Randomized Initialization Protocols for Ad Hoc Networks

Koji Nakano, *Member*, *IEEE Computer Society*, and Stephan Olariu, *Member*, *IEEE Computer Society*

Abstract—Ad hoc networks are self-organizing entities that are deployed on demand in support of various events including collaborative computing, multimedia classroom, disaster-relief, search-and-rescue, interactive mission planning, and law enforcement operations. One of the fundamental tasks that have to be addressed when setting up an ad hoc network (AHN, for short) is *initialization*. This involves assigning each of the n stations in the AHN a distinct ID number (e.g., a local IP address) in the range from 1 to n. Our main contribution is to propose efficient randomized initialization protocols for AHNs. We begin by showing that if the number n of stations is known beforehand, an n-station, single-channel AHN can be initialized with probability exceeding $1-\frac{1}{n}$, in $en+O(\sqrt{n\log n})$ time slots, regardless of whether the AHN has collision detection capability. We then go on to show that even if n is not known in advance, an n-station, single-channel AHN with collision detection can be initialized with probability exceeding $1-\frac{1}{n}$, in $\frac{10n}{3}+O(\sqrt{n\ln n})$ time slots. Using this protocol as a stepping stone, we then present an initialization protocol for the n-station, k-channel AHN with collision detection that terminates with probability exceeding $1-\frac{1}{n}$, in $\frac{10n}{3k}+O\Big(\sqrt{\frac{n\ln n}{k}}\Big)$ time slots. Finally, we look at the case where the collision detection capability is not present. Our first result in this direction is to show that the task of electing a leader in an n-station, single-channel AHN can be completed with probability exceeding $1-\frac{1}{n}$, in fewer than $11.37(\log n)^2+2.39\log n$ time slots. This leader election protocol allows us to design an initialization protocol for the n-station, single-channel AHN with no collision detection that terminates with probability exceeding $1-\frac{1}{n}$, in fewer than $5.67n+O(\sqrt{n\ln n})$ time slots, even if n is not known beforehand. We then discuss an initialization protocol for the n-station, k-channel AHN with no collision detection that terminates with probability exceeding $1-\frac{1}{n}$, in fewer than $5.67\frac{n}{k}+O\Big(\sqrt{\frac{n\ln n}{k}}\Big)$ time slots, whenever $k\le\frac{n}{(\log n)^3}$.

Index Terms—Ad hoc networks, hierarchical networks, clustering, leader election, collaborative computing, initialization.

1 Introduction

In recent years, wireless and mobile communications have seen an explosive growth both in terms of the number of services provided and the types of technologies that have become available. Indeed, cellular telephony, radio paging, cellular data, and even rudimentary cellular multimedia services have become commonplace and the demand for enhanced capabilities will continue to grow into the foreseeable future [6], [10], [13], [18], [20]. It is anticipated that in the not-so-distant future, mobile users will be able to access their data and other services such as electronic mail, video telephony, stock market news, map services, and electronic banking while on the move [13], [19], [20], [22].

Unlike the well-studied cellular systems that assume the existence of a robust infrastructure, ad hoc networks must be rapidly deployable, possibly multihop, self-organizing, and capable of multimedia service support. Ad hoc networks are well-suited to the needs specific to collaborative computing, multimedia classroom, disaster-relief, search-and-rescue, law enforcement, distance learning, and other special-purpose applications [17], [21], [26], [27], [33], [34].

- *K. Nakano is with the Department of Electrical and Computer Engineering, Nagoya Institute of Technology, Show-ka, Nagoya, 466, Japan. E-mail: nakano@elcom.nitech.ac.jp.*
- *S. Olariu is with the Department of Computer Science, Old Dominion University, Norfolk, VA 23259. E-mail: olariu@cs.odu.edu.*

Manuscript received 7 July 1999; revised 28 Dec. 1999; accepted 23 March 2000.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 110184.

1.1 Ad Hoc Networks

An ad hoc network (AHN, for short) is a distributed system with no central arbiter, consisting of n mobile radio transceivers, henceforth referred to as *stations*. Typically, these stations are small, inexpensive, bulk-produced, commodity devices running on batteries that are deployed on demand in support of various events including collaborative computing, interactive mission planning, disaster-relief, search-and-rescue, and law enforcement operations. Since the AHN is deployed on demand, in places that may not have an existing infrastructure, they must be able to self-organize. Since they are bulk-produced, we assume that it is impossible or impractical to distinguish individual stations by serial and/or manufacturing number.

Scalability and energy-efficiency concerns suggest a hierarchical organization of AHN systems with the lowest level in the hierarchy being a *cluster*. As argued in [21], [23], [25], [26], [35], [37], [38], [40], in addition to helping with scalability and robustness, aggregating stations into clusters and, further, into super-clusters has the additional benefits of 1) conserving battery power, 2) promoting spatial code reuse, and 3) concealing the details of global network topology from individual stations.

