge of k or n. In the same paper, the sawtooth technique used in [14] is informally described in a paragraph while pointing out that it yields linear time for contention resolution thanks to non-monotonicity, but no analysis is provided.

Later on, an optimal protocol for Gossiping in Radio Networks was presented in [8]. The sawtooth technique embedded in [14] is used in that paper as a subroutine to resolve contention in linear time as in EXP BACK-ON/BACK-OFF. However, the algorithm makes use of n to achieve the desired probability of success and the analysis is only asymptotical.

A randomized adaptive protocol for static k-selection in a one-hop Radio Network without collision detection was presented in [9]. The protocol is denoted LOG-FAILS ADAPTIVE$(\xi_\delta, \xi_\beta, \xi_t, \varepsilon)$, where ξ_δ, ξ_β, and ξ_t are constants such that $0 < \xi_\delta < 1$, $0 < \xi_\beta < 0.27$ and $0 < \xi_t \leq 1/2$, $1/\xi_t \in \mathbb{N}$. By tuning these constants appropriately, it was shown in [9] that the protocol is able to solve the problem in $(e+1+\xi)k + O(\log^2(1/\varepsilon))$ steps with probability at least $1-2\varepsilon$, where $\xi > 0$ is an arbitrarily small constant and $0 < \varepsilon \leq 1/(n+1)$. Modulo a constant factor, the protocol is optimal if $\varepsilon \in \Omega(2^{-\sqrt{n}})$. However, the algorithm makes use of the value of ε, which must be upper bounded as above in order to guarantee the running time. Therefore, knowledge of n is required.

Our Results: In this paper, we present a novel randomized protocol for static k-selection in a one-hop Radio Network, and we recreate a previously used technique suiting the constants for our purpose and without making use of n. Both protocols work without collision detection and do not require information about the number of contenders. As mentioned, these protocols are called ONE-FAIL ADAPTIVE(δ) and EXP BACK-ON/BACK-OFF(δ). The model for these two protocols differ with respect to the channel feedback required. While in EXP

Back-on/Back-off(δ) only the transmitter needs a confirmation that its transmission was unique, in One-fail Adaptive(δ) all nodes must receive such confirmation. In a one-hop Radio Network, such resource comes at no expense. But the difference may be relevant in other settings. It is proved that One-fail Adaptive(δ) solves static k-selection within $2(\delta+1)k+O(\log^2 k)$ steps, with probability at least $1-2/(1+k)$, for $\mathrm{e} < \delta \leq \sum_{j=1}^{5}(5/6)^j$. On the other hand, Exp Back-on/Back-off(δ) is shown to solve static k-selection within $4(1+1/\delta)k$ steps with probability at least $1-1/k^c$ for some constant $c > 0$, $0 < \delta < 1/\mathrm{e}$, and big enough k. Given that k is a lower bound for this problem, both protocols are optimal (modulo a small constant factor) for big enough number of contenders.

Observe that the bounds and the probabilities obtained are given as functions of the parameter k, as done in [4], since this is the input parameter of our version of the problem. A fair comparison with the results obtained as function of k and n would require that k is large enough, so that $n = \Omega(k^c)$, for some constant c. Both protocols presented are of interest because, although protocol Exp Back-on/Back-off is simpler, One-fail Adaptive achieves a better multiplicative factor, although the constant in the sublinear additive factor may be big for small values of k.

Additionally, results of the evaluation by simulation of the average behavior of One-fail Adaptive and Exp Back-on/Back-off and a comparison with Log-fails Adaptive and Loglog-iterated Back-off are presented. Both algorithms One-fail Adaptive and Exp Back-on/Back-off run faster than Loglog-iterated Back-off on average, even for small values of k. Although Loglog-iterated Back-off has higher asymptotic complexity, one may have expected that it may run fast for small networks. On the other hand, the knowledge on a bound of k assumed by Log-fails Adaptive seems to provide an edge with respect to One-fail Adaptive and Exp Back-on/Back-off for large values of k. However, Log-fails Adaptive has a much worse behavior than the proposed protocols for small to moderate network sizes ($k \leq 10^5$). In any case, for all values of k simulated, One-fail Adaptive and Exp Back-on/Back-off have a very stable and efficient behavior.

3 One-fail Adaptive

As in Log-fails Adaptive [9], One-fail Adaptive is composed by two interleaved randomized algorithms, each intended to handle the communication for different levels of contention. One of the algorithms, which we call AT, is intended for delivering messages while the number of nodes contending for the channel is above some logarithmic threshold (to be defined later). The other algorithm, called BT, has the purpose of handling message delivery after that number is below that threshold. Nonetheless, a node may transmit using the BT (resp. AT) algorithm even if the number of messages left to deliver is above (resp. below) that threshold.

Algorithm 1: One-fail Adaptive(δ). Pseudocode for node x. δ is a constant such that $e < \delta \leq \sum_{j=1}^{5}(5/6)^j$.

```
upon message arrival do
    κ̃ ← δ + 1                                              // Density estimator
    σ ← 0                                                   // Messages-received counter
    start tasks 1, 2 and 3
Task 1
    foreach communication-step = 1, 2, ... do
        if communication-step ≡ 0 (mod 2) then              // BT-step
            transmit ⟨x, message⟩ with prob 1/(1 + log(σ + 1))
        else                                                // AT-step
            transmit ⟨x, message⟩ with probability 1/κ̃
            κ̃ ← κ̃ + 1
Task 2
    upon reception from other node do
        σ ← σ + 1
        if communication-step ≡ 0 (mod 2) then              // BT-step
            κ̃ ← max{κ̃ − δ, δ + 1}
        else                                                // AT-step
            κ̃ ← max{κ̃ − δ − 1, δ + 1}
Task 3
    upon message delivery stop
```

Both algorithms, AT and BT, are based on transmission trials with certain probability and what distinguishes them is just the specific probability value used. It is precisely the particular values of probability used in each algorithm what differentiates One-fail Adaptive from Log-fails Adaptive. For the BT algorithm, the probability of transmission is inversely logarithmic on the number of messages already transmitted, while in Log-fails Adaptive that probability was fixed. For the AT algorithm the probability of transmission is the inverse of an estimation on the number of messages left to deliver. In One-fail Adaptive this estimation is updated continuously, whereas in Log-fails Adaptive it was updated after some steps without communication. These changes yield a protocol still linear, but now it is not necessary to know n. Further details can be seen in Algorithm 1.

Observe in Algorithm 1 that initially all nodes transmit with probability 1 in the BT slots. This is rather sensible if the chances of having a single node transmitting are high (which is reasonable in a lightly loaded system). If this is not the case, Algorithm 1 could be changed so the initial probability is 1/2. Although such improvement would not change the asymptotic behavior, it would speed up the execution in practice for some values of k.

