Doubly-logarithmic Energy-efficient Initialization Protocols for Single-hop Radio Networks *

Jacir L. Bordim†, Jiangtao Cui‡, Naohiro Ishii‡, and Koji Nakano†

†School of Information Science
Japan Advanced Institute of Science and Technology
1-1, Asahidai, Tatsunokuchi, Ishikawa 923-1292, Japan
(bordim,knakano)@jaist.ac.jp

‡Department of Intelligence and Computer Science
Nagoya Institute of Technology
Showa-ku, Nagoya 466-8555, Japan
cjto@egg.ics.nitech.ac.jp
ishii@ics.nitech.ac.jp

Abstract

A radio network is a distributed system with no central shared resource, consisting of n stations each equipped with a radio transceiver. One of the most important parameters to evaluate protocols in the radio networks is the number of awake time slots in which each individual station sends/receives a data packet. We are interested in devising energy-efficient initialization protocols in the single-hop radio network (RN, for short) that assign unique IDs in the range $[1, n]$ to the n stations using few awake time slots. It is known that the RN can be initialized in $O(\log \log n)$ awake time slots, with high probability, if every station knows the number n of stations in the RN. Also, it has been shown that the RN can be initialized in $O(\log n)$ awake time slots even if no station knows n. However, it has been open whether the initialization can be performed in $O(\log \log n)$ awake time slots when no station knows n.

Our main contribution is to provide the breakthrough: we show that even if no station knows n, the RN can be initialized by our protocol that terminates, with high probability, in $O(n)$ time slots with no station being awake for more than $O(\log \log n)$ time slots. We then go on to design an initialization protocol for the k-channel RN that terminates, with high probability, in $O(\frac{n}{k} + (\log n)^2)$ time slots with no station being awake for more than $O(\log \log n)$ time slots.

*Work partially supported by Grant-in-Aid for Encourangement of Young Scientists (12780213) from Ministry of Education, Science, Sports, and Culture of Japan and by The Hori Information Science Promotion Foundation.

keywords: *radio networks, initialization protocols, energy-efficient, randomized algorithms*

1 Introduction

A radio network is a distributed system with no central shared resource, consisting of n *stations* each equipped with a radio transceiver. A station may directly communicate with any other stations in the single-hop radio network (RN, for short).

In this work, we adopt the time-slotted communication model where time is assumed to be slotted and all transmissions take place at slotted boundaries. All the stations run the same protocol and they can generate random bits that provide local data on which the stations may perform computations. We also assume that in any time slot a station can tune to one radio channel and/or transmit on at most one, possibly the same, channel. A transmission involves a packet whose length is such that it can be completed within one time slot.

We employ the commonly-accepted assumption that when two or more packets are being transmitted on the channel, in the same time slot, these packets *collide* and are garbled beyond recognition. We further assume that the system has *collision detection* (CD, for short) capability. Accordingly, the channel status at the end of a time slot can be: **NULL**, if no packet has been transmitted in the current time slot, **SINGLE** if exactly one packet has been transmitted in the current time slot, or **COLISSION** if two or more packets have been transmitted in the current time slot.

In practice, the stations are commonly small, bulk-produced hand-held devices running on batteries. Since batteries' lifetime is not expected to improve much in the foreseeable future, is exceedingly important to save power to allow the stations to run for a longer period of time. Consequently, we are interested in developing protocols that allow stations to *power off* their transceiver to the largest extent possible. Accordingly, we judge the goodness of a protocol by the following two yardsticks: (i) the overall number of time slots required by the protocol to terminate; (ii) for each individual station, the total number of time slots it has to be awake in order to transmit/receive packets. Henceforth, *awake* and *sleep* refer to the station's transceiver activation and deactivation respectively.

The goal of optimizing the above parameters are, of course, conflicting. One can minimize overall completion time at the expense of energy consumption. Conversely, energy can be minimized at the expense of completing time. The challenge is to find a balance between these two conflicting parameters.

The *initialization* problem is to assign to each of the n stations in the RN, an integer ID, in the range $[1, n]$ such that no two stations are assigned the same ID. The initialization problem is fundamental in both network design and multiprocessor systems [2, 3, 5, 7]. Recent advances in wireless communications and mobile computing have exacerbated the need for efficient protocols for wireless networks. Indeed, a large number of such protocols have been reported in the literature [1, 4, 6]. However, many of these protocols function under the assumption that the stations in the wireless networks have been initialized in advance. The highly non-trivial task of assigning distinct ID numbers to the stations (i.e., initializing the stations), is often ignored in the literature. It is, therefore, of importance to design efficient initialization protocols for RNs.

Recently, Nakano and Olariu [7] presented an initialization protocol in which no station needs to awake for more than $O(\log\log n)$ time slots with probability exceeding $1-\frac{1}{n}$. Their protocol terminates in $O(n)$ time slots if every station knows n in advance. It has been open whether or not the RN with n stations can be initialized in $O(\log\log n)$ awake time slots, with high probability, even if no station knows the number n of stations. Our main contribution is to provide the breakthrough. We show that even if the number n of stations is not known beforehand, a single-channel RN can be initialized by our protocol that terminates, with high probability, in $O(n)$ time slots with no station being awake for more than $O(\log\log n)$ time slots. The key idea to initialize the RN with n station in $O(\log\log n)$ awake time slots for unknown n, is to combine two protocols that initializes the RN in $O(\log n)$ awake time slots for unknown n [2] and in $O(\log\log n)$ awake time slots for known n [7]. More precisely, we modify the approximation protocol for n in [2] such that the stations are awake for $O(\log\log n)$ time slots. After that, we use $O(\log\log n)$-awake-time initialization protocol [7] for known n to initialize all the stations.