In most practical situations, the stations of a cluster are located in close physical proximity—the diameter of a typical cluster rarely exceeds a few hundred meters. An important side-benefit of clustering is that in a cluster every

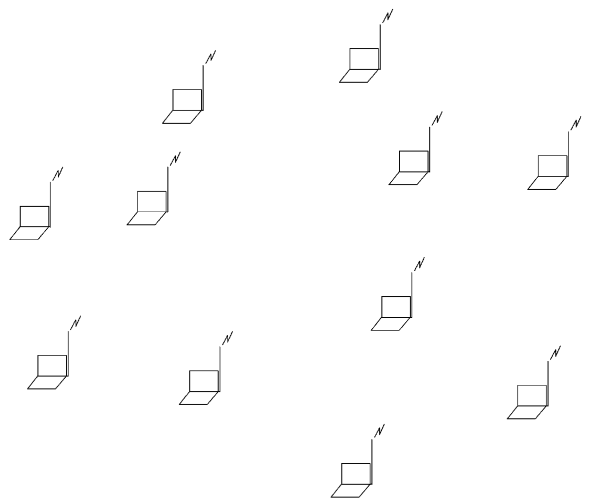

Fig. 1. Illustrating an eleven-station cluster.

station is within transmission range of every other station, implementing, essentially, a local area ad hoc network [8], [38]. Thus, all the stations in a cluster can use the same transmitter-oriented CDMA code, avoiding inter–cluster collisions [21], [26]. We refer the reader to Fig. 1 illustrating an eleven-station cluster.

As it turns out, clusters are key ingredients in handling routing in hierarchically organized multihop AHNs [8], [10], [21], [25], [26], [38]. In such an environment, routing of messages proceeds in stages: first routing takes place locally within a cluster; next, repeaters or *gateways* are being used to route the message from one cluster to the next. This is then repeated until, eventually, the message is delivered to the intended destination [21], [25], [26].

Our focus in this paper is on quasistatic environments, such as multimedia classrooms, or collaborative environments where topology changes are infrequent, reorganizations are not required, and the stations are in relative close proximity. A typical example is a group of people, equipped with laptops, in a conference room wishing to organize themselves into a wireless network, without resorting to a centralized infrastructure.

1.2 The Communication Model

As customary, time is assumed slotted and all transmissions take place at slot boundaries [8], [10]. We assume that the stations have a local clock that keeps synchronous time by interfacing with a Global Positioning System (GPS, for short) [16], [24], [35], [36]. It is worth noting that, under current technology, the commercially-available GPS systems provide location information accurate to within 22 meters, as well as time information accurate[1] to within 100 nanoseconds [16]. In particular, this allows the stations to detect time slot boundaries and, thus, to synchronize even if located indoors [39].

The stations are assumed to have the computing power of a usual laptop computer; in particular, they all execute the same protocol and can generate random bits that provide local data on which the stations may perform computations. We assume that the random numbers generated by each station by combining their random bits are totally independent.

The stations communicate using k radio frequency channels. We assume that in any time slot, a station can tune to one such channel and/or transmit on at most one, possibly the same, channel. A transmission involves a data packet whose length is such that the transmission can be completed within one time slot. Importantly, we assume an ideal propagation of radio transmissions. This is partly justified by the fact that within a cluster there is typically line-of-sight communication between stations.

We employ the commonly-accepted assumption that when two or more stations are transmitting on a channel in the same time slot, the corresponding packets *collide* and are garbled beyond recognition [10], [11], [26], [27]. We distinguish between AHN systems based on their capability to detect collisions. Specifically, in the AHN with *collision detection* (CD, for short), at the end of a time slot the status of a radio channel is:

- NULL—ambient noise, i.e., no transmission on the channel
- SINGLE—exactly one transmission on the channel
- COLLISION—two or more transmissions on the channel.

In the AHN with *no collision detection*(no-CD, for short), at the end of a time slot the status of a radio channel is:

- NOISE—ambient noise: either no transmission or collision on the channel
- SINGLE—exactly one transmission on the channel.

In other words, the AHN with no-CD cannot distinguish between ambient noise and two or more transmissions colliding on the channel.

Since a typical AHN is populated with inexpensive, commodity hand-held stations, the no-CD assumption makes a lot of sense since collision detection requires specialized hardware that may not be cost and energy-effective to provide. Moreover, several workers have argued that, even if the collision detection capability is present, it may be of limited value especially in the presence of noisy channels [8], [9]. However, a number of radio and cellular networks including AMPS, GSM, ALOHA-net, as well as the well-known Ethernet are known to rely on their sophisticated collision detection capabilities [1], [2], [10], [12], [13], [30].

1.3 Our Contributions

There are two fundamental tasks that have to be completed by a self-organizing AHN: leader election and initialization, that we discuss next.

