Adversary Immune Size Approximation of Single-Hop Radio Networks *

Jędrzej Kabarowski, Mirosław Kutylowski, and Wojciech Rutkowski

Institute of Mathematics and Computer Science, Wrocław University of Technology
Jedrzej.Kabarowski@pwr.wroc.pl, Miroslaw.Kutylowski@pwr.wroc.pl, Wojciech.Rutkowski@pwr.wroc.pl

Abstract. We design a time and energy efficient algorithm concerning size approximation for single-hop radio networks. The most important feature of the algorithm is that it is immune against an adversary that may scramble a certain number of communication steps. The previous algorithms presented in the literature provide false estimations if an adversary causes certain communication collisions.

1 Introduction

Ad hoc networks have gained a lot of attention due to their broad potential applications. However, optimistic reports about future perspectives often disregard some fundamental design problems. While on the hardware side the advances are encouraging, many algorithmic problems of self-organization of ad hoc networks still need to be solved.

Algorithmic issues for ad hoc networks are quite different from the classical ones: the communication channel has different characteristics than in wired networks, the network might change quite fast, the network stations may move, etc. Due to technical limitations new complexity measures are to be considered: one of the most important ones is the energy cost. It is related to the amount of time that a station uses for sending or listening (not necessarily getting any message).

One of the crucial issues which has been almost completely disregarded in the algorithm design are transmission faults. Some work has been done on the hardware side – however, this approach must be limited to a “standard” fault rate. Above this level it is quite inefficient to provide immunity to transmission faults by hardware means. It seems that higher levels of communication protocols should take care of this.

Random transmission faults of physical nature are not the worst things that may happen. Since everybody may have physical access to the shared communication channel, a malicious user or an adversary may cause transmission faults at chosen moments. On the other hand, many classical algorithms (also those for ad hoc networks) have “hot spots” and their efficiency and correctness depends on a faultless communication. For this reason such algorithms are broken down through an adversary that knows the algorithm details.

* partially supported by Polish Committee for Scientific Research grant 3 T11C 033 (the third author) and the EU within the 6th Framework Programme under contract 001907 (DELIS).

Copyright Springer-Verlag, to appear in LNCS in TAMC’2006 Proceedings
This paper is concerned with size approximation of ad hoc networks, where the main focus is to make it immune to an adversary that may cause a limited number of transmission faults.

2 Model

Radio Network A radio network (RN) consists of a number of stations, which are small devices with weak computational power. Each station is equipped with a radio transmitter and a radio receiver. A station can be in the following internal states:

- dead: it means that the station did not survive deployment of the network or its battery is exhausted,
- transmitting: it means that the station broadcasts some data via its transmitter,
- receiving: it means that the receiver is switched on and the station monitors the radio channel,
- inactive: it means that the station has its antenna turned off, but the station can do some internal calculations.

We assume that a station cannot be in two different states in the same moment, in particular, it cannot simultaneously send and receive, as stated by IEEE 802 standard.

The stations communicate through a single broadcast channel. We consider here a single-hop model. That is, a message sent by one station can be received by every other station, unless other message is sent at the same time. When two or more stations send messages simultaneously, then a collision occurs. We work here with the no-collision detection model. That is, we assume that a collision is indistinguishable from a random noise which appears when no station broadcasts. (In practice, this is a strong assumption, but we would like to design solutions that do not depend on detecting collisions).

We assume that the stations have synchronized clocks and start the algorithm execution in the same moment. The execution consists in a number of synchronous steps. Each step is executed within a single time slot. Within a step a station is in one of the states listed above and cannot change it.

Complexity Measures The parameters concerned are:

- time complexity, which is the maximum number of steps executed by a station of the network during algorithm execution,
- energy usage, which is the maximum number of steps within which a station is either transmitting or listening, taken over all stations.

If a RN executes a probabilistic algorithm, we consider also the probability that it does not reach its goal within the specified number of steps.

Adversary Model We consider two factors that can influence algorithm execution. First, it is possible that burst errors occur on the broadcast channel caused by some physical conditions. Then the messages sent by the stations participating in the protocol
will be lost. The second situation is that an adversary causes collisions on the broadcast channel at certain moments. The effect is the same - nobody receives a message. This may have profound consequences, since the popular approach for size approximation is to make estimations based on observations whether certain messages came through the channel. Of course, if an algorithm works for the second scenario, it works also for the first one.