We also extend our single-channel to the k-channel RN such that the initialization can be performed, with high probability, in $O(\frac{n}{k} + (\log n)^2)$ time slots with no station being awake for more than $O(\log\log n)$ time slots.

2 A Brief Refresher of Basic Probability Theory

This section reviews elementary probability theory results that are useful to analyze the performance of our protocols. For a random variable X, $E[X]$ denotes the expected value of X. Let X be a random variable denoting the number of successes in n independent Bernoulli trials with parameters p and $1-p$. It is well known that X has a *binomial distribution* and that for every r, $(0 \le r \le n)$,

$$\Pr[X = r] \;=\; \begin{pmatrix} n \\ r \end{pmatrix} p^r (1-p)^{n-r}.$$

Further, the expected value of X is given by

$$E[X] \;=\; \sum_{r=0}^{n} r \cdot \Pr[X = r] = np.$$

To analyze the tail of X, we shall make use of the following estimate, commonly referred to as *Chernoff bound*:

$$\Pr[X > (1+\delta)E[X]] < \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{E[X]} \quad (0 \le \delta). \tag{1}$$

We also use the following estimates derived from (1):

$$\Pr[X < (1-\mu)E[X]] < e^{-\frac{\mu^2}{2}E[X]} \;\; (0 \le \mu \le 1). \tag{2}$$

3 Initialization Protocol for Known Lower and Upper Bound of n

If the number n of stations in the RN is known beforehand, Nakano and Olariu [7] presented two initialization protocols for n-station RNs, for both single and k channels. We briefly describe their protocols and give simple explanations in Appendix, for details, we refer the reader to [7]. Their fruitful work encourages us to propose initialization protocols to address the case where the number n of stations is unknown beforehand. The following two Theorems are important basis of our initialization protocols.

Theorem 1 *If the number n of stations is known beforehand, with probability exceeding $1 - \frac{1}{n}$, a single-channel, n-station RN can be initialized in $48n - \frac{n}{4 \ln n} - 1$ time slots with no station being awake for more than $3 \log\log n + 18$ time slots.*

Theorem 2 [1] *If the number n of stations is known beforehand, with probability exceeding $1 - \frac{1}{n}$, a k-channel, n-station RN can be initialized in $O(\frac{n}{k} + \log n)$ time slots, with no station being awake for more than $O(\log\log n)$ time slots, for every $k \geq 1$.*

If n is unknown, we have to design a protocol to find a good approximation of n. We begin by addressing the case where a good approximation N of n satisfies the following double inequality:

$$\frac{N}{c} \leq n \leq N \tag{3}$$

where $c > 1$ is a fixed constant.

By substituting the upper bound N for n in **Step 1** and substituting the lower bound $\frac{N}{c}$ for n in **Step 3**, in the initialization protocols for Theorems 1 and 2 (see Appendix), the following two Corollaries can be obtained:

Corollary 1 *Given an approximation N of n satisfying (3), a single-channel n-station RN can be initialized in $48N - \frac{N}{4c \ln N} - 1$ time slots, with no station being awake for more than $3 \log\log N + 18 + 2 \log c$ time slots with probability exceeding $1 - \frac{1}{N}$.*

Corollary 2 *Given an approximation N of n satisfying (3), a k-channel n-station RN can be initialized in $O(\frac{N}{k} + \log N)$ time slots with no station being awake for more than $O(\log\log N)$ time slots with probability exceeding $1 - \frac{1}{N}$, for every $k \geq 1$.*

Note that each execution of the protocol for Corollary 1 fails with probability of at most $\frac{1}{N}$. If we run the protocol for three times, all the three executions would fail with probability of at most $(\frac{1}{N})^3 = N^{-3}$. Concretely, after one execution of the protocol for Corollary 1, the stations that have not received their IDs transmit, and all the stations monitor the channel. If the channel status is not NULL, there exist at least one station that did not receive its ID, thus we rerun the protocol for Corollary 1. If the channel status is NULL, all the stations have been correctly initialized. ¿From the above discussion, the readers should have no difficulty to confirm the following two lemmas:

Lemma 1 *Given an approximation N of n satisfying (3), a single-channel, n-station RN can be initialized in $3(48N - \frac{N}{4c \ln N} - 1) + 2 < 144N$ time slots, with no station being awake for more than $3(3 \log\log N + 18 + 2 \log c) + 2 = O(\log\log N)$ time slots, with probability exceeding $1 - N^{-3}$.*

Lemma 2 *Given an approximation N of n satisfying (3), a k-channel, n-station RN can be initialized in $O(\frac{N}{k} + \log N)$ time slots, with no station being awake for more than $O(\log\log N)$ time slots, with probability exceeding $1 - N^{-3}$, for every $k \geq 1$.*