- The classical *leader election* problem asks to designate one of the stations of the AHN as the *leader*. In other words, after performing the leader election protocol, exactly one station learns that it was elected the

1. It is well-documented that GPS systems using military codes achieve a level of accuracy that is orders of magnitude better than their commercial counterparts [16], [24].

leader, while the remaining stations learn the identity of the leader elected. The leader election problem is fundamental, for many other problems rely directly or indirectly on the presence of a leader in a network [4], [5], [28], [41], [42], [43].

- The *initialization* problem is to assign to each of the n stations in the AHN a distinct ID number in the range from 1 to n. The initialization problem is fundamental in both network design and in multiprocessor systems [4], [5], [14], [15], [28], [32].

The main contribution of this work is to propose efficient randomized initialization protocols for AHN. We begin by showing that if the number n of stations is known beforehand, the single-channel AHN can be initialized with probability exceeding $1-\frac{1}{n}$, in $en + O(\sqrt{n\log n})$ time slots, regardless of whether the AHN has collision detection capability.

We then go on to show that, even if n is not known in advance, an n-station, single-channel AHN with collision detection can be initialized with probability exceeding $1-\frac{1}{n}$, in $\frac{10n}{3} + O(\sqrt{n\ln n})$ time slots. Using this protocol as a stepping stone, we then present an initialization protocol for the n-station, k-channel AHN with collision detection that terminates with probability exceeding $1-\frac{1}{n}$, in $\frac{10n}{3k} + O\left(\sqrt{\frac{n\ln n}{k}}\right)$ time slots.

Finally, we address the tasks of leader election and initialization in AHNs where the collision detection capability is not present. Our first result in this direction is to show that the task of electing a leader in an n-station, single-channel AHN can be completed with probability exceeding $1-\frac{1}{n}$, in fewer than $11.37(\log n)^2 + 2.39\log n$ time slots.

In turn, this leader election protocol allows us to design an initialization protocol for the n-station, single-channel AHN with no collision detection that terminates with probability exceeding $1-\frac{1}{n}$ in fewer than $5.67n + O(\sqrt{n\ln n})$ time slots, even if n is not known beforehand. We then discuss an initialization protocol for the n-station, k-channel AHN with no collision detection that terminates with probability exceeding $1-\frac{1}{n}$ in fewer than $5.67\frac{n}{k} + O\left(\sqrt{\frac{n\ln n}{k}}\right)$ time slots, provided that $k \le \frac{n}{(\log n)^3}$.

The remainder of this work is organized as follows: Section 2 offers a brief refresher of basic probability theory results that will be useful in assessing the performance of our protocols. Section 3 discusses the initialization problem for known n. Section 4 discusses the more realistic case where the number of stations in the AHN is not known beforehand. In Section 5, we address the problems of leader election and initialization in the AHN with no-CD in case the number n of stations in not known beforehand. Finally, Section 6 offers concluding remarks and points out directions for further investigations.

2 A Brief Refresher of Probability Theory

The main goal of this section is to review elementary probability theory results that are useful for analyzing the performance of our protocols. For a more detailed discussion of background material, we refer the reader to [31].

Throughout, $\Pr[A]$ will denote the probability of event A. Let $E_1, E_2, \ldots, E_m$ be arbitrary events over a sample space. The well-known De Morgan law states that

$$\overline{\bigcap_{i=1}^{m} E_i} = \bigcup_{i=1}^{m} \overline{E_i}, \tag{1}$$

where $\overline{E_i}$ is the event that occurs if and only if E_i does not. In addition, it is known that

$$\Pr\left[\bigcup_{i=1}^{m} E_i\right] \le \sum_{i=1}^{m} \Pr[E_i], \tag{2}$$

with equality holding only if the events are independent.

For example, assume that the stations of an AHN have been partitioned into k groups $G_1, G_2, \ldots, G_k$ and that for a given i, $(1 \le i \le k)$, the probability of the event $\overline{E_i}$ that group G_i fails to satisfy a predicate $\mathcal{P}$ is p_i. Now, (1) and (2) combined guarantee that the probability of the event that *all* the groups satisfy predicate $\mathcal{P}$ is:

$$\begin{aligned} \Pr[E_1 \cap E_2 \cap \cdots \cap E_k] &= 1 - \Pr[\overline{E_1 \cap E_2 \cap \cdots \cap E_k}] \\ &= 1 - \Pr[\overline{E_1} \cup \overline{E_2} \cup \cdots \cup \overline{E_k}] \\ &\ge 1 - \sum_{i=1}^{m} p_i. \end{aligned} \tag{3}$$

Notice that (3) holds regardless of whether the events $\overline{E_i}$ are independent.