For clarity, Algorithms AT and BT are analyzed separately taking into account in both analyses the presence of the other. We show first the efficiency of the AT algorithm in producing successful transmissions while the number of messages left is above some logarithmic threshold, and afterwards the efficiency of the BT algorithm handling the communication after that threshold is crossed. For the latter, we use standard probability computations to show our time upper bound. For the AT algorithm, we use concentration bounds to show that the messages are delivered with large enough probability, while the density estimator $\widetilde{\kappa}$ does not exceed the actual number of messages left. This second proof is more involved since it requires some preliminary lemmas as follows.

Communication steps are referred to by the name of the algorithm used, i.e. a communication step is either an AT-step or a BT-step. The following notation will be used throughout the analysis.

Let κ be the number of messages not delivered yet (i.e., the number of active nodes), called the *density*, and let $\widetilde{\kappa}$ be called the *density estimator*. Consider the execution of Algorithm 1 divided in *rounds* as follows. The first round begins with the first step of the execution, and a new round starts on each step that $\widetilde{\kappa}$ reaches or exceeds a multiple of $\tau \triangleq 300\delta \ln(1+k)$ for the first time. (Hence, a new round may start only in an AT-step.) More precisely, let the rounds be numbered as $r \in \{1, 2, \dots\}$ and the AT-steps within a round as $t \in \{1, 2, \dots\}$. Let T_r be the set of AT-steps of round r. Let $\widetilde{\kappa}_{r,t}$ be the density estimator used at the AT-step t of round r. Then,

$$\forall i, j, t \in \mathbb{N} : \widetilde{\kappa}_{j,1} \geq (j-1)\tau \wedge ((i < j \wedge t \in T_i) \Rightarrow \widetilde{\kappa}_{i,t} < (j-1)\tau).$$

Thus, round 1 is the sequence of AT-steps from initialization when $\widetilde{\kappa} = 1$ until the last step before $\widetilde{\kappa} \geq \tau$ for the first time, round 2 begins on the AT-step when $\widetilde{\kappa} \geq \tau$ for the first time and ends right before $\widetilde{\kappa} \geq 2\tau$ for the first time, and so on. Let $X_{r,t}$ be an indicator random variable such that, $X_{r,t} = 1$ if a message is delivered at the AT-step t of round r, and $X_{r,t} = 0$ otherwise. Let $\kappa_{r,t}$ be the density at the beginning of the AT-step t of round r. Then, $Pr(X_{r,t} = 1) = (\kappa_{r,t}/\widetilde{\kappa}_{r,t})(1 - 1/\widetilde{\kappa}_{r,t})^{\kappa_{r,t}-1}$ is the probability of a successful transmission in the AT-step t of round r. Also, for a round r, let the number of messages delivered in the interval of AT-steps $[1, t)$ of r be denoted as $\sigma_{r,t}$.

The following intermediate results will be useful. First, we state the following useful facts. The proof of Fact 2 is included in the Appendix for the interested reader.

Fact 1 *[25, §2.68]* $e^{x/(1+x)} \leq 1 + x \leq e^x, 0 < |x| < 1.$

Fact 2 *Given any constant* $a > 1$*, the function* $f : \mathbb{R}^+ \to \mathbb{R}^+$*, such that* $f(x) \triangleq (a/x)(1 - 1/x)^{a-1}$*, is non decreasing for* $x < a$ *and maximized for* $x = a$*.*

Lemma 1 *For any round* r *and any* $t, t+1 \in T_r$ *such that* $\widetilde{\kappa}_{r,t} < \kappa_{r,t}$*, it is* $Pr(X_{r,t} = 1) \leq Pr(X_{r,t+1} = 1)$*, conditioned to the fact that at the AT-step* t *of round* r *there was no successful transmission (and hence,* $\widetilde{\kappa}_{r,t+1} = \widetilde{\kappa}_{r,t} + 1$*).*

Proof We want to show

$$\frac{\kappa_{r,t}}{\widetilde{\kappa}_{r,t}} \left(1 - \frac{1}{\widetilde{\kappa}_{r,t}}\right)^{\kappa_{r,t}-1} \leq \frac{\kappa_{r,t+1}}{\widetilde{\kappa}_{r,t+1}} \left(1 - \frac{1}{\widetilde{\kappa}_{r,t+1}}\right)^{\kappa_{r,t+1}-1}$$

Given that the density estimator was increased between the t-th AT-step and the $(t+1)$-st AT-step and that $\delta > 1$, we know that there was no successful transmission, neither at the AT-step t, nor at the BT-step between the AT-steps t and $t+1$ (see Algorithm 1 Lines 7-11). Thus, $\kappa_{r,t+1} = \kappa_{r,t}$. Replacing,

$$\frac{\kappa_{r,t}}{\widetilde{\kappa}_{r,t}}\left(1-\frac{1}{\widetilde{\kappa}_{r,t}}\right)^{\kappa_{r,t}-1} \leq \frac{\kappa_{r,t}}{\widetilde{\kappa}_{r,t}+1}\left(1-\frac{1}{\widetilde{\kappa}_{r,t}+1}\right)^{\kappa_{r,t}-1}$$

Which, due to Fact 2, is true for $\widetilde{\kappa}_{r,t} < \kappa_{r,t}$. □

Lemma 2 *For any round r where $\widetilde{\kappa}_{r,1} \leq \kappa_{r,1}-\gamma$, $\gamma \geq (\delta-1)(3-\delta)/(\delta-2) \geq 0$, and any $t, t+1 \in T_r$ such that $\delta < \widetilde{\kappa}_{r,t} \leq \kappa_{r,t}$, and $\delta - 1 < (\kappa_{r,t}-\gamma)(\kappa_{r,t}-\gamma-1)/(\kappa_{r,t}-\gamma+1)$, it is $Pr(X_{r,t}=1) \geq Pr(X_{r,t+1}=1)$, conditioned to the fact that at the AT-step t of round r there was a successful transmission (and hence, $\widetilde{\kappa}_{r,t+1} < \widetilde{\kappa}_{r,t}$).*

Proof We want to show

$$\frac{\kappa_{r,t}}{\widetilde{\kappa}_{r,t}}\left(1-\frac{1}{\widetilde{\kappa}_{r,t}}\right)^{\kappa_{r,t}-1} \geq \frac{\kappa_{r,t+1}}{\widetilde{\kappa}_{r,t+1}}\left(1-\frac{1}{\widetilde{\kappa}_{r,t+1}}\right)^{\kappa_{r,t+1}-1} .$$