4 Finding a Good Approximation of n

At the heart of our energy-efficient initialization protocol for RNs, where the number n of stations is unknown beforehand, lies a simple approximation protocol for n. Specifically, this protocol returns an integer I satisfying, with probability at least $1 - O(n^{-1.88})$, the double inequality:

$$\frac{\log n}{2} - 3 < I \leq \frac{\log n}{2} - 1. \tag{4}$$

Once a good approximation of n is available, we can use the protocol discussed in Section 3 to initialize the entire RN. Our protocol begins with each station generating uniformly at random a real number x in the range $(0, 1]$. Let $g(i)$, ($i \geq 1$), denote the group of stations for which $\frac{1}{2^i} < x \leq \frac{1}{2^{i-1}}$ and let $|g(i)|$ denote the number of stations in $g(i)$. Also, let $G(j)$ represent a collection of all the groups $g(i)$ satisfying $2^{j-1} \leq i \leq 2^j - 1$ for every j ($j \geq 1$). Thus, $G(1) = \{g(1)\}, G(2) = \{g(2), g(3)\}, \ldots, G(j) = \{g(2^{j-1}), g(2^{j-1}+1), \ldots, g(2^j - 1)\}$. Clearly, $|G(j)| = 2^{j-1}$. Then, with $\frac{2^i}{2}$ as the lower bound and $32 \cdot 2^i$ as the upper bound on the number of stations in $g(i)$ we can use the protocol for Lemma 1 and try to initialize each group in $g(i)$ within $144 \times 32 \cdot 2^i = 4608 \cdot 2^i$ time slots in turn. In other words, we try to initialize each group in $G(1), G(2) \ldots$ in turn. After executing initialization on group $g(i)$, we check whether or not the following conditions are satisfied:

- all the stations in $g(i)$ correctly received their IDs;
- $|g(i)|$ is really in between $\frac{2^i}{2}$ and $32 \cdot 2^i$.

We say that $g(i)$ is *successful* if the above two conditions are satisfied. After trying to initializing all the groups in $G(j)$, we check whether or not there exist successful groups in $G(j)$. The above process is repeated until successful groups are encountered. Let $g(I)$ be the first successful group and assume that $g(I)$ belongs to $G(l)$. Then, we use binary search in $G(l)$ to find the group $g(I)$. As we will show latter, I satisfies the inequality (4) with high probability. The details of our approximation protocol are spelled out as follows:

[1] Since the evaluation for k-channel depends on k, we just give the asymptotic bound for Theorem 2, Corollary 2 and Lemma 2.

Protocol `Approximation`
each station generates uniformly at random a real number x in $(0, 1]$;
let $g(i)$, $(i \geq 1)$, denote the group of stations for which $\frac{1}{2^i} < x \leq \frac{1}{2^{i-1}}$;
for $j \leftarrow 1$ **to** ∞ **do**
  for $i \leftarrow 2^{j-1}$ **to** $2^j - 1$ **do**
  /* try to initialize each group $g(i)$ in $G(j)$ and check whether or not $g(i)$ is successful */
    initialize $g(i)$ in $4608 \cdot 2^i$ time slots using the protocol for Lemma 1 with $\frac{2^i}{2}$ and $32 \cdot 2^i$ as the lower and upper bounds, respectively.
    the stations that failed to receive their IDs in $g(i)$ transmit on the channel;
    if the channel status is NULL **and** $g(i)$ contains at least $\frac{2^i}{2}$ and at most $32 \cdot 2^i$ stations $\cdots(*)$
    then the last station of $g(i)$ records its success;
  end for
  the last stations recorded success in groups within $G(j)$ transmit and all the stations in the RN monitor the channel;
  if the channel status is not NULL **then**
    use binary search to find out the first successful group in $G(j)$;
    the last station of the first successful group transmits its group index, and all the stations in the RN receive the group index and denote it as I.
    all the stations in the RN exit the protocol.
end for

After initializing $g(i)$, all the stations in the group that did not receive their IDs transmit on the channel in an additional time slot and the last station (i.e., the station having the largest local ID) in $g(i)$ monitors the channel. Note that the last station has been identified according to Theorem 1. If all the stations in $g(i)$ have received their IDs, then, the channel status of the last station is NULL and the initialization task succeeded in assigning IDs to all the stations in $g(i)$. Since the last station knows the number of stations in $g(i)$, it can judge whether or not the condition $(*)$ is satisfied. Obviously, $g(i)$ is successful if it satisfied the condition $(*)$.

After initializing all the groups in $G(j)$ $(j \geq 1)$, the last station of each successful group $g(i)$ $(2^{j-1} \leq i \leq 2^j - 1)$ in $G(j)$ transmits, and all the stations in the RN monitor the channel. If the channel status is NULL, then there was no successful group within $G(j)$ and the execution is transferred to $G(j+1)$. Otherwise, there exist at least one successful group in $G(j)$.

As assumed before, $g(I)$ belongs to $G(l)$, which implies that $2^{l-1} \leq I \leq 2^l - 1$. Clearly, the **for**-loop of protocol `Approximation` is repeated for $|G(1)| + |G(2)| + \cdots + |G(l)| = 2^l - 1 < 2I$ iterations and all the stations in the RN are awake for $l = \lfloor \log I \rfloor + 1$ time slots to verify whether or not I exists.