For a random variable X, $E[X]$ denotes the expected value of X. Let X be the random variable denoting the number of successes in n independent Bernoulli trials with parameter p. It is well known that X has a *binomial distribution*, and that for every r, $(0 \le r \le n)$,

$$\Pr[X = r] = \binom{n}{r} p^r (1-p)^{n-r}. \tag{4}$$

Further, the expected value of X is given by

$$E[X] = \sum_{r=0}^{n} r \cdot \Pr[X = r] = np. \tag{5}$$

To analyze the tail of the binomial distribution, we shall make use of the following estimates, commonly referred to as *Chernoff bounds* [31]:

$$\Pr[X < (1-\epsilon)E[X]] < e^{\frac{-\epsilon^2}{2}E[X]} \quad (0 \le \epsilon \le 1) \tag{6}$$

$$\Pr[X > (1+\epsilon)E[X]] < e^{\frac{-\epsilon^2}{3}E[X]} \quad (0 \le \epsilon \le 1). \tag{7}$$

Let X be the random variable denoting the number of successes in a number $f(n) > n$ of independent Bernoulli trials, each succeeding with probability p. Clearly, $E[X] = p \cdot f(n)$. Our goal is to determine the values of ϵ and $f(n)$ in such a way that (6) yields:

$$\Pr[X < n] = \Pr[X < (1-\epsilon)E[X]] < e^{\frac{-\epsilon^2}{2}E[X]} = \frac{1}{n}. \quad (8)$$

It is easy to verify that (8) holds whenever[2]

$$\begin{cases} (1-\epsilon)E[X] = n \\ \frac{\epsilon^2}{2}E[X] = \ln n \end{cases} \quad (9)$$

hold true. Solving for ϵ and $E[X]$ in (9), we obtain:

$$0 < \epsilon = \frac{2}{1 + \sqrt{\frac{1+2n}{\ln n}}} < 1 \quad (10)$$

and

$$E[X] = n + \ln n + \sqrt{1 + \frac{2n}{\ln n}} \ln n = n + O(\sqrt{n \ln n}). \quad (11)$$

Equation (11) allows the desired determination of $f(n)$. We note here that (11) and several of its easy variants will be used repeatedly in the remainder of this work to bound the tail of various binomial distributions.

Finally, we take note of two well-known results that will be used frequently in the remainder of this work.

Lemma 2.1. *For every positive constant $(o < c < n)$, the sequence $\left(1 - \frac{c}{n}\right)^n$ is monotonically increasing and*

$$\lim_{n\to\infty} \left(1 - \frac{c}{n}\right)^n = \frac{1}{e^c}. \quad (12)$$

Lemma 2.2. *For every positive constant $(o < c < n)$, the sequence $\left(1 - \frac{c}{n}\right)^{n-1}$ is monotonically decreasing and*

$$\lim_{n\to\infty} \left(1 - \frac{c}{n}\right)^{n-1} = \frac{1}{e^c}. \quad (13)$$

3 Initializing an n-Station, Single-Channel AHN for Known n

Consider an n-station, single-channel AHN where the number n of stations is known in advance. The idea of our protocol is quite simple and intuitive. To begin, each station transmits on the channel with probability $\frac{1}{n}$ until the status of the channel is SINGLE. The station that has transmitted last receives the ID of 1 and leaves the protocol. At this time, the remaining stations transmit on the channel with probability $\frac{1}{n-1}$ until the status of the channel is SINGLE. Again, the station that transmitted in the last time slot receives the ID of 2 and leaves the protocol. This is then continued until all the stations have received their IDs. The details are spelled out as follows:

Protocol `Initialization-for-known-n`
for $m \leftarrow n$ **downto** 1 do
  repeat
    each station transmits on the channel with probability $\frac{1}{m}$ $(\star)$
  until the status of the channel is SINGLE;
  the station that has transmitted in the previous time slot receives ID number $n - m + 1$ and leaves the protocol
endfor

The correctness of protocol `Initialization-for-known-n` being easily seen, we now turn to the task of evaluating the number of time slots it takes the protocol to terminate. We say that the current time slot in step $(\star)$ is *successful* if the status of the channel is SINGLE. Let X be the random variable denoting the number of stations transmitting in a given round. Then, by virtue of (4) and of Lemma 2.2, at the end of this round the status of the channel is SINGLE with probability

$$\Pr[X = 1] = \binom{m}{1}\left(\frac{1}{m}\right)^1\left(1 - \frac{1}{m}\right)^{m-1} = \left(1 - \frac{1}{m}\right)^{m-1} > \frac{1}{e}.$$

Clearly, the protocol `Initialization-for-known-n` requires n successful time slots to terminate. Let Y be the random variable denoting the number of successful time slots among the first $f(n)$ time slots in step $(\star)$ of the protocol. It is clear that $E[Y] > \frac{f(n)}{e}$. We wish to determine $f(n)$ such that:

$$\Pr[Y < n] = \Pr[Y < (1-\epsilon)E[Y]] < \frac{1}{n}.$$

Now, using (11), we obtain $E[Y] = n + O(\sqrt{n \ln n})$ and, therefore, $f(n) = en + O(\sqrt{n \ln n})$.