Given that the density estimator was reduced between the t-th AT-step and the $(t+1)$-st AT-step, we know that, there were successful transmissions. Furthermore, notice that even if the density is superlogarithmic, the BT algorithm could deliver a message in the BT-step between the AT-steps t and $t+1$ (see Algorithm 1 Lines 7-11). Hence, we have to consider three cases: either there was only one successful transmission at the AT-step t, or there was only one successful transmission at the BT-step between the AT-steps t and $t+1$, or there were successful transmissions in both. Thus, we have to show that

$$\frac{\kappa_{r,t}}{\widetilde{\kappa}_{r,t}}\left(1-\frac{1}{\widetilde{\kappa}_{r,t}}\right)^{\kappa_{r,t}-1} \geq \tag{1}$$

$$\frac{\kappa_{r,t}-1}{\widetilde{\kappa}_{r,t}-\delta}\left(1-\frac{1}{\widetilde{\kappa}_{r,t}-\delta}\right)^{\kappa_{r,t}-2}, \text{ if BT-step not successful,} \tag{2}$$

$$\frac{\kappa_{r,t}-1}{\widetilde{\kappa}_{r,t}-\delta+1}\left(1-\frac{1}{\widetilde{\kappa}_{r,t}-\delta+1}\right)^{\kappa_{r,t}-2}, \text{ if AT-step not successful,} \tag{3}$$

$$\frac{\kappa_{r,t}-2}{\widetilde{\kappa}_{r,t}-2\delta}\left(1-\frac{1}{\widetilde{\kappa}_{r,t}-2\delta}\right)^{\kappa_{r,t}-3}, \text{ if both steps successful.} \tag{4}$$

Similar results were shown in the proof of Lemma 3.2 in [9], and the proofs here follow similar lines. We start by showing (1) $\geq$ (3). Reordering the inequality (1) $\geq$ (3) it is obtained that

$$\frac{\widetilde{\kappa}_{r,t}-\delta}{\widetilde{\kappa}_{r,t}}\left(\frac{\widetilde{\kappa}_{r,t}-1}{\widetilde{\kappa}_{r,t}}\frac{\widetilde{\kappa}_{r,t}-\delta+1}{\widetilde{\kappa}_{r,t}-\delta}\right)^{\kappa_{r,t}-1} \geq \frac{\kappa_{r,t}-1}{\kappa_{r,t}}. \tag{5}$$

The left-hand side is monotonically non-increasing for $\delta < \widetilde{\kappa}_{r,t} \leq \kappa_{r,t}$, which can be shown using calculus. Then, given that $\widetilde{\kappa}_{r,t} \leq \widetilde{\kappa}_{r,1} - \sigma_{r,t} \leq \kappa_{r,1} - \sigma_{r,t} - \gamma \leq \kappa_{r,t} - \gamma$, it is enough to show

$$\frac{\kappa_{r,t}}{\kappa_{r,t}-1} \cdot \frac{\kappa_{r,t}-\gamma-\delta}{\kappa_{r,t}-\gamma} \cdot \left(\frac{\kappa_{r,t}-\gamma-1}{\kappa_{r,t}-\gamma} \frac{\kappa_{r,t}-\gamma-\delta+1}{\kappa_{r,t}-\gamma-\delta}\right)^{\kappa_{r,t}-1} \geq 1. \quad (6)$$

As assumed in the statement of the lemma, the left-hand side of this inequality is monotonically non-increasing on $\kappa_{r,t}$ for $\gamma \geq (\delta-1)(3-\delta)/(\delta-2)$ and $\delta - 1 < (\kappa_{r,t}-\gamma)(\kappa_{r,t}-\gamma-1)/(\kappa_{r,t}-\gamma+1)$, which again can be shown using calculus. Then, it is enough to show that, in the limit, the left-hand side of Inequality 6 tends to 1. Also verifiable using calculus. The details are omitted for brevity.

From (1) $\geq$ (3), and given that $\kappa_{r,t}-1 \geq \widetilde{\kappa}_{r,t}-\delta$, we know from Fact 2 that (3) $\geq$ (2). Then, transitively, we know that also (1) $\geq$ (2) for the conditions of this lemma.

Using the same techniques as for (1) $\geq$ (3), it can be shown that (1) is at least as large as

$$\frac{\kappa_{r,t}-2}{\widetilde{\kappa}_{r,t}-2\delta+2}\left(1-\frac{1}{\widetilde{\kappa}_{r,t}-2\delta+2}\right)^{\kappa_{r,t}-3}. \quad (7)$$

Given that $\kappa_{r,t} - 2 \geq \widetilde{\kappa}_{r,t} - 2\delta$, we know from Fact 2 that (7) $\geq$ (4). Then, transitively, we know that (1) $\geq$ (4) for the conditions of this lemma. □

Lemma 3 *For any β such that $(\delta+1)\ln\beta > 1$, and for any round r where $\kappa_{r,1} - \alpha \leq \widetilde{\kappa}_{r,1} < \kappa_{r,1}$, $\alpha \geq 0$ and for any AT-step t in r such that $1 < \widetilde{\kappa}_{r,t} \leq \kappa_{r,t}$ and $\sigma_{r,t} \leq \kappa_{r,1}\frac{\ln\beta-1}{(\delta+1)\ln\beta-1} - \frac{(\alpha+1-t)\ln\beta-1}{(\delta+1)\ln\beta-1}$, the probability of a successful transmission is at least $Pr(X_{r,t}=1) \geq 1/\beta$.*

Proof We want to show

$$\frac{\kappa_{r,t}}{\widetilde{\kappa}_{r,t}}\left(1-\frac{1}{\widetilde{\kappa}_{r,t}}\right)^{\kappa_{r,t}-1} \geq 1/\beta.$$

Because $\widetilde{\kappa}_{r,t} \leq \kappa_{r,t}$ it is enough to prove

$$\left(1-\frac{1}{\widetilde{\kappa}_{r,t}}\right)^{\kappa_{r,t}-1} \geq 1/\beta.$$

Because $\widetilde{\kappa}_{r,t} > 1$, using Fact 1 we know that $(1-1/\widetilde{\kappa}_{r,t})^{\kappa_{r,t}-1} \geq \exp\left(-\frac{\kappa_{r,t}-1}{\widetilde{\kappa}_{r,t}-1}\right)$. Thus, it is enough to prove

$$\exp\left(-\frac{\kappa_{r,t}-1}{\widetilde{\kappa}_{r,t}-1}\right) \geq 1/\beta$$
$$\kappa_{r,t}-1 \leq (\widetilde{\kappa}_{r,t}-1)\ln\beta. \quad (8)$$