Our next task is to find $g(I)$, that is, the first successful group in $G(l)$. To begin with, groups in $G(l)$ are partitioned into two equal-sized subsets, $P_1 = \{g(i) \mid 2^{l-1} \leq i \leq 2^{l-1} + \frac{2^{l-1}}{2} - 1\}$ and $P_2 = \{g(i) \mid 2^{l-1} + \frac{2^{l-1}}{2} \leq i \leq 2^l - 1\}$. Every last station in all the successful groups of P_1 transmits on the channel. If the channel status is either SINGLE or COLLISION, the first successful group in P_1 is found, recursively. Otherwise, the first successful group in P_2 is found, recursively. The reader should have no difficulty to see that this recursive operation is essentially the well-known binary search. Clearly, finding the group $g(I)$ takes $l - 1 = \lfloor \log I \rfloor$ time slots. To put it in a nutshell, the outermost **for**-loop of protocol `Approximation` is repeated for less than $2I$ iterations and according to Lemma 1, initializing $g(i)$ takes $4608 \cdot 2^i$ time slots, thus, the overall number of time slots is less than $\sum_{i=1}^{2I} 4608 \cdot 2^i$ time slots. Additional $2\lfloor \log I \rfloor + 1$ time slots are necessary to make the value I known to all the stations in the RN. Thus, the total number of time slots of protocol `Approximation` is less than

$$\sum_{i=1}^{2I} 4608 \cdot 2^i + 2\log I + 1 = O(2^{2I}). \tag{5}$$

We now evaluate the maximum number of time slots that each individual station has to be awake. Note that each station belongs to exactly one of the groups $g(1), g(2), \ldots$, and therefore, each station in group $g(i)$ is awake for $O(\log\log 2^i) = O(\log I)$ time slots during the initialization. Furthermore, all the stations must monitor the channel to check if I is in the range 2^{j-1} to 2^j. This takes additional $\log I + 1$ time slots. Using binary search to obtain I takes $\log I$ time slots. Altogether, each station needs to be awake for at most $O(\log I)$ time slots.

Our next task is to show that I satisfies condition (4) with high probability. For this purpose we rely on the following technical results.

Lemma 3 *If i satisfies $1 \leq i \leq \frac{\log n}{2} - 3$ then $\Pr[|g(i)| > 32 \cdot 2^i] > 1 - n^{-2.88}$.*

Clearly, $i \leq \frac{\log n}{2} - 3$ implies that

$$2^i \leq \frac{\sqrt{n}}{8} \leq \frac{n}{8\sqrt{n}} \leq \frac{n}{64 \cdot 2^i} \text{ and } 64 \cdot 2^i \leq 8\sqrt{n} \leq \frac{n}{2^i}$$

Since $g(i)$ consists of stations generating a real number x satisfying $\frac{1}{2^i} < x \leq \frac{1}{2^{i-1}}$, the expected number of stations in $g(i)$ is $E[|g(i)|] = \frac{n}{2^i}$. Using the Chernoff bound in (2) with $\mu = \frac{1}{2}$, we can evaluate the probability that group $g(i)$ contains strictly less than $32 \cdot 2^i$ stations as follows:

$$\begin{aligned} &Pr[|g(i)| < 32 \cdot 2^i] \\ \leq\ & \Pr[|g(i)| < (1 - \frac{1}{2})\frac{n}{2^i}] < e^{-(\frac{1}{2})^2 \cdot \frac{1}{2} \cdot \frac{n}{2^i}} \\ \leq\ & e^{-(\frac{1}{2})^2 \cdot 4 \cdot \sqrt{n}} < e^{-(\frac{1}{2})^2 \cdot 8 \cdot \log n} \\ <\ & n^{-2.88} \end{aligned}$$

This completes the proof of Lemma 3. □

Lemma 4 *If* $i = \lfloor \frac{\log n}{2} \rfloor - 1$*, then* $\Pr[|g(i)| < 32 \cdot 2^i] > 1 - n^{-2.22}$.

If $i = \lfloor \frac{\log n}{2} \rfloor - 1$ then clearly,

$$\frac{\log n}{2} - 2 < i \leq \frac{\log n}{2} - 1.$$

This also implies that

$$\frac{\sqrt{n}}{4} < 2^i \leq \frac{\sqrt{n}}{2} \text{ and that}$$
$$16 \cdot 2^i > 4\sqrt{n} > \frac{n}{2^i} \geq 2\sqrt{n} > 4 \cdot 2^i.$$

Using these relations, we can evaluate the probability that group $g(i)$ contains strictly more than $32 \cdot 2^i$ stations as follows:

$$\begin{aligned} &Pr[|g(i)| > 32 \cdot 2^i] \\ <\ & \Pr[|g(i)| > 2\frac{n}{2^i}] < (\frac{e^1}{2^2})^{\frac{n}{2^i}} \\ \leq\ & (\frac{e^1}{2^2})^{2\sqrt{n}} < (\frac{e^1}{2^2})^{4 \log n} \\ <\ & n^{-2.22}. \end{aligned}$$

This completes the proof of Lemma 4 . □

Lemma 5 *If* i *satisfies* $1 \leq i \leq \frac{\log n}{2} - 1$*, then* $\Pr[|g(i)| > \frac{2^i}{2}] > 1 - n^{-2.20}$.