If the communication channel is scrambled all the time, no algorithm may achieve its goal. So it is necessary to limit the energy cost of an adversary. In this paper we focus on adversaries that have limited energy cost. This models an adversary (or adversaries) holding similar devices as the legitimate stations.

We assume that the legitimate users have a shared secret unknown by the adversary. The secret can be used to protect communication between the network users.

3 Size Approximation Problem

We assume that we are given a (single-hop) RN consisting of an unknown number of stations. Each station knows only its unique ID and an upper bound on the number of radio stations in this RN. (For physical reasons, in practice we always have some bound on the number of stations.) It has no information on the actual size $N$ of the network and the ID’s of the other stations in the network.

The goal of size approximation is to find a number $n$ such that, for some constants $c$ and $d$, which are parameters of the algorithm, the following inequality holds

$$\frac{1}{c} \cdot n - d \leq N < c \cdot n + d.$$

Previous results on size approximation  Size approximation problem for single-hop RN was considered in [9]. Later, more efficient solutions with respect to time and energy complexity were found. In [4] a deterministic solution for the exact counting the number of stations was presented, it runs in time $O(n)$ and has energy cost $O((\log n)^{\varepsilon})$. Paper [2] presents a size approximation algorithm with runtime $O((\log n)^{2+\varepsilon})$ and energy cost $O((\log \log n)^{\varepsilon})$ for any constant $\varepsilon > 0$.

Each of these algorithms is quite fragile. Namely, even a single transmission error could yield false estimations or no estimation at all.

Adversary immune algorithms Adversary immune algorithms for single hop RN were presented in two papers: [7] presents a randomized leader election algorithm running in time $O(\log^3 N)$ with energy cost $O(\sqrt{\log N})$. Paper [8] presents a randomized initialization algorithm with runtime $O(N)$ and energy cost $O(\sqrt{\log N})$. The adversary may use more energy than the protocol participants, namely $O(\log N)$. Both papers mentioned above assume that the approximate network size is known.

Main Result We present here the following result:
Theorem 1. Size approximation problem of a single-hop radio network consisting of \( N \) stations can be solved in time \( O(\log^{2.5} N \cdot \log \log N) \) and energy cost \( O(\log \log N \cdot \sqrt{\log N}) \) where a correct output is given with probability at least \( 1 - 2^{-z} \) where \( z = \Omega(\sqrt{\log N}) \) in the presence of an adversary with energy cost \( \log(N) \). The same (correct) answer will be known to all station except \( o(N/2\sqrt{\log N}) \) of them.

4 Basic Techniques

Basic Experiment Size approximation is usually based on the following simple trick with Bernoulli trials. Suppose we have \( K \) radio stations. If each of these stations decides to transmit a message with probability \( p \), then the probability that exactly one station transmitted equals \( K \cdot p \cdot (1 - p)^{K-1} \). It is well known that this expression is maximized for \( p = 1/K \) and the value achieved is about \( 1/e \). The event mentioned will be called SINGLE for the rest of the paper.

We repeat this experiment \( t \) times and call it a basic experiment. The expected number of SINGLES in a basic experiment is about \( t/e \). The algorithm examines the number of SINGLES for basic experiments for different probabilities \( p \). If the maximal number of SINGLES is achieved for \( p_0 \), then we take \( 1/p_0 \) as an approximation of the number of stations.

Time Windows To make things harder for an adversary, we can waste some time to reduce adversary’s ability to make collisions. For this purpose we combine \( K \) consecutive time slots into one time window. Within each time window we perform one step of the algorithm executed. The time slot within the window to be used by the algorithm is determined by a strong cryptographic pseudorandom function generating numbers in the range \( [1..K] \) from the shared secret as a seed.

Now, the adversary can still make collisions, but since the transmissions occur at random moments within a window, either the chances of a collision are reduced or the adversary has to use more energy.