We just proved that with probability exceeding $1 - \frac{1}{n}$, among the first $en + O(\sqrt{n \ln n})$ time slots there are at least n successful ones.

Importantly, our protocol does not rely on the existence of the collision detection capability. Therefore, we have the following result:

Theorem 3.1. *The task of initializing an n-station, single-channel AHN with known n terminates, with probability exceeding $1 - \frac{1}{n}$, in $en + O(\sqrt{n \ln n})$ time slots, regardless of whether the system has collision detection capabilities.*

At this time, we do not know whether, for known n, one can initialize an n-station AHN with CD faster than what Theorem 3.1 offers. In the remainder of the paper, we shall see that if the number of stations in the AHN is *not known* in advance, the CD capability indeed makes a difference.

2. In this work, we let $\ln n$ and $\log n$ denote the natural logarithm and the logarithm to base 2, respectively.

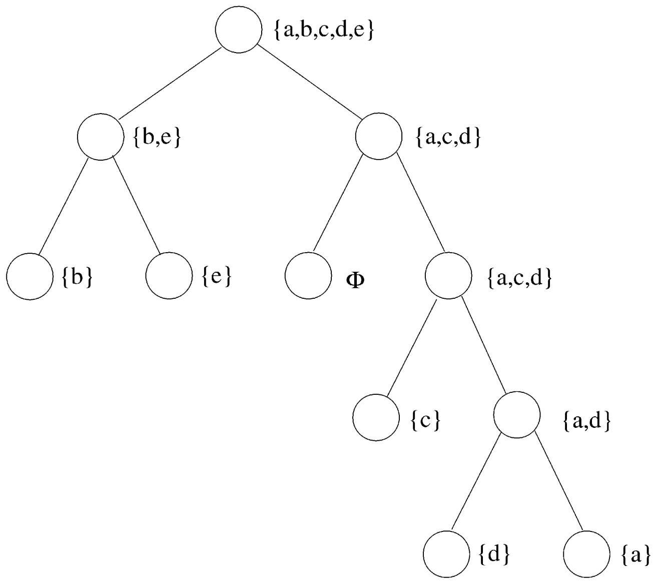

Fig. 2. Illustrating a partition tree.

4 Initializing n-Station AHNs with CD and Unknown n

The main goal of this section is to discuss initialization protocols for n-station AHNs with CD, assuming that the number n of stations is not known beforehand. We begin by presenting an initialization protocol for the single-channel AHN that terminates with probability exceeding $1-\frac{1}{n}$, in $\frac{10n}{3}+O(\sqrt{n\ln n})$ time slots. We then generalize this result to the case of the k-channel AHN with CD. Specifically, we discuss an initialization protocol for this case that terminates, with probability exceeding $1-\frac{1}{n}$, in $\frac{10n}{3k}+O\left(\sqrt{\frac{n\ln n}{k}}\right)$ time slots, subject to k satisfying a certain condition.

4.1 Initializing the Single-Channel AHN

The idea behind our initialization protocol for the single-channel AHN is to construct a full binary tree[3] that we call a *partition tree*. As it turns out, the leaves of the partition tree are, with high probability, individual stations of the AHN. Some leaves may be empty but, as we shall see, this event has a vanishing probability. The internal nodes of the partition tree are associated with groups of two or more stations. By flipping coins, these stations will be assigned to the left or right subtree rooted at that node. For an illustration of a partition tree corresponding to a 5-station AHN, we refer the reader to Fig. 2.

Each station maintains local variables l, L, and N. Let $\mathcal{P}_i$ denote the group of stations whose local variable l has value i. Notice that the collision detection capability allows us to determine whether a given node is a leaf or an internal node. This is done, simply, by mandating the stations associated with that node to transmit and by recording the corresponding status of the channel. The details of the initialization protocol follow.

3. That is, one each of whose internal nodes have exactly two children.

Protocol `Initialization-with-CD`($\mathcal{S}$);
each station sets $l \leftarrow L \leftarrow N \leftarrow 1$;
// set $\mathcal{P}_L \leftarrow \mathcal{S}$;
while $L \geq 1$ **do**
  all stations in $\mathcal{P}_L$ transmit on the channel;
  if Channel_status = COLLISION **then**
    each station in $\mathcal{P}_L$ flips a fair coin;
    all stations set $L \leftarrow L+1$;
    all stations that flipped "heads" set $l \leftarrow L$
  else
    if Channel_status = SINGLE **then**
      the unique station in $\mathcal{P}_L$ sets ID $\leftarrow N$ and leaves the protocol;
      all stations set $N \leftarrow N+1$
    endif
    all stations set $L \leftarrow L-1$
  endif
endwhile

To address the correctness of protocol `Initialization-with-CD`, we note that no station is associated with more than one node at any given height h in the partition tree. Moreover, if a station is associated with an internal node of the partition tree, then it is associated with all the nodes along the unique path to the root. Consequently, each station will end up in exactly one leaf of the tree. Since the partition tree is traversed in a depth-first fashion, each station is guaranteed to eventually receive an ID. The fact that N is only incremented when the status of the channel is SINGLE guarantees that and no two stations will receive the same ID and that the IDs are consecutive numbers from 1 to n.