Similarly, $i \leq \frac{\log n}{2} - 1$ implies that

$$2^i \leq \frac{\sqrt{n}}{2} \leq \frac{n}{2\sqrt{n}} \leq \frac{n}{4 \cdot 2^i} \text{ and } 4 \cdot 2^i \leq 2\sqrt{n} \leq \frac{n}{2^i}$$

Using the Chernoff bound in (2) with $\mu = \frac{7}{8}$, we can evaluate the probability that group $g(i)$ contains strictly less than 2^i stations as follows:

$$\begin{aligned} &Pr[|g(i)| < \frac{2^i}{2}] \\ \leq\ & \Pr[|g(i)| < (1 - \frac{7}{8})\frac{n}{2^i}] < e^{-(\frac{7}{8})^2 \cdot \frac{1}{2} \cdot \frac{n}{2^i}} \\ \leq\ & e^{-(\frac{7}{8})^2 \cdot \sqrt{n}} < e^{-(\frac{7}{8})^2 \cdot 2 \log n} \\ <\ & n^{-2.20}. \end{aligned}$$

This completes the proof of Lemma 5. □

Lemma 3 and Lemma 4 combined assure that when $\frac{\log n}{2} - 3 < i < \frac{\log n}{2} - 1$, with probability exceeding $1 - n^{-2.88} - n^{-2.22}$, $|g(i)| < 32 \cdot 2^i$. Lemma 5 assures that when $i < \frac{\log n}{2} - 1$, with probability exceeding $1 - n^{-2.20}$, $|g(i)| > \frac{2^i}{2}$. Thus, we have proved the following result:

Lemma 6 *When the first successful group* $g(I)$ *is verified, condition (4) holds with probability of at least* $1 - O(n^{-2.20})$.

According to equation (5), the total number of time slots of protocol `Approximation` is at most $O(2^{2I}) = O(2^{2(\frac{\log n}{2} - 1)}) = O(n)$ time slots and each station is awake for no more than $O(\log I) = O(\log \log n)$ time slots. From Lemma 6, when the number of stations in $g(i)$ reduces to around $\sqrt{n}$, condition $(*)$ of protocol `Approximation` will be satisfied for the first time with high probability. According to Lemma 1, we can assure that group $g(I)$ is correctly initialized with probability exceeding $1 - \frac{1}{(\sqrt{n})^3} = 1 - n^{-1.5}$. Thus, we have proved the following important result:

Theorem 3 *Protocol* `Approximation` *returns an integer* I *satisfying condition (4) with probability of at least* $1 - O(n^{-1.5})$*. Moreover, the protocol terminates in* $O(n)$ *time slots with no station being awake for more than* $O(\log \log n)$ *time slots.*

5 An Energy-efficient Initialization Protocol for Unknown n

According to Theorem 3, protocol `Approximation` returns an integer I that satisfies condition (4) with probability of at least $1 - O(n^{-1.5})$. Note that condition (4) also implies that

$$4 \cdot 2^{2I} \leq n < 64 \cdot 2^{2I} \leq 16n. \tag{6}$$

Thus, we can initialize the whole RN using $4 \cdot 2^{2I}$ and $64 \cdot 2^{2I}$ as the lower and upper bounds of n. According to Lemma 1, with probability exceeding $1 - (64 \cdot 2^{2I})^{-3} > 1 - n^{-3}$, the initialization can be completed in $144 \times 64 \cdot 2^{2I} < 144 \times 16n = 2304n = O(n)$ time slots with no station being awake for more than $O(\log \log (64 \cdot 2^{2I})) = O(\log \log n)$ time slots. By combining the above result and Theorem 3, we obtain:

Theorem 4 *Even if no station knows the number* n *of stations beforehand, an* n*-station RN can be initialized in* $O(n)$ *time slots, with no station being awake for more than* $O(\log \log n)$ *time slots, with probability exceeding* $1 - O(n^{-1.5})$.

6 An Energy-efficient Initialization Protocol for the k-channel and Unknown n

In this section we extend the initialization protocol presented in Section 5 to the multichannel case. Consider a RN where k channels are available. By Lemma 2, it is straightforward to see that initializing $g(i)$ in protocol `Approximation`, when ported to k channels, terminates in $O(\frac{2^i}{k} + \log 2^i)$ time slots with no station being awake for more than $O(\log i)$ time slots. The **for** loop in protocol `Approximation` takes altogether

$$\sum_{i=1}^{2I} O(\frac{2^i}{k} + \log 2^i) = \sum_{i=1}^{\log n - 2} O(\frac{2^i}{k} + i) = O(\frac{n}{k} + (\log n)^2),$$

time slots, and no station needs to be awake for more than $O(\log \log n)$ time slots. As performed in the single-channel case, we can use channel 1 to check if I is in the range 2^{j-1} to 2^j and then run binary search to find I, which takes, altogether, $O(\log \log n)$ time slots.