Interleaving Assume that an algorithm is designed so that there are \( K \) independent groups of stations performing the same algorithm. Therefore, we can run these groups simultaneously so that each time slot is devoted to one group, but the assignment of the time slots to the groups is hidden from the adversary. From the adversary point of view it is hard to attack a single group – the situation looks like in the case of time windows. On the other hand, each time slot is utilized by the algorithm.

Assignment of the time slots to the groups are either by a pseudorandom generator yielding permutations over \( \{1, \ldots, K\} \) (then we have time windows of width \( K \)), or by a pseudorandom function generating numbers in the range \( [1..K] \), where the number denotes which group should transmit at a given time slot.

5 Algorithm Description

The algorithm consists of phases executed sequentially. Within phase \( i \) the algorithm checks if the size of the network is between \( 2^i \) and \( 2^{i+1} \) and finds an appropriate
approximation if this is the case. In fact, for small size networks we substitute the first three phases by one and execute it differently (see [6]). A phase ends with a common agreement on the size of the network. In case there is no agreement, the algorithm steps into the next phase. Now let us describe phase $i$. It consists of five subphases.

**Subphase 1.** We consider $2^i$ groups of stations, each group consisting of $2^i$ subgroups (so there are $2^{2i}$ subgroups in total).

The stations are assigned to groups independently at random. No communication is required. Namely, each station decides to join a single subgroup chosen at random. So from the point of view of a single subgroup this is a Bernoulli process with $N$ trials and success probability $2^{-2i}$. These Bernoulli processes for different subgroups are not stochastically independent, but if $2^{-2i}$ is small, then the numbers of stations assigned to subgroups have approximately the same probability distribution as in the case of independent Bernoulli trials (for a detailed discussion see [2]).

**Subphase 2.** There are 16 time slots assigned to each subgroup. We use interleaving technique for mixing the time slots assigned to the subgroups.

Each subgroup performs the basic experiment consisting of 8 trials. For this purpose, we assign probabilities of broadcasting – in the first subgroup the probability is set to $\frac{1}{3} \cdot 2^{-2i+1}$, and it decreases twice when we increase the subgroup number by 1. So in the last subgroup the probability equals $\frac{1}{3} \cdot 2^{-2i+1}$. A station that transmits at any trial is called an active subgroup member. Each active subgroup member listens during all time slots assigned to its subgroup (except for the moments when it transmits).

We will be interested in subgroups such that the broadcast probability is approximately inversely proportional to the expected number of stations within this subgroup. In that case, the expected number of SINGLES within eight trials in such a subgroup is about $\frac{8}{e} \approx 3$. Therefore, we seek subgroups with 3 SINGLES.

For each subgroup, we would like to inform all its active members whether 3 SINGLES have occurred in this subgroup. For this purpose, after eight time slots described above, we use eight additional time slots which mirror the first 8 time slots. It means that a station transmits during time slot $j$ if and only if it has transmitted in time slot $j - 8$. The message sent by this station in the mirror phase is slightly different: it contains a vector of length 8 that contains 1 at position $s$, if and only if it has received a valid message during time slot $s$, for $s \leq 8$. (Additionally, it contains a 1 at position $j$.)

It is easy to see that if at least two SINGLES have occurred, then all active subgroup members get informed about all SINGLES in this subgroup. Indeed, a station that has successfully transmitted in time slot $s$ gets a confirmation about this SINGLE within a mirror time slot from every station that has transmitted without a collision. For $s \leq 8$, a station that has successfully transmitted in trial $s$ gets ID $s$. If there is exactly one SINGLE, then all but one station knows this SINGLE, but the station that has transmitted successfully cannot say if its message came through or there was a collision. For this reason, in such a case the subgroup will not be used in subsequent subphases as the source of SINGLES.

If 3 SINGLES have occurred in a subgroup with broadcast probability $p$, then $\frac{1}{p}$ can be taken as an estimate for the number of stations in this subgroup, and $\frac{1}{p} \cdot 2^{2i}$ as an
estimate for the total number of stations. However, the algorithm takes an estimate from all subgroups with 3 SINGLES from all groups. For this purpose we collect all such subgroups, sort their list based on the subgroup number and finally take a subgroup that is in a middle position of the sorted list (so we take the median). Then we use broadcast probability \( p \) of the subgroup chosen and take \( \frac{1}{p} \cdot 2^{2i} \) as an estimate for the number of stations.