Given that nodes are active until their message is delivered, we know that $\kappa_{r,t} = \kappa_{r,1} - \sigma_{r,t}$. Additionally, we know that $\widetilde{\kappa}_{r,t} = \widetilde{\kappa}_{r,1} - \delta\sigma_{r,t} + t - \sigma_{r,t}$ (see Algorithm 1 Lines 7-11 and 15-18) and that $\widetilde{\kappa}_{r,1} \geq \kappa_{r,1} - \alpha$ by hypothesis. Replacing in Inequality 8, we obtain

$$\kappa_{r,1} - \sigma_{r,t} - 1 \leq \ln\beta(\kappa_{r,1} - \alpha - (\delta+1)\sigma_{r,t} + t - 1)$$
$$\sigma_{r,t} \leq \kappa_{r,1}\frac{\ln\beta - 1}{(\delta+1)\ln\beta - 1} - \frac{(\alpha+1-t)\ln\beta - 1}{(\delta+1)\ln\beta - 1}.$$

Which is feasible for any β such that $(\delta+1)\ln\beta > 1$ and the claim follows. □

The following notation defined for succinctness will be used in the next two lemmas. Let $S = 2\sum_{j=0}^{4}(5/6)^j\tau$ and $\gamma = (\delta-1)(3-\delta)/(\delta-2)$.

The following lemma, shows the efficiency and correctness of the AT-algorithm.

Lemma 4 *For any* $e < \delta \leq \sum_{j=1}^{5}(5/6)^j$*, if the number of messages to deliver is more than*

$$M = \frac{(\delta+1)\ln\delta - 1}{\ln\delta - 1}S + \frac{(\gamma+2\tau+1)\ln\delta - 1}{\ln\delta - 1}, \qquad (9)$$

where $S = 2\sum_{j=0}^{4}(5/6)^j\tau$ *and* $\gamma = (\delta-1)(3-\delta)/(\delta-2)$*, after running the AT-algorithm for* $(\delta+1)k$ *AT-steps, the number of messages left to deliver is reduced to at most* M *with probability at least* $1 - 1/(1+k)$*.*

Proof First, we describe the overall proof line. Then, we move into details. We start by analyzing the first round r when the density estimator is somehow "near" the actual density. As a worst case scenario, we assume that no message is delivered before that round. We show that, before the density estimator is increased enough to reach a new multiple of τ leaving round r, at least τ messages are delivered with high probability. In order to prove this, we divide the round in sub-rounds where we compute the probability of the existence of a sub-round conditioned on the messages delivered in the previous sub-round. Then, the argument is repeated over the next round r' when the density estimator is again "near" the actual density. Folding all the probabilities we show that still all these events occur with high probability.

Consider the first round r such that $\kappa_{r,1}-\gamma-2\tau \leq \widetilde{\kappa}_{r,1} < \kappa_{r,1}-\gamma-\tau$. Unless the number of messages left to deliver is reduced to at most M before, such a round exists because the density estimator is increased only one by one (see Algorithm 1 Line 11). In other words, the estimator cannot be increased by τ in one step. Furthermore, given that $\widetilde{\kappa}_{r,1} < \kappa_{r,1} - \gamma - \tau$, even if no message is transmitted in round r, it holds that $\widetilde{\kappa}_{r,t} < \kappa_{r,t}$ for any t in r by the definition of a round. Additionally, we will show that, before leaving round r, at least τ messages are delivered with big enough probability so that in some future round $r' > r$ the condition $\kappa_{r',1} - \gamma - 2\tau \leq \widetilde{\kappa}_{r',1} < \kappa_{r',1} - \gamma - \tau$ holds again.

Consider round r divided in consecutive sub-rounds of size $\tau, (5/6)\tau, (5/6)^2\tau, \ldots$ as depicted in Figure 1. (The fact that a number

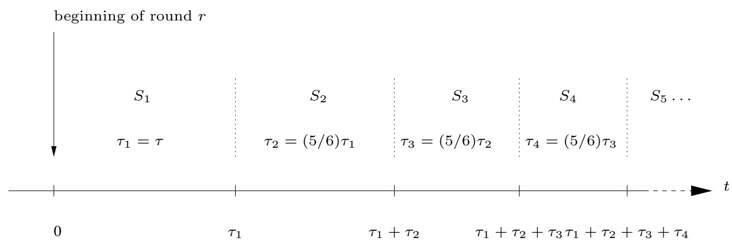

Fig. 1 Illustration for Lemma 4: subrounds of round r. Only AT steps are depicted.

of steps is an integer is omitted throughout for clarity.) More specifically, denoting the sequence of sub-rounds of r as $S_1, S_2, S_3, \dots$, and the length of sub-round S_i as τ_i, it is $\tau_1 = \tau$ and $\tau_i = (5/6)\tau_{i-1}$ for $i \geq 2$.

Notice that the number of sub-rounds in round r depends on the number of messages delivered, the reason follows. Recall that a new round starts each time that the estimator reaches a new multiple of τ. Then, even if no message is delivered, round r still has at least the sub-round S_1 because the estimator is increased by one on each AT step (see Algorithm 1 Line 11) if no message is delivered, and decreased otherwise (see Algorithm 1 Lines 15-18). However, in order to have a second sub-round S_2, some messages have to be delivered during S_1. Otherwise, a new multiple of τ would have been reached and round r would be finished. Given that each delivered message delays the end of round r by δ AT-steps (see Algorithm 1 Line 18), the existence of a complete sub-round S_2, of length $(5/6)\tau$, is conditioned on having at least $5\tau/(6\delta)$ messages delivered in sub-round S_1. We generalize this observation as follows. For each $i \geq 1$, let Y_i be a random variable such that $Y_i = \sum_{t \in S_i} X_{r,t}$. For $i \geq 2$, the existence of sub-round S_i is conditioned on $Y_{i-1} \geq 5\tau_{i-1}/(6\delta)$. In what follows, we show that with big enough probability round r has 5 sub-rounds and at least τ messages are delivered.

Even if messages are delivered in every step of the 5 sub-rounds, including messages delivered in BT-steps, the total number of messages delivered during the 5 sub-rounds would be $2\sum_{j=0}^{4}(5/6)^j\tau = S$. Then, from the definition of M in Equation 9 we have that the total number of messages delivered would be at most

$$S = M\frac{\ln\delta - 1}{(\delta+1)\ln\delta - 1} - \frac{(\gamma + 2\tau + 1)\ln\delta - 1}{(\delta+1)\ln\delta - 1}.$$

But given that $\kappa_{r,1} > M$ from the hypothesis of the lemma, we have that the total number of messages delivered would be at most

$$S < \kappa_{r,1}\frac{\ln\delta - 1}{(\delta+1)\ln\delta - 1} - \frac{(\gamma + 2\tau + 1 - t)\ln\delta - 1}{(\delta+1)\ln\delta - 1}.$$

Taking $\alpha = \gamma + 2\tau$ and $\beta = \delta$, Lemma 3 can be applied because $\kappa_{r,1} - \alpha \leq \widetilde{\kappa}_{r,1} < \kappa_{r,1}$ and $(\delta+1)\ln\beta > 1$. Hence, from Lemma 3, the expected number of messages delivered in S_i is $E[Y_i] \geq \tau_i/\delta$.