After getting a good approximation, we are now in position to initialize the whole RN using the same lower and upper bounds of equation (6). By Lemma 2, initializing the whole RN can be completed in $O(\frac{n}{k} + \log n)$ time slots with no station being awake for more than $O(\log \log n)$ time slots when k channels are available. Thus, we have the following result:

Theorem 5 *Even if no station knows the number n of stations beforehand, a k-channel, n-station RN can be initialized in $O(\frac{n}{k} + (\log n)^2)$ time slots, with no station being awake for more than $O(\log \log n)$ time slots with probability exceeding $1 - O(n^{-1.5})$, for every $k \geq 1$.*

7 Conclusion

In this work we have presented energy-efficient initialization protocols for RNs. We have shown that even if the number n of stations is not known beforehand, a single-channel RN with collision detection capability can be initialized by our protocol that terminates, with high probability, in $O(n)$ time slots with no station being awake for more than $O(\log \log n)$ time slots. We then presented an initialization protocol for the k-channel RN that terminates, with high probability, in $O(\frac{n}{k} + (\log n)^2)$ time slots with no station being awake for more than $O(\log \log n)$ time slots.

References

[1] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, "Single-round simulation on radio networks," J. Algorithms, vol.13, no.2 pp.188–210, Jun.1992.

[2] J. L. Bordim, J. Cui, T. Hayashi, K. Nakano, and S. Olariu, "Energy-efficient initialization protocols for ad-hoc radio network," Proc. International Symposium on Algorithms and Computation, pp.215-224,Channel, India, LNCS 1741, Springer-Verlag, 1999.

[3] T. Hayashi, K. Nakano, S. Olariu, "Randomized Initialization Protocols for Packet Radio Networks," Proc. IEEE 13th International Parallel Processing Symposium, pp.544–548, Apr.1999.

[4] K. Nakano, "Optimal sorting algorithms on bus-connected processor arrays," IEICE Trans. Fundamentals, vol.E-76A, no.11, pp.2008–2015, Nov.1994.

[5] K. Nakano, "Optimal initializing algorithms for a reconfigurable mesh," J. Parallel and Distributed Computing, vol.24, pp.218–223, Jan.1995.

[6] K. Nakano, S. Olariu, and J. L. Schwing, "Broadcast-efficient protocols for mobile radio network," IEEE Trans. parallel and Distributed Systems, vol.10, no.12, pp.1276-1289, Dec.1999.

[7] K. Nakano and S. Olariu, "Energy-Efficient Initialization Protocols for Single-Hop Radio Networks with no Collision Detection," IEEE Trans. parallel and Distributed Systems, vol.11, no.8, pp.851-1863, Aug.2000.

[8] K. Nakano and S. Olariu, "Energy-Efficient Randomized Routing in Radio Networks," Proc. of 4th Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM), pp.35–44, Aug. 2000.

Appendix

This appendix briefly reviews the initialization protocols for both single and k channels proposed by Nakano and Olariu in [7], which result in Theorems 1 and 2. In what follows, we will show how to modify the initialization protocols for Theorems 1 and 2 to obtain Corollaries 1 and 2. For later reference, we list the following three lemmas, which are important ingredients of the initialization protocols. For further details, see [7].

Lemma 7 *The prefix-sums of a sequence of n numbers can be solved in $n - 1$ time slots on a single-channel RN with no station being awake for more than 2 time slots.*

Lemma 8 *The prefix-sums of a sequence of n numbers can be computed on a k-channel, n-station RN in $\frac{n}{k} + O(\log n)$ time slots, with no station being awake for more than 8 time slots, for every $k \geq 1$.*

Lemma 9 *A single-channel, n-station RN, where each stations has an unique ID in the range $[1..m]$ ($m \geq n$) can be initialized in $2m-2$ time slots with no station being awake for more than $2\log m$ time slots and the station holding the largest number in $[1..m]$ is aware that it is the last station with the largest ID number after initialization.*

The Proof of Theorem 1

In outline, the Nakano and Olariu's initialization protocol can be roughly divided into three steps: **Step 1** assigns temporary IDs in the range 1 to $16n$ to the n stations. In **Step 2**, the temporary IDs are redistributed uniformly over the range 1 to $16n$. In **Step 3**, the n stations are partitioned into groups and each group is initialized individually. Finally, by computing the prefix-sums, each station gets its unique ID in the range 1 to n.

We now give a more detailed explanation of each step. In **Step 1**, the protocol `Assign-TemporaryID -toAll` assigns temporary IDs in the range $[1..16n]$ to all the stations in the RN. The details of the protocol are spelled out as follows:

Protocol `Assign-TemporaryID-toAll`(n)
$s \leftarrow 0$;
for $i \leftarrow 1$ **to** $\log\log n + 3$ **do**
  each station that has not received its temporary ID
   picks an integer uniformly at random in the range
   1 to $\frac{n}{2^{i-4}}$ and then goes to sleep;
  for $j \leftarrow 1$ **to** $\frac{n}{2^{i-4}}$ **do**
    stations that have picked integer j wake up and
     transmit on the channel;
    if the channel status is SINGLE **then**
      the unique station that has transmitted records
       the number $s+j$ as its temporary ID.
  endfor
  $s \leftarrow s + \frac{n}{2^{i-4}}$;
endfor

It was proved that with probability exceeding $1-\frac{1}{n}$, $\log\log n + 3$ iterations of the outermost **for** loop is enough to successfully assign temporary IDs in the range 1 to $16n$ to all the stations. In their protocol, an important result was pointed out that after the first iteration of the outermost **for** loop, at least $\frac{3n}{4}$ stations have been uniformly assigned temporary IDs in the range 1 to $8n$.