The main problem is to construct a list \( v \) of all subgroups with 3 SINGLES. This is nontrivial, since for most subgroups there are no 3 SINGLES and inspecting all subgroups by a station would require too much energy. Our strategy is first to find all subgroups with 3 SINGLES within each group – this is done during Subphase 3. We take advantage of the fact that there is a small number of such subgroups within each group. Then we perform gossiping between the groups (Subphase 4).

**Subphase 3.** Each group separately collects indexes of its subgroups with 3 SINGLES. For this purpose, \( 8\sqrt{2^i} \) time slots are used (which is much less than the number of subgroups \( 2^i \)). The idea is that only for a few neighboring subgroups 3 SINGLES should occur.

An active subgroup member listens during all \( 8\sqrt{2^i} \) time slots appointed to its group. Let us consider a subgroup \( r \) with 3 SINGLES. Let \( m = r \mod \sqrt{2^i} \). Then the active member of subgroup \( r \), which has transmitted in trial number \( j \) of its subgroup, transmits at time slot \( 8 \cdot m + j \) of its group in Subphase 3. The message transmitted contains the subgroup number.

Of course, a collision occurs, if there are two subgroups with 3 SINGLES and indexes differing by a multiple of \( \sqrt{2^i} \). However, such events have probabilities \( o(\frac{1}{2^{i}}) \) as shown in the next section and therefore can be neglected.

As the adversary can scramble the transmission, the whole procedure is repeated \( i \) times.

Each active subgroup member maintains a list \( v \) of known subgroups with 3 SINGLES. If it receives a message with a subgroup number, say \( j \), (it is a subgroup number where 3 SINGLES occurred) it checks if there is an entry for \( j \) in its local vector \( v \). If not, then \( j \) together with the group number is appended to the local vector \( v \).

Finally, using a deterministic algorithm, the active members from a group elect one representative of the group. It is done by choosing the station corresponding to the first SINGLE in the subgroup \( i \) with the lowest index and exactly 3 SINGLES which succeeds in transmitting at least one message in Subphase 3. As transmission errors can occur, let us explain some details. The main problem is that the potential leader must be sure that all other stations will agree that it is the leader. For this purpose we assume that during Subphase 3 every message includes also information which messages came through so far. The first case is that the second or the third message from subgroup \( i \) comes through. Then the first station with a SINGLE from subgroup \( i \) can consider itself the leader, since all other active stations have heard this message as well. The second case is when only the first message from subgroup \( i \) comes through. If there is another subgroup with 3 SINGLES such that at least one its message comes through, then this message confirms the message from the first active station in subgroup \( i \). So
again, it can safely decide to be the leader. The last case is that no other message comes through. Then no leader will be elected.

During the next subphases, each group will be represented by its leader.

**Subphase 4:** During this subphase the leaders collect information on subgroups with 3 SINGLES over all groups. We execute a simple gossiping algorithm among the group leaders. It consists of $\Theta(\sqrt{2^i})$ rounds, where a round uses $2^i + 2^{2i}$ time slots. During a round, $2^i$ out of $2^{2i}$ time slots are assigned to the group leaders, one slot per group. The remaining time slots are unused and serve for confusing an adversary. The choice of the slots used for communication is pseudorandom and based on the secret known to network participants. Each leader transmits during its time slot. A message sent is a collection of known pairs $(j, v_j)$, where $j$ denotes a group number and $v_j$ is the $v$ list of this group.

At each round a leader listens during $i$ time slots and updates its list of vectors $v$ by appending yet unknown pairs. Namely, a leader chooses $i$ groups at random and listens at the moments when the leaders of these groups transmit (provided that they exist).

After this part, with high probability, all active leaders have the same information about the SINGLES in all groups, so that they can get the same estimate of the number of stations in the network computed locally by each leader.

**Subphase 5:** During this subphase the non-leaders get the knowledge of leaders on 3 SINGLES collected during Subphase 4. Each leader transmits its estimate of the number of stations and each non-leader listens at some moments chosen at random. Namely, during each of $\sqrt{2^i}$ rounds the leaders use $2^i$ time slots to broadcast, each leader responsible for $O(1)$ time slots. Since the leaders know themselves with high probability, each of them can derive locally for which of the $2^i$ time slots it is responsible for. As before, time slots used for transmission are dispersed among $2^{2i}$ time slots in a pseudorandom way. Unlike in Subphase 4, each time slot devoted to transmission is now used.