Next, we turn to the task of evaluating the number of time slots it takes the protocol to terminate. Call an internal node of the partition tree *proper* if both its left and right subtrees are nonempty. It is clear that every partition tree of an n-station AHN must have exactly $n-1$ proper internal nodes.

Consider an internal node u of the partition tree that has a group U of m, $(m \geq 2)$, stations associated with it. We refer to node u as *basic* whenever U contains exactly two stations. Node u will be called *regular* if U contains at least three stations (i.e., $m \geq 3$). Let A be the event that node u is proper. Clearly, the assumption $m \geq 2$ allows us to write

$$\Pr[A] = 1 - \frac{1}{2^{m-1}}. \tag{14}$$

It follows that a basic internal node is proper with probability at least $\frac{1}{2}$, while a regular internal node is proper with probability at least $\frac{3}{4}$.

Since the partition of a basic node results in two leaves, it follows that the number of proper basic nodes in a partition tree is bounded by $\frac{n}{2}$. Similarly, since the total number of proper internal nodes is always $n-1$ and at most $\frac{n}{2}$ of these are basic, the number of proper regular nodes cannot be smaller than $\frac{n}{2}-1$.

Let Y be the random variable denoting the number of regular internal nodes partitioned in $\phi(n)$ trials. We wish to determine $\phi(n)$ such that

$$\Pr[Y < \frac{n}{2} - 1] < e^{-\ln 2n} = \frac{1}{2n}. \quad (15)$$

By using a variant of (11), we obtain $E[Y] = \frac{n}{2} + O(\sqrt{n \ln n})$. Now, since with probability at least $\frac{3}{4}$ a regular internal node is proper,

$$\phi(n) = \frac{2n}{3} + O(\sqrt{n \ln n}). \quad (16)$$

Similarly, let Z be the random variable denoting the number of basic internal nodes partitioned in $\varphi(n)$ trials. We wish to determine $\varphi(n)$ such that

$$\Pr[Z < \frac{n}{2}] < e^{-\ln 2n} = \frac{1}{2n}. \quad (17)$$

By using a variant of (11), we obtain

$$E[Z] = \frac{n}{2} + O(\sqrt{n \ln n}).$$

Since with probability at least $\frac{1}{2}$ a basic internal node is proper,

$$\varphi(n) = n + O(\sqrt{n \ln n}). \quad (18)$$

Therefore, (3), (16), and (18) combined guarantee that, with probability exceeding $1 - \frac{1}{n}$, the first $\frac{5n}{3} + O(\sqrt{n \ln n})$ iterations of the while loop will result in at least $n - 1$ proper nodes.

Notice that a partition tree with $\frac{5n}{3} + O(\sqrt{n \ln n})$ internal nodes has $\frac{5n}{3} + O(\sqrt{n \ln n})$ leaves. Since one time slot is needed to detect each leaf, we have proven the following result.

Theorem 4.1. *Even if n is not known beforehand, an n-station, single-channel AHN with CD can be initialized with probability exceeding $1 - \frac{1}{n}$, in $\frac{10n}{3} + O(\sqrt{n \ln n})$ time slots.*

We note at this time that, as shown in Theorem 3.1, knowledge of the number n of stations in the AHN allowed us to design an initialization protocol terminating, with high probability, in $en + O(\sqrt{n \log n})$ time slots. If n is not known in advance, the best we can offer is $\frac{10n}{3} + O(\sqrt{n \ln n})$ time slots. It is an interesting open problem to close this gap or to show that it cannot be closed.

4.2 Initializing an n-Station, k-Channel AHN with CD and Unknown n

Section 4.1 has shown that, even if n is not known in advance, a single-channel AHN consisting of n stations can be initialized in $\frac{10n}{3} + O(\sqrt{n \ln n})$ time slots. The main purpose of this subsection is to generalize the result of Section 4.1 to the case where k channels are available to the AHN. Of course, the best performance that we can hope to attain is roughly $\frac{10n}{3k} + O(\sqrt{\frac{n \ln n}{k}})$ time slots. As we are about to show, this performance can, indeed, be achieved.