Step 2 is devoted to shuffling the temporary IDs so as to redistribute them uniformly to the range 1 to $16n$. Concretely, let: **Group A:** denote all the stations whose temporary IDs are between 1 to $8n$, and **Group B:** denote all the stations with temporary IDs between $8n+1$ to $16n$. Shuffle the temporary IDs in the whole range 1 to $16n$ by using the following trick: (i) for $1 \leq i \leq 8n, i \rightarrow 2i+1$, and (ii) for $8n+1 \leq i \leq 16n, i \rightarrow 2i-16n$

In the other words, after the shuffling, the original IDs of the stations in group A become *odd* numbers spread over the range 1 to $16n$, while the original IDs of stations in group B become *even* numbers spread over the whole range 1 to $16n$. Since the original IDs 1 to $8n$ are picked up uniformly, the shuffling numbers are still uniformly distributed over the range 1 to $16n$.

In **Step 3**, the stations with temporary IDs in the range 1 to $16n$ are partitioned into $\frac{n}{4\ln n}$ groups $G'(1), G'(2), \ldots, G'(\frac{n}{4\ln n})$ where each of them has $64\ln n$ consecutive integers. It was proved that any group $G'(i)$ contains, with high probability, at least one station. According to Lemma 9, the stations in each group $G'(i)$, $(1 \leq i \leq \frac{n}{4\ln n})$ can be initialized individually in at most $128\ln n - 2$ time slots with no station being awake for more than $2\log(64\ln n) = 2\log\ln n + 12 < 2\log\log n + 12$. The overall number of time slots is $\frac{n}{4\ln n}\cdot(128\ln n - 2) = 32n - \frac{n}{2\ln n}$. At this point, each station in group $G'(i)$, $(1 \leq i \leq \frac{n}{4\ln n})$, has a local ID within $G'(i)$. Finally, executing the prefix-sums on $|G'(1)|, |G'(2)|, \ldots, |G'(\frac{n}{4\ln n})|$ can reassign global IDs in the range 1 to n to all the stations. Since, with high probability, none of the groups is empty, we can use the first station in each group $G'(i)$ for this computation. By using Lemma 7, this task terminates in $\frac{n}{4\ln n} - 1$ time slots with no station being awake for more than 2 time slots. After that, each station updates its local ID in the obvious way to obtain the desired final ID. Note that the last station in group $G'(\frac{n}{4\ln n})$ is aware that it is the last station in the entire RN and holds the largest ID n. Summarizing the analysis of **Step 3**, we initialize all the stations in $32n - \frac{n}{2\ln n} + \frac{n}{4\ln n} - 1 = 32n - \frac{n}{4\ln n} - 1$ time slots with no station being awake for more than $2\log\log n + 15$ time slots. Recall that protocol `Assign-TemporaryID-toAll`(n) takes $16n$ time slots with no station being awake for more than $\log\log n+3$ time slots. Thus, Theorem 1 have been proved.□

The Proof of Theorem 2

We are now in position to extend the initialization protocol for single channel, aforementioned, to the case where k channels are available.

It is not difficult to see that the i^{th} iteration of the outermost **for** loop, when ported to k channels, terminates in $\lceil\frac{n}{2^{i-4}k}\rceil$ time slots and each station is awake for only one time slot. The **for** loop in protocol `Assign-TemporaryID-toAll`(n) takes altogether

$$\sum_{i=1}^{\log\log n+3} \lceil\frac{n}{2^{i-4}k}\rceil \quad < \quad \sum_{i=1}^{\log\log n+3} (\frac{n}{2^{i-4}k}+1)$$

$$< \quad \frac{16n}{k} + \log\log n + 3$$

time slots, and no station needs to be awake for more than $\log\log n + 3$ time slots.

Once all the stations have received their temporary IDs, in the range 1 to $16n$, and have been shuffled, all the stations are divided into $\frac{n}{4\ln n}$ groups $G'(1), G'(2), \ldots, G'(\frac{n}{4\ln n})$. Using one channel, the stations in each group can be initialized in $128\ln n - 2$ time slots with no station being awake for more than $2\log\log n + 12$ time slots. If $k \leq \frac{n}{4\ln n}$, by assigning the $\frac{n}{4\ln n}$ groups to the k channels, and initializing all the $\frac{n}{4\ln n}$ groups takes $\frac{n}{4\ln n\cdot k}(128\ln n - 2) = \frac{32n}{k} - \frac{n}{2\ln n\cdot k}$ time slots. If $k > \frac{n}{4\ln n}$, then we use $\frac{n}{4\ln n}$ channels, assign one channel to each group, and initialize every group simultaneously. In such a case, initializing all the groups takes $128\ln n - 2 = O(\log n)$ time slots. By Lemma 8, computing the prefix sums of $|G'(1)|, |G'(2)|, \ldots, |G'(\frac{n}{4\ln n\cdot k})|$ takes $\frac{n}{4\ln n\cdot k} + O(\log\frac{n}{4\ln n})$ time slots with no station being awake for more than 8 time slots, for every $k \geq 1$. Theorem 2 summarizes the above analysis. □