In order to use Lemmas 1 and 2, we verify first their preconditions. As argued above, $\widetilde{\kappa}_{r,t} < \kappa_{r,t}$ during the whole round. Thus, Lemma 1 can be applied. As for Lemma 2, we know that $\delta < \widetilde{\kappa}_{r,t}$ (see Algorithm 1 Lines 2, 16, and 18), $\widetilde{\kappa}_{r,1} \leq \kappa_{r,1} - \gamma$ in the round under consideration, and $\gamma \geq (\delta-1)(3-\delta)/(\delta-2)$ by hypothesis. Then, $(\kappa_{r,t} - \gamma)(\kappa_{r,t} - \gamma - 1)/(\kappa_{r,t} - \gamma + 1) > \delta - 1$ follows from $M \geq 2\delta + \gamma - 1$ and $\kappa_{r,t} > M$.

Then, we use Lemmas 1 and 2 as follows. If $\widetilde{\kappa}_{r,t+1} < \widetilde{\kappa}_{r,t}+1$, Lemma 2 holds and $Pr(X_{r,t+1} = 1) \leq Pr(X_{r,t} = 1)$. On the other hand, if $\widetilde{\kappa}_{r,t+1} = \widetilde{\kappa}_{r,t} + 1$, Lemma 1 holds and $Pr(X_{r,t+1} = 1) > Pr(X_{r,t} = 1)$. Assuming instead that $Pr(X_{r,t+1} = 1) = Pr(X_{r,t} = 1)$ can not increase the value of Y_i. Therefore, in order to bound from below Y_i, we assume that the variables $X_{r,t}, X_{r,t+1}$ for any t in r are not positively correlated and we use the following Chernoff-Hoeffding bound [26].

For $0 < \varphi < 1$,

$$\begin{cases} Pr(Y_1 \leq (1-\varphi)\tau_1/\delta) \leq e^{-\varphi^2\tau_1/(2\delta)} \\ Pr(Y_i \leq (1-\varphi)\tau_i/\delta | Y_{i-1} \geq 5\tau_{i-1}/(6\delta)) \leq e^{-\varphi^2\tau_i/(2\delta)}, \forall i : 2 \leq i \leq 5. \end{cases}$$

Taking $\varphi = 1/6$,

$$\begin{cases} Pr(Y_1 \leq 5\tau_1/(6\delta)) \leq e^{-\varphi^2 300\ln(1+k)/2} \\ Pr(Y_i \leq 5\tau_i/(6\delta) | Y_{i-1} \geq 5\tau_{i-1}/(6\delta)) \leq e^{-\varphi^2 (5/6)^{i-1} 300\ln(1+k)/2}, \\ \forall i : 2 \leq i \leq 5. \end{cases}$$

$$\begin{cases} Pr(Y_1 \leq 5\tau_1/(6\delta)) < e^{-2\ln(1+k)} \\ Pr(Y_i \leq 5\tau_i/(6\delta) | Y_{i-1} \geq 5\tau_{i-1}/(6\delta)) < e^{-2\ln(1+k)}, \forall i : 2 \leq i \leq 5. \end{cases}$$

Given that $e^{-2\ln(1+k)} \leq 1/(1+k(1+k))$, more than $(5/(6\delta))\tau_i$ messages are delivered in any sub-round S_i with probability at least $1-1/(1+k(1+k))$. Given that each success delays the end of round r in δ AT-steps, we know that, for $1 \leq i \leq 4$, sub-round S_{i+1} exists with probability at least $1-1/(1+k(1+k))$. If, after any sub-round, the number of messages left to deliver is at most M, we are done. Otherwise, conditioned on these events, the total number of messages delivered over the 5 sub-rounds is at least $\sum_{j=1}^{5} Y_j > \sum_{j=1}^{5}(5/(6\delta))^j \delta^{j-1}\tau = (\tau/\delta)\sum_{j=1}^{5}(5/6)^j \geq \tau$ because $\delta \leq \sum_{j=1}^{5}(5/6)^j$.

Now, either the number of messages left to deliver is eventually reduced to at most M before, or by the same argument used to prove the existence of round r, a round $r' > r$ will be reached such that $\kappa_{r',1} - \gamma - 2\tau \leq \widetilde{\kappa}_{r',1} < \kappa_{r',1} - \gamma - \tau$. Then, the same analysis can be repeated over the round r'. The same analysis is repeated over various rounds until all messages have been delivered or the number of messages left is at most M. Then, using conditional probability, the overall probability of success is at least $(1-1/(1+k(1+k)))^k$. Using Fact 1 twice, that probability is at least $1-1/(1+k)$.

It remains to be shown the time complexity of the AT algorithm. The difference between the number of messages to deliver and the density estimator right after initialization is less than k (see Algorithm 1 Line 2). This difference is increased with each message delivered by at most δ. Then, that difference is never more than $k(\delta+1)$. Given that the density estimator never exceeds the actual density, the claim follows. □

The following lemma shows the correctness and time complexity of the BT Algorithm.

Lemma 5 *If the number of messages left to deliver is at most*

$$M = \frac{(\delta+1)\ln\delta - 1}{\ln\delta - 1}S + \frac{(\gamma+2\tau+1)\ln\delta - 1}{\ln\delta - 1},$$

where $S = 2\sum_{j=0}^{4}(5/6)^j\tau$ *and* $\gamma = (\delta-1)(3-\delta)/(\delta-2)$*, there exists a constant* $\xi > 0$ *such that, after running the BT Algorithm for* $\xi \log k \ln(1+k)$ *BT-steps, all messages are delivered with probability at least* $1 - 1/(1+k)$*.*

Proof Let $\sigma(t)$ be the number of messages delivered up to BT-step t. Then, the probability that a given message is not delivered at BT-step t is

$$1 - \frac{1}{1+\log(\sigma(t)+1)}\left(1 - \frac{1}{1+\log(\sigma(t)+1)}\right)^{k-\sigma(t)-1}.$$

Which, given that $\sigma(t) \geq k - M$, is at most

$$1 - \frac{1}{1+\log(k+1)}\left(1 - \frac{1}{1+\log(k-M+1)}\right)^{M-1}.$$

Therefore, the probability that a given message is not delivered for $\xi \log k \ln(1+k)$ BT-steps is at most

$$\left(1 - \frac{1}{1+\log(k+1)}\left(1 - \frac{1}{1+\log(k-M+1)}\right)^{M-1}\right)^{\xi \log k \ln(1+k)}.$$