Consider an n-station, k-channel AHN, where the number n of stations is not known in advance. The k channels are enumerated as $\mathrm{C}(1), \mathrm{C}(2), \ldots, \mathrm{C}(k)$ and are known to all the stations.

In outline, our initialization protocol involves two stages. In the first stage, we assign uniformly at random the n stations of the AHN to the k channels and proceed to initialize the group of stations assigned to each of the channels individually. Once this is done, we need to update the local IDs to reflect the overall picture. The details follow.

Protocol `Initialize-k-channel-AHN`

Stage 1 Each station chooses uniformly at random an integer between 1 and k. Let $P(i)$, $(1 \le i \le k)$, denote the group of stations that have chosen integer i. Dedicate channel $\mathrm{C}(i)$ to group $P(i)$ and use protocol `Initialization-with-CD` to initialize $P(i)$. Let $p_{i,j}$, $(1 \le i \le k; 1 \le j \le |P(i)|)$, denote the jth station in group $P(i)$. At the end of this stage, each station $p_{i,j}$ knows its local ID number j within $P(i)$.

Stage 2 For every i, $(2 \le i \le k)$, determine the sum $Z_{i-1} = |P(1)| + |P(2)| + \cdots + |P(i-1)|$. Each station $p_{i,j}$, $(2 \le i \le k)$, updates its local ID from j to $Z_{i-1} + j$.

We begin by evaluating the number of time slots involved in completing Stage 1. For this purpose, we show that, somewhat surprisingly, no group $P(i)$ is likely to contain many stations.

Fix a channel and let X be the random variable denoting the number of stations associated with that channel. It should be clear that $E[X] = \frac{n}{k}$. Write

$$\epsilon = \sqrt{\frac{3k \ln 2nk}{n}}.$$

Simple evaluations confirm that

$$0 < \epsilon < 1 \quad (19)$$

whenever

$$k \le \frac{n}{4.16 \log n + 2.08}. \quad (20)$$

By using (7) with the value of ϵ from (19), we can bound the probability that the given channel is associated with $\frac{n}{k} + \sqrt{\frac{3k \ln 2nk}{n}}$ or more stations as follows:

$$\Pr[X \ge \frac{n}{k} + \sqrt{\frac{3k \ln 2nk}{n}}] < e^{\frac{-\epsilon^2}{3} \frac{n}{k}} = e^{-\ln 2nk} = \frac{1}{2nk}. \quad (21)$$

Now, (3) and (21), combined, guarantee that with probability exceeding $1 - \frac{1}{2n}$, none of the channels are associated with more than $\frac{n}{k} + \sqrt{\frac{3k \ln 2nk}{n}}$ stations.

Thus, in the worst case, each of the k channels is associated with exactly $\frac{n}{k} + \sqrt{\frac{3k \ln 2nk}{n}}$ stations. At this point, protocol `Initialization-with-CD` is run in each of the k channels with the goal of initializing the stations associated with that channel.

Consider an arbitrary channel. Reasoning as in Subsection 4.1, any partition tree corresponding to the group of $\frac{n}{k} + \sqrt{\frac{3k \ln 2nk}{n}}$ stations associated with that channel must have at most $\frac{n}{2k} + \frac{1}{2}\sqrt{\frac{3k \ln 2nk}{n}}$ basic and at least $\frac{n}{2k} + \frac{1}{2}\sqrt{\frac{3k \ln 2nk}{n}} - 1$ regular nodes.

Let Y be the random variable denoting the number of basic internal nodes partitioned in $f(n,k)$ trials. We wish to determine $f(n,k)$ such that

$$\Pr[Y < \frac{n}{2k} + \frac{1}{2}\sqrt{\frac{3k\ln 2nk}{n}}] < e^{-\ln 2n} = \frac{1}{2n}. \qquad (22)$$

By using a variant of (11), we obtain

$$E[Y] = \frac{n}{2k} + O\left(\sqrt{\frac{n\ln n}{k}}\right).$$

Now, since with probability at least $\frac{1}{2}$ a basic internal node is proper, we have

$$f(n,k) = \frac{n}{k} + O\left(\sqrt{\frac{n\ln n}{k}}\right). \qquad (23)$$

Similarly, let Z be the random variable denoting the number of regular internal nodes partitioned in $g(n,k)$ trials. We wish to determine $g(n,k)$ such that

$$\Pr[Z < \frac{n}{2k} + \frac{1}{2}\sqrt{\frac{3k\ln 2nk}{n}}] < e^{-\ln 2n} = \frac{1}{2n}. \qquad (24)$$

By using a variant of (11), we obtain

$$E[Z] = \frac{n}{2k} + O\left(\sqrt{\frac{n\ln n}{k}}\right).$$

Since with probability at least $\frac{3}{4}$ a regular internal node is proper,

$$g(n,k) = \frac{2n}{3k} + O\left(\sqrt{\frac{n\ln n}{k}}\right). \qquad (25)$$

Therefore, (3), (23), and (25), combined, guarantee that with probability exceeding $1-\frac{1}{n}$ the first $\frac{5n}{3k} + O\left(\sqrt{\frac{n\ln n}{k}}\right)$ iterations of the **while** loop in protocol `Initialization-with-CD` will result in at least $\frac{n}{k} + \sqrt{\frac{3k\ln 2nk}{n}} - 1$ proper nodes.