The Proof of Corollary 1

Assume that the exact number n of stations in the RN is unknown, and that a lower and upper bound of n, denoted as $\frac{N}{c}$ and N, are available. Then, by substituting the upper bound N for n in **Step 1** and substituting the lower bound $\frac{N}{c}$ for n in **Step 3** (in the initialization protocols for Theorem 1), the RN can still be initialized energy-efficiently. More specifically, we use the upper bound N for n in **Step 1** and run the protocol `Assign-TemporaryID-toAll` to assign temporary IDs in the range 1 to $16N$ to all the stations in the RN. This task terminates, with probability exceeding $1 - \frac{1}{N}$, in $O(N)$ time slots with no station being awake for more than $O(\log\log n)$ time slots. **Step 2** shuffles the temporary IDs so as to redistribute temporary IDs uniformly over the range 1 to $16N$. In **Step 3**, we partition all the stations with temporary IDs in the range 1 to $16N$ into $\frac{N}{4c\ln N}$ group $G'(1), G'(2), \ldots G'(\frac{N}{4c\ln N})$ of $64c\ln N$ consecutive integers each, such that each group $G'(i)$ involves all the stations with temporary IDs from $64c\ln N\cdot i + 1$ to $64c\ln N\cdot(i+1)$. After the first iteration of the outermost **for** loop, with probability of at least $1 - \frac{1}{N}$, at least $\frac{3n}{4}$ stations have been uniformly assigned temporary IDs over the range 1 to $8N$. The shuffling, uniformly redistributed the stations over the range 1 to $16N$. Consequently, for an arbitrary fixed odd number $2i - 1$, $(1 \leq i \leq 8N)$, the probability that a station has received its temporary ID $2i - 1$ is at least $\frac{\frac{3n}{4}}{8N} > \frac{\frac{3\cdot N}{4c}}{8N} = \frac{3}{32c}$ (note that here we substitute $\frac{N}{c}$ for n). Group $G'(i)$, $(1 \leq i \leq \frac{n}{4c\ln N})$, consists of $64c\ln N$ consecutive integers and has $32c\ln N$ odd numbers. Since $(1-\frac{1}{x})^x < e^{-1}$ for every $x > 1$, the probability that at least one station is contained in any group is $(1 - \frac{3}{32c})^{32c\ln N} < e^{-3\ln N} = N^{-3}$. Hence, with high probability exceeding $1 - \frac{1}{4c\ln N\cdot N^2}$, each of the $\frac{N}{4c\ln N}$ groups has at least one station.

We can then initialize the stations in each group $G'(i), (1 \leq i \leq \frac{N}{4c\ln N})$, individually. According to Lemma 9, this task terminates in at most $128c\ln N - 2$ time slots with no station being awake for more than $2\log(64c\ln N) = 2\log\ln N + 12 + 2\log c < 2\log\log N + 12 + 2\log c$. The overall number of time slots is $\frac{N}{4c\ln N}\cdot(128c\ln N - 2) = 32N - \frac{N}{2c\ln N}$. At this point, each station in $G'(i), (1 \leq i \leq \frac{N}{4c\ln N})$ has received a local ID. Finally, the prefix-sums of $|G'(1)|, |G'(2)|, \ldots, |G'(\frac{N}{4c\ln N})|$ are computed. Clearly, we can use the first station in each group $G'(i)$ for this computation since, with probability exceeding $1 - \frac{1}{4c\ln N\cdot N^2}$, none of the groups is empty. By Lemma 7, this task takes $\frac{N}{4c\ln N} - 1$ time slots with no station being awake for more than 2 time slots. After that, each station updates its local ID, in the obvious way, to obtain its final ID. From the above discussion, we have proved the results in Corollary 1. □

The Proof of Corollary 2

In this subsection we extend the initialization protocol for Corollary 1 to the k-channel RN. In **Step 1**, the outermost **for** loop in protocol `Assign-TemporaryID-toAll`(N) takes altogether

$$\sum_{i=1}^{\log\log n+3} \lceil \frac{N}{2^{i-4}k} \rceil \quad < \quad \sum_{i=1}^{\log\log n+3} (\frac{N}{2^{i-4}k} + 1)$$
$$< \quad \frac{16N}{k} + \log\log n + 3$$

time slots, and no station needs to be awake for more than $\log\log n + 3$ time slots. After the shuffling, all the stations are divided into $\frac{N}{4c\ln N}$ groups $G'(1), G'(2), \ldots, G'(\frac{N}{4c\ln N})$. Using one channel, the stations in each group can be initialized in $128c\ln N - 2$ time slots with no station being awake for more than $2\log\log N + 12 + 2\log c$ time slots. If $k \leq \frac{N}{4c\ln N}$, we assign $\frac{N}{4c\ln N}$ groups to the k channels. Thus, initializing all the $\frac{N}{4c\ln N}$ groups takes $\frac{N}{4c\ln N\cdot k}(128c\ln N - 2) = \frac{32N}{k} - \frac{N}{2c\ln N\cdot k}$ time slots. If $k > \frac{N}{4c\ln N}$, then assign one channel to each group and initialize all groups simultaneously. In such a case, initializing all the groups takes $128c\ln N - 2 = O(\log N)$ time slots. By Lemma 8, computing the prefix-sums of $|G'(1)|, |G'(2)|, \ldots, |G'(\frac{N}{4c\ln N\cdot k})|$ takes $\frac{N}{4c\ln N\cdot k} + O(\frac{N}{4c\ln N\cdot k})$ time slots with no station being awake for more than 8 time slots, for every $k \geq 1$. From the above discussion, we have just proved the results in Corollary 2. □