Thus, we want to show,

$$\left(1 - \frac{1}{1+\log(k+1)}\left(1 - \frac{1}{1+\log(k-M+1)}\right)^{M-1}\right)^{\xi \log k \ln(1+k)} \leq 1/(1+k).$$

Using Fact 1 twice,

$$\xi \geq \frac{1+\log(k+1)}{\log k}\exp\left(\frac{M-1}{\log(k-M+1)}\right)$$

Since $M = c\ln(1+k)$, for some constant c,

$$\xi \geq \frac{1+\log(k+1)}{\log k}\exp\left(\frac{c\ln(1+k)-1}{\log(k-c\ln(1+k)+1)}\right)$$

Which is at most a constant. □

Algorithm 2: Window size adjustment in Exp Back-on/Back-off(δ). $0 < \delta < 1/\mathrm{e}$ is a constant.

```
for i = {1, 2, ...} do
    w ← 2^i
    while w ≥ 1 do
        Choose uniformly a step within the next w steps
        w ← w · (1 − δ)
```

We establish now the main theorem, which is direct consequence of Lemmas 4 and 5.

Theorem 3 *For any* $e < \delta \leq \sum_{j=1}^{5}(5/6)^j$ *and for any one-hop Radio Network under the model detailed in Section 1,* One-fail Adaptive *solves static* k*-selection within* $2(\delta+1)k + O(\log^2 k)$ *communication steps, with probability at least* $1 - 2/(1+k)$.

4 Exp Back-on/Back-off

The algorithm presented in this section is based in contention windows. That is, each node repeatedly chooses uniformly one time slot within an interval, or *window*, of time slots to transmit its message. Regarding the size of such window, our protocol follows a back-on/back-off strategy. Namely, the window is increased in an outer loop and decreased in an inner loop, as detailed in Algorithm 2.

The intuition for the algorithm is as follows. Let m be the number of messages left at a given time right before using a window of size w. We can think of the algorithm as a random process where m balls (modelling the messages) are dropped uniformly in w bins (modelling time slots). We will show that, if $m \leq w$, for large enough m, with high probability, at least a constant fraction of the balls fall alone in a bin. Now, we can repeat the process removing this constant fraction of balls and bins until all balls have fallen alone. Since nodes do not know m, the outer loop increasing the size of the window is necessary. The analysis follows.

Lemma 6 *For* $k \geq m \geq (2\mathrm{e}/(1-\mathrm{e}\delta)^2)(1 + (\beta + 1/2)\ln k)$, $0 < \delta < 1/\mathrm{e}$, $m \leq w$, *and* $\beta > 0$, *if* m *balls are dropped in* w *bins uniformly at random, the probability that the number of bins with exactly one ball is less than* δm *is at most* $1/k^\beta$.

Proof Since a bigger number of bins can only reduce the number of bins with more than one ball, if the claim holds for $w = m$ it also holds for $w > m$. Thus, it is enough to prove the first case. The probability for a given ball to fall alone in a given bin is $(1/m)(1-1/m)^{m-1} \geq 1/(\mathrm{e}m)$. Let X_i be a random variable that indicates if there is exactly one ball in bin i. Then, $Pr(X_i = 1) \geq 1/\mathrm{e}$. To handle the dependencies that arise in balls and bins problems, we approximate the joint distribution of the number of balls in all bins by assuming the load in each bin is an independent Poisson random variable with mean 1. Let X be a random variable that indicates the total number of bins with exactly one ball. Then, $\mu = E[X] = m/\mathrm{e}$. Using Chernoff-Hoeffding bounds [26], $Pr(X \leq \delta m) \leq \exp\left(-m\left(1-\mathrm{e}\delta\right)^2/(2\mathrm{e})\right)$, because $0 < \delta < 1/\mathrm{e}$.

As shown in [26], any event that takes place with probability p in the Poisson case takes place with probability at most $p\mathrm{e}\sqrt{m}$ in the exact case. Then, we want to show that $\exp\left(-m(1-\mathrm{e}\delta)^2/(2\mathrm{e})\right)\mathrm{e}\sqrt{m} \leq k^{-\beta}$, which is true for $m \geq \frac{2\mathrm{e}}{(1-\mathrm{e}\delta)^2}\left(1+\left(\frac{1}{2}+\beta\right)\ln k\right)$. □

Theorem 4 *For any constant* $0 < \delta < 1/\mathrm{e}$*,* Exp Back-on/Back-off *solves static k-selection within* $4(1+1/\delta)k$ *steps with probability at least* $1-1/k^c$*, for some constant $c > 0$ and big enough k.*

Proof Consider an execution of the algorithm on k nodes. Let a round be the sequence of time steps corresponding to one iteration of the inner loop of Algorithm 2, i.e. the time steps of a window. Let a phase be the sequence of rounds corresponding to one iteration of the outer loop of Algorithm 2, i.e. when the window is monotonically reduced.

Consider the first round when $k \leq w < 2k$. Assume no message was transmitted successfully before. (Any messages transmitted could only reduce the running time.) By Lemma 6, we know that, for $0 < \delta < 1/\mathrm{e}$ and $\beta > 0$, at least δk messages are transmitted in this round with probability at least $1-1/k^\beta$, as long as $k \geq \tau$, where $\tau \triangleq (2\mathrm{e}/(1-\mathrm{e}\delta)^2)(1+(\beta+1/2)\ln k)$.

Conditioned on this event, for some $\delta_1 \geq \delta$ fraction of messages transmitted in the first round, using the same lemma we know that in the following round at least $\delta(1-\delta_1)k$ messages are transmitted with probability at least $1-1/k^\beta$, as long as $(1-\delta_1)k \geq \tau$. This argument can be repeated for each subsequent round until the number of messages left to be transmitted is less than τ. Furthermore, given that the size of the window is monotonically reduced within a phase until $w = 1$, even if the fraction of messages transmitted in each round is just δ, the overall probability of reducing the number of messages left from k to τ within this phase is at least $(1-1/k^\beta)^{\log_{1/(1-\delta)}(2k)}$.

Consider now the first round of the following phase, i.e. when $2k \leq w < 4k$. Assume that at most τ nodes still hold a message to be transmitted. Using the union bound, the probability that two or more of m nodes choose a given step in a window of size w is at most $\binom{m}{2}/w^2$. Applying again the union bound, the probability that in any step two or more nodes choose to transmit is at most $\binom{m}{2}/w \leq \binom{\tau}{2}/(2k) = \tau(\tau+1)/(4k)$.