During each round a non-leader listens during $i$ time slots randomly chosen from the ones used by leaders to transmit.

### 6 Algorithm Complexity and Correctness

**Energy Cost and Runtime** First we compute energy cost. Consider Phase $i$. Subphase 1 requires no communication. Energy cost of Subphase 2 is 16. During Subphase 3, if there are 3 SINGLES in a subgroup, the active subgroup members use $\sqrt{2^i} \cdot 8$ time slots to learn the other subgroups with 3 SINGLES within its group. In Subphase 4 and 5 energy cost is $(i+1) \cdot \sqrt{2^i} + O(1)$ for the leaders and $i \cdot \sqrt{2^i}$ for the other stations. Therefore, the energy expense for all phases equals $O(i \cdot \sqrt{2^i})$, which is $O(\sqrt{\log N \log \log N})$.

Time complexity is as follows: $O(1)$ for Subphase 1, $2^i \cdot 16$ for Subphase 2; Subphase 3 requires $O(2^{1.5i})$ time slots, whereas Subphase 4 and 5 require $2 \cdot (2^{2i}) \cdot \sqrt{2^i}$ time slots, as the leaders perform twice the $\sqrt{2^i}$ rounds, each consisting of $2^{2i}$ slots. After summing over all phases we get $O(\log^{2.5} N \log \log N)$. 
Correctness of the Results  There are two reasons for which a transmission within the algorithm may fail. The first one is a collision between legitimate participants of the protocol. The second one is a collision caused by an adversary. The first subphase is not a problem, since no communication takes place. During the second phase the collisions between the participants occur, but as already mentioned in Section 4 if the broadcast probability is about inverse of the number of stations choosing to broadcast, then with probability approximately \( \frac{1}{2} \) a transmission succeeds. Even if the adversary knows the size of the network he cannot change this significantly. Indeed, the number of subgroups is \( \Omega(\log^2 N) \), the subgroups where 3 SINGLES occur are dispersed at random from the adversary point of view. So in order to reduce the number of 3 SINGLES significantly the adversary has to hit \( \Theta(\log N) \) subgroups out of \( \Omega(\log^2 N) \). This occurs with probability of the magnitude \( 1/N \) and can be neglected.

Now we consider Subphase 3. We observe that the probability of getting 3 SINGLES in the subgroup with broadcast probability \( 2^{0.5\sqrt{\tau}} \) times greater or lesser than in the optimal subgroup is \( o(2^{-z}) \) for \( z = \Omega(\sqrt{\log N}) \). Indeed, the probability for 3 SINGLES in the subgroup with broadcast probability at least \( 2^{0.5\sqrt{\tau}} \) times greater than optimal \( (p_{opt} = 1/K) \), where \( K \) is the number of stations) can be estimated according to a formula from Section 4 (substituting \( p = 2^{0.5\sqrt{\tau}}/K \)):

\[
\left( \frac{8}{3} \right) \left( 2^{0.5\sqrt{\tau}} \left( \frac{1}{2} \right)^{2^{0.5\sqrt{\tau}}} \right)^3 \left( 1 - 2^{0.5\sqrt{\tau}} \left( \frac{1}{2} \right)^{2^{0.5\sqrt{\tau}}} \right)^5 < \frac{1}{N},
\]

for the subgroup with broadcast probability at least \( 2^{0.5\sqrt{\tau}} \) times smaller the estimation is:

\[
\left( \frac{8}{3} \right) \left( 2^{-0.5\sqrt{\tau}} \left( \frac{1}{2} \right)^{2^{-0.5\sqrt{\tau}}} \right)^3 \left( 1 - 2^{-0.5\sqrt{\tau}} \left( \frac{1}{2} \right)^{2^{-0.5\sqrt{\tau}}} \right)^5 < \frac{1}{2^{\sqrt{\log N}}},
\]

So it is easy to see that the probability that for at least half of the groups with the optimal broadcast probability and 3 SINGLES a collision will occur is less than \( \frac{1}{N} \).