Therefore, using conditional probability, in order to complete the proof, it is enough to show that

$$\left(1-\frac{\tau(\tau+1)}{4k}\right)\left(1-\frac{1}{k^\beta}\right)^{\log_{1/(1-\delta)}(2k)} \geq 1-\frac{1}{k^c}, \text{ for some constant } c>0$$

$$\exp\left(-\frac{\tau(\tau+1)}{4k-\tau(\tau+1)}-\frac{\log_{1/(1-\delta)}(2k)}{k^\beta-1}\right) \geq \exp\left(-\frac{1}{k^c}\right)$$

$$\frac{\tau(\tau+1)}{4k-\tau(\tau+1)}+\frac{\log_{1/(1-\delta)}(2k)}{k^\beta-1} \leq \frac{1}{k^c}. \quad (10)$$

Given that δ is a constant and fixing $\beta > 0$ as a constant, Inequality 10 is true for some constant $c < \min\{1, \beta\}$, for big enough k. Telescoping the number of steps up to the first round when $w = 4k$, the running time is less than $4k + 2k\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}(1-\delta)^j/2^i = 4(1+1/\delta)k$. □

5 Evaluation

In order to evaluate the expected behavior of the algorithms One-fail Adaptive and Exp Back-on/Back-off, and compare it with the previously proposed algorithms Loglog-iterated Back-off and Log-fails Adaptive, we have simulated the four algorithms. The simulations measure the number of steps that the algorithms take until the static k-selection problem has been solved, i.e., each of the k activated nodes of the Radio Network has delivered its message, for different values of k. Several of the algorithms have parameters that can be adapted. The value of these parameters is the same for all the simulations of the same algorithm (except the parameter ε of Log-fails Adaptive that has to depend on k). For Exp Back-on/Back-off(δ) the parameter is chosen to be $\delta = 0.366$. For One-fail Adaptive(δ) the parameter is chosen to be $\delta = 2.72$. For Log-fails Adaptive($\xi_\delta, \xi_\beta, \xi_t, \varepsilon$), the parameters are chosen to be $\xi_\delta = \xi_\beta = 0.1$ and $\varepsilon \approx 1/(k+1)$, while two values of ξ_t have been used, $\xi_t = 1/2$ and $\xi_t = 1/10$. As mentioned, the constants $0 < \xi_\delta < 1$, $0 < \xi_\beta < 0.27$ and $0 < \xi_t \leq 1/2$, $1/\xi_t \in \mathbb{N}$, are defined in the algorithm in [9]. Intuitively, ξ_t is the ratio BT/AT steps, which in that algorithm may be less than one, and ξ_β and ξ_δ tune the increase and decrease of the estimator respectively, which in that algorithm is not increased in every round. Finally, Loglog-iterated Back-off is also simulated.

Figure 2 presents the average number of steps taken by the simulation of the algorithms. The plot shows the the average of 10 runs for each algorithm as a function of k. In this figure it can be observed that Log-fails Adaptive takes significantly larger number of steps than the other algorithms for moderately small values of k (up to 10^5). Beyond $k = 10^5$ all algorithms seem to have a similar behavior.

A higher level of detail can be obtained by observing Table 1, which presents the ratio obtained by dividing the number of steps (plotted in Figure 2) by the value of k, for each k and each algorithm. In this table, the bad

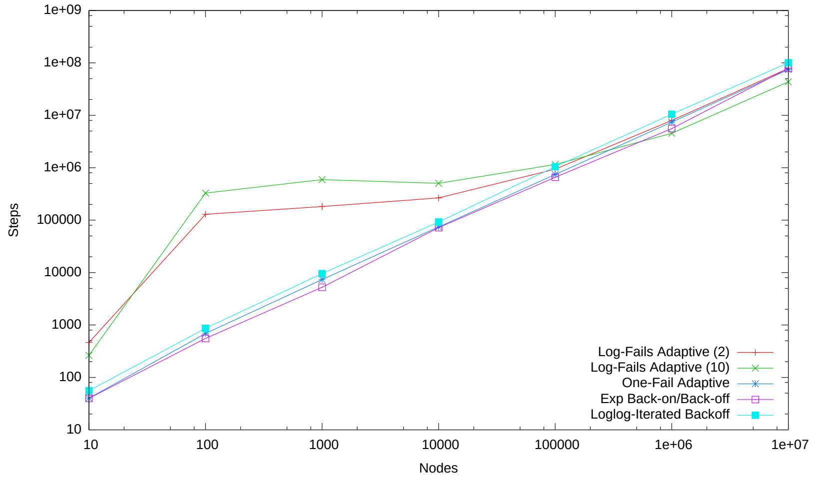

Fig. 2 Number of steps to solve static k-selection, per number of nodes k.

k	10	10^2	10^3	10^4	10^5	10^6	10^7	Analysis
Log-fails Adaptive $\xi_t = 1/2$	46.4	1292.4	181.9	26.6	9.4	8.0	7.8	7.8
Log-fails Adaptive $\xi_t = 1/10$	26.3	3289.2	593.8	50.3	11.5	4.5	4.4	4.4
One-fail Adaptive	4.0	6.9	7.4	7.4	7.4	7.4	7.4	7.4
Exp Back-on/Back-off	4.0	5.5	5.2	7.2	6.6	5.6	7.9	14.9
Loglog-iterated Back-off	5.6	8.6	9.6	9.2	10.5	10.5	10.1	$\Theta\left(\frac{\log\log k}{\log\log\log k}\right)$

Table 1 Ratio steps/nodes as a function of the number of nodes k.

behavior of Log-fails Adaptive for moderate values of k can be observed, with values of the ratio well above those for large k. It seems like the value of ξ_t used has an impact in this ratio, so that the smaller value $\xi_t = 1/10$ causes larger ratio values. Surprisingly, for large values of k ($k \geq 10^6$), the ratios observed are almost exactly the constant factors of k obtained from the analysis [9]. (Recall that all the analyses we refer to are with high probability while the simulation results are averages.) This may indicate that the analysis with high probability is very tight and that the term $O(\log^2(1/\varepsilon))$ that appears in the complexity expression is mainly relevant for moderate values of k. In addition, it seems to imply that the random processes involved in this algorithm have a small variance. The ratio obtained for large k by Log-fails Adaptive with $\xi_t = 1/10$ is the smallest we have obtained in the set of simulations. Observe that these results do not imply that the worst case ratio can be bounded in the worst case. In fact, they can be arbitrarily high. Loglog-iterated Back-off, on its hand, seems to have a constant ratio of around 10. In reality this ratio is not constant but, since it is sublogarithmic, this fact can not be observed for the (relatively small) values of k simulated.