So the number of groups with 3 SINGLES observed at Subphase 3 is \( \Omega(2^z) \) with high probability. From now on assume that the number of such groups is \( c \cdot 2^z \).

For Subphase 4 it is crucial to calculate the rate of spreading knowledge about 3 SINGLES among the leaders. Let us consider a single leader \( L \). Initially, only \( L \) knows that his group has a leader. At each round the number of stations knowing \( L \) may increase. Let \( X_t \) be this number immediately after round \( t \) of the Subphase 4. Let us consider two stages: the first one lasts as long as \( X_t \leq \frac{c \log N}{\log \log N} \).

6.1 Stage 1: \( X_t \leq \frac{c \log N}{\log \log N} \)

It is easy to see that the number of informed users will increase geometrically, eventually reaching \( \frac{c \log N}{\log \log N} \). Let us compute conditional expected value of \( X_{t+1} \), that is \( E(X_{t+1} | X_t) \). A conditional probability that a leader \( L' \) has not learned about \( L \) equals:

\[
\prod_{j=0}^{\log \log N - 1} \left( 1 - \frac{X_j}{\log \log N} \right) \leq \left( 1 - \frac{X_t}{\log \log N} \right)^{\log \log N} \leq 1 - \frac{X_t \log \log N}{\log N} + \frac{1}{2} \left( \frac{X_t \log \log N}{\log N} \right)^2
\]
(the last inequality follows from Taylor expansion). So the conditional probability that $L'$ has received the information about $L$ is at least:

$$\frac{X_t \log \log N}{\log N} \left(1 - \frac{c}{2}\right).$$

By multiplying the above value by $c \log N - X_t$ we get a lower bound of the expected number of leaders, which learn about $L$ at this round:

$$X_t \left(1 - \frac{c}{2}\right) c (\log \log N - 1) = \omega(X_t) \geq d \cdot X_t$$

for some $d$ and sufficiently large $N$.

For estimating the number of rounds in Stage 1 we call a round successful, if $X_t$ increases at least $d$ times. This occurs with probability $\Theta(1)$. So we can talk about a process with success probabilities higher than in a Bernoulli process. After $\log_d\left(\frac{c \log N}{\log \log N}\right) = O(\log \log N)$ successes Stage 1 must terminate. Using Chernoff Bound we can easily see that this happens with probability at least $1 - 2^{-\Theta(\sqrt{\log N})}$ after $\sqrt{\log N}$ rounds.

6.2 Stage 2: $X_t > \frac{c \log N}{\log \log N}$

At round $t + 1$, probability that a leader $L'$ does not learn about (yet unknown) leader $L$ is less than $1 - \frac{c}{\log \log N}$. As we have $\sqrt{\log N}$ rounds and in each round the leader $L'$ listens $\log \log N$ times, the probability that $L'$ does not learn about $L$ is at most:

$$\left(1 - \frac{c}{\log \log N}\right)^{\log \log N \sqrt{\log N}} \approx e^{-c \sqrt{\log N}}$$

So, probability that at least one of $c \log N$ leaders does not know about any of remaining leaders is at most $c \log N \cdot e^{-c \sqrt{\log N}} = O(2^{-\sqrt{\log N}})$. In the opposite case, if there is no adversary, all stations have to query only once to get the size estimate. The only problem appears when an adversary causes collisions.

An adversary may disturb the algorithm in Subphases 3 and 4 in two ways: he can collide messages so that stations cannot make a common decision in a single group, and he can collide messages so that propagation of information in Subphase 4 in order to disable possibility to derive a common quantity estimate of the network size. In the first case the adversary has to collide at least a constant fraction of groups - otherwise a smaller number of the leaders still suffices to derive the estimate. Since making a collision occurs with probability about $\log N / \log^2 N$, it follows easily from a Chernoff Bound that hitting $\Omega(\log N)$ groups has probability bounded by $e^{-g \log N \sqrt{\log N}}$ for some constant $g$ which is $o(1/N)$. In the opposite case, if there is no adversary, all stations have to query only once to get the size estimate. The only problem appears when an adversary causes collisions.

An adversary may disturb the algorithm in Subphases 3 and 4 in two ways: he can collide messages so that stations cannot make a common decision in a single group, and he can collide messages so that propagation of information in Subphase 4 in order to disable possibility to derive a common quantity estimate of the network size. In the first case the adversary has to collide at least a constant fraction of groups - otherwise a smaller number of the leaders still suffices to derive the estimate. Since making a collision occurs with probability about $\log N / \log^2 N$, it follows easily from a Chernoff Bound that hitting $\Omega(\log N)$ groups has probability bounded by $e^{-g \log N \sqrt{\log N}}$ for some constant $g$ which is $o(1/N)$.

For the first part of Subsection 4, it is easy to see that the adversary cannot change significantly each probability concerned and so the bound derived previously still holds.

Now consider a non-leader that tries to get an estimate during Subphase 4. Probability that a time slot monitored by this station is scrambled is not higher than $\log N / \log^2 N$, hence probability that every transmission is scrambled by the adversary is at most $(\log N)^{-\log \log N \sqrt{\log N}} = o(2^{-\sqrt{\log N}})$. Hence the expected number of stations that get no estimation is $o(N/2^{-\sqrt{\log N}})$. By Chernoff Bounds one can easily derive that probability that $N/2^{-\sqrt{\log N}}$ get not informed about the estimate is $o(1/N)$. 
7 Practical Implementation Issues

Good asymptotic behavior of an algorithm does not automatically mean that it is relevant for practical applications. Especially, it is often the case for algorithms where complexity measures are polylogarithmic and sublogarithmic. Moreover, for obvious reasons we should assume that the number of stations $N$ is fairly small. Also, failure probability of order $O(1/N)$ is less interesting than, say, smaller than 0.01.

We have implemented the algorithm presented and tuned several parameters. Here we summarize the most important observations. The probability of 3 SINGLES in a specific group is not symmetric with respect to the optimal subgroup (the one with broadcast probability corresponding to $N$). In order to decrease the bias of the total stations quantity estimate, we propose to use median instead of mean. The advantage of the median is that it disregards the extreme values (which are rare, but if occur, then they change the mean value significantly). Another advantage is that if two stations have different knowledge which 3 SINGLES have occurred, the median is still likely to be the same. On the other hand, the mean value has the advantage that it is less quantized so the estimate might better fit the actual size.

Below we present results of 100 000 simulations for $N \in \{2^8, 2^{16}\}$. Then there are 8 groups, each consisting of 8 subgroups. For each simulation, estimates based on mean and median were constructed. We computed the error for both estimates. Furthermore, we examined how many stations have the same knowledge and how many fail to get any estimate. Let $\nabla$ be a random variable denoting the percentage of stations that share the same knowledge on 3 SINGLES. The results are as follows:

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>median</th>
</tr>
</thead>
<tbody>
<tr>
<td>average estimation error (%)</td>
<td>-11.48</td>
<td>-2.26</td>
</tr>
<tr>
<td>average absolute value of estimation error (%)</td>
<td>35.61</td>
<td>31.75</td>
</tr>
<tr>
<td>average value of $\nabla$ (%)</td>
<td>98.30</td>
<td>98.47</td>
</tr>
<tr>
<td>standard deviation of $\nabla$ (%)</td>
<td>4.71</td>
<td>4.38</td>
</tr>
<tr>
<td>95% quantile of $\nabla$ (%)</td>
<td>92.00</td>
<td>92.90</td>
</tr>
<tr>
<td>99% quantile of $\nabla$ (%)</td>
<td>78.12</td>
<td>78.40</td>
</tr>
<tr>
<td>lack of knowledge(%)</td>
<td>1.13</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Simulation results

We can clearly see that the estimate based on median performs significantly better than the one based on mean. In only 0.7% of cases there was no subgroup with 3 SINGLES.

Final Remarks

In the algorithm presented a certain fraction of stations get no estimation on the size of the network. If we increase the time complexity to $2^{O(\sqrt{\log N})}$, then with probability $O(1/N)$ all non-leaders get informed about the estimation. It is fairly easy to see that for an algorithm with a polylogarithmic runtime and an adversary with runtime $O(\log n)$ the expected number of stations that do not get an estimate is always $\omega(1)$.

Let us remark that the results hold also if each transmission fails with a constant probability.
References