Regarding the ratios obtained for the algorithms proposed in this paper, they seem to show that the constants obtained in the analyses (with high probability) are very accurate. Starting at moderately large values of k (10^3 and up) the ratio for One-fail Adaptive becomes very stable and equal to the value of 7.4 obtained in the analysis. The ratios for the Exp Back-on/Back-off simulations, on their hand, move between 4 and 8, while the analysis for the value of δ used yields a constant factor of 14.9. Hence, the ratios are off by only a small constant factor. To appreciate these values it is worth to note that the smallest ratio expected by any algorithm in which nodes use the same probability at any step is e, so these values are only a small factor away from this optimum ratio. In summary, the algorithms proposed here have small and stable ratios for all values of k considered.

6 Conclusions and Open Problems

In this work, we have shown optimal randomized protocols (up to constant factors) for static k-selection in Radio Networks that do not require any knowledge on the number of contenders. Future work includes the study of the dynamic version of the problem when messages arrive at different times under the same model, either assuming statistical or adversarial arrivals. The stability of monotonic strategies (exponential back-off) has been studied in [4]. In light of the improvements obtained for batched arrivals, the application of non-monotonic strategies to the dynamic problem is promising.

References

1. Anantharamu, L., Chlebus, B.S., Kowalski, D.R., Rokicki, M.A.: Deterministic broadcast on multiple access channels. In: Proceedings of the IEEE 29th International Conference on Computer Communications, pp. 146–150 (2010)
2. Balador, A., Movaghar, A., Jabbehdari, S.: History based contention window control in ieee 802.11 mac protocol in error prone channel. Journal of Computer Science **6**(2), 205–209 (2010)
3. Bar-Yehuda, R., Goldreich, O., Itai, A.: On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. Journal of Computer and System Sciences **45**, 104–126 (1992)
4. Bender, M.A., Farach-Colton, M., He, S., Kuszmaul, B.C., Leiserson, C.E.: Adversarial contention resolution for simple channels. In: ACM 17th Annual Symposium on Parallel Algorithms and Architectures, pp. 325–332 (2005)
5. Capetanakis, J.: Tree algorithms for packet broadcast channels. IEEE Transactions on Information Theory **25**(5), 505–515 (1979)
6. Chlebus, B.S.: Randomized communication in radio networks. In: P.M. Pardalos, S. Rajasekaran, J.H. Reif, J.D.P. Rolim (eds.) Handbook on Randomized Computing, vol. 1, pp. 401–456. Kluwer Academic Publishers (2001)
7. Clementi, A., Monti, A., Silvestri, R.: Selective families, superimposed codes, and broadcasting on unknown radio networks. In: Proceedings of the ACM-SIAM 12th Annual Symposium on Discrete Algorithms, pp. 709–718 (2001)
8. Farach-Colton, M., Fernández Anta, A., Mosteiro, M.A.: Optimal memory-aware sensor network gossiping. Theoretical Computer Science **472**, 60–80 (2013)
9. Fernández Anta, A., Mosteiro, M.A.: Contention resolution in multiple-access channels: k-selection in radio networks. Discrete Mathematics, Algorithms and Applications **2**(4), 445–456 (2010)

10. Fernández Anta, A., Mosteiro, M.A., Ramón Muñoz, J.: Unbounded contention resolution in multiple-access channels. In: Proceedings of the 25th International Symposium on Distributed Computing, *Lecture Notes in Computer Science*, vol. 6950, pp. 225–236. Springer-Verlag, Berlin (2011)
11. Gąsieniec, L., Pelc, A., Peleg, D.: The wakeup problem in synchronous broadcast systems. SIAM Journal on Discrete Mathematics **14**(2), 207–222 (2001)
12. Gerèb-Graus, M., Tsantilas, T.: Efficient optical communication in parallel computers. In: Proceedings of the ACM 4th Annual Symposium on Parallel Algorithms and Architectures, pp. 41–48 (1992)
13. Greenberg, A., Winograd, S.: A lower bound on the time needed in the worst case to resolve conflicts deterministically in multiple access channels. Journal of the ACM **32**, 589–596 (1985)
14. Greenberg, R.I., Leiserson, C.E.: Randomized routing on fat-trees. Advances in Computing Research **5**, 345–374 (1989)
15. Gusella, R.: A measurement study of diskless workstation traffic on an ethernet. IEEE Transactions on Communications **38**(9), 1557–1568 (1990)
16. Hayes, J.F.: An adaptive technique for local distribution. IEEE Transactions on Communications **26**, 1178–1186 (1978)
17. Indyk, P.: Explicit constructions of selectors and related combinatorial structures, with applications. In: Proceedings of the ACM-SIAM 13th Annual Symposium on Discrete Algorithms, pp. 697–704 (2002)
18. Jurdziński, T., Stachowiak, G.: Probabilistic algorithms for the wakeup problem in single-hop radio networks. Theory of Computing Systems **38**(3), 347–367 (2005)
19. Komlòs, J., Greenberg, A.: An asymptotically nonadaptive algorithm for conflict resolution in multiple-access channels. IEEE Transactions on Information Theory **31**, 303–306 (1985)
20. Kowalski, D.R.: On selection problem in radio networks. In: Proceedings of the ACM 24th Annual Symposium on Principles of Distributed Computing, pp. 158–166 (2005)
21. Kushilevitz, E., Mansour, Y.: An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks. SIAM Journal on Computing **27**(3), 702–712 (1998)
22. Leland, W.E., Taqqu, M.S., Willinger, W., Wilson, D.V.: On the self-similar nature of ethernet traffic (extended version). IEEE/ACM Transactions on Networking **2**, 1–15 (1994)
23. Martel, C.U.: Maximum finding on a multiple access broadcast network. Information Processing Letters **52**, 7–13 (1994)
24. Mikhailov, V., S.Tsybakov, B.: Free synchronous packet access in a broadcast channel with feedback. Problemy Peredachi Informatsii **14**(4), 32–59 (1978)
25. Mitrinović, D.S.: Elementary Inequalities. P. Noordhoff Ltd. - Groningen (1964)
26. Mitzenmacher, M., Upfal, E.: Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press (2005)
27. Nakano, K., Olariu, S.: A survey on leader election protocols for radio networks. In: Proceedings of the 6th International Symposium on Parallel Architectures, Algorithms and Networks, pp. 63–68 (2002)
28. Willard, D.E.: Log-logarithmic selection resolution protocols in a multiple access channel. SIAM Journal on Computing **15**, 468–477 (1986)

Appendix

A Proof of Fact 2

Proof To see that the claim is true, we find the derivative

$$f'(x) = \left(\frac{a}{x^2}\right)\left(1 - \frac{1}{x}\right)^{a-2}\left(-\frac{x-1}{x} + \frac{a-1}{x}\right).$$

Which, for any $a > 1$, is positive for $x < a$, zero for $x = a$, and negative for $x > a$. Thus, the claim follows.