Since one time slot is needed to detect each leaf, we have proven the following result:

Lemma 4.2. *With probability exceeding $1-\frac{1}{n}$, Stage 1 of protocol* `Initialize-k-channel-AHN` *terminates in $\frac{10n}{3k} + O\left(\sqrt{\frac{n\ln n}{k}}\right)$ time slots, provided that $k \le \frac{n}{4.16\log n + 2.08}$.*

It is important to note that in order to ensure correctness, before the protocol can advance to Stage 2, we must check whether all k instances of `Initialization-with-CD` running in the k channels have finished executing. For this purpose, the execution of `Initialization-with-CD` is suspended in time slots $1, 4, 9, \ldots, i^2, \ldots$. In these suspended time slots every station that has not yet been assigned a local ID number transmits on channel C(1). Clearly, if the status of C(1) is NULL, we know that Stage 1 must have terminated and Stage 2 may begin.

To assess the additional overhead incurred by the task of checking termination, assume that for some integer t the status of channel C(1) is COLLISION or SINGLE in time slot t^2 and NULL in time slot $(t+1)^2$. In other words, the last remaining instance has finished execution somewhere between time slot t^2 and time slot $(t+1)^2$. Thus, we can write

$$t^2 \le \frac{10n}{3k} + O\left(\sqrt{\frac{n\ln n}{k}}\right) < (t+1)^2. \qquad (26)$$

In turn, (26) guarantees that Stage 1 will be interrupted exactly $t+1$ times and, moreover,

$$t \in O\left(\sqrt{\frac{10n}{3k}}\right) \subseteq O\left(\sqrt{\frac{n\ln n}{k}}\right). \qquad (27)$$

Since at most $(t+1)^2 - t^2 - 1 = 2t$ time slots could have elapsed between the actual termination and the moment when termination was detected, (27) guarantees that the task of checking termination of Stage 1 incurs a penalty of $O\left(\sqrt{\frac{n\ln n}{k}}\right)$ that does not have a serious impact on performance, since it only affects lower order terms in Lemma 4.2.

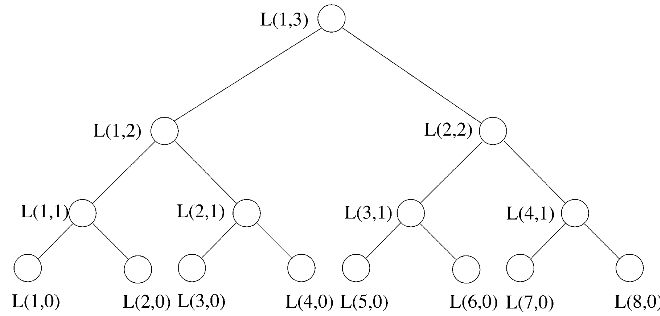

Fig. 3. Illustrating the recursive numbering of the internal nodes of $\mathcal{T}$.

Recall that $P(i)$, $(1 \le i \le k)$, denotes the group of stations associated with channel $C(i)$ in Stage 1. We assume that k is a power of 2. The reader will not fail to see that this is assumed for simplicity only and that the general case is similar.

Stage 2 of the protocol is *deterministic* and works regardless of whether some of the $P(i)$s are empty. In case group $P(i)$ is not empty, we let $L(i,0)$ stand for a leader of $P(i)$, taken to be the station having local ID of 1 in $P(i)$. It is clear that at the end of Stage 1, station $L(i,0)$ knows the number $|P(i)|$ of stations in $P(i)$. Also, we write for all i, $(1 \le i \le k)$, $A(i,0) = |P(i)|$.

Consider the complete binary tree $\mathcal{T}$ whose leaves, enumerated in preorder, are the k leaders $L(1,0), L(2,0), \ldots, L(k,0)$ and refer to Fig. 3. The internal nodes of $\mathcal{T}$ will be labeled recursively. For now, we let $T(i,j)$ be the subtree of $\mathcal{T}$ whose leaves are

$$L((i-1)2^{j-1}+1, 0), L((i-1)2^{j-1}+2, 0), \ldots L(i2^j, 0).$$

At this point, we are ready to define $L(i,j)$ and $A(i,j)$ for all j, $(1 \le j \le \log k)$. For every i, $(1 \le i \le \frac{k}{2^j})$, we write

$$A(i,j) = A(2i-1, j-1) + A(2i, j-1). \qquad (28)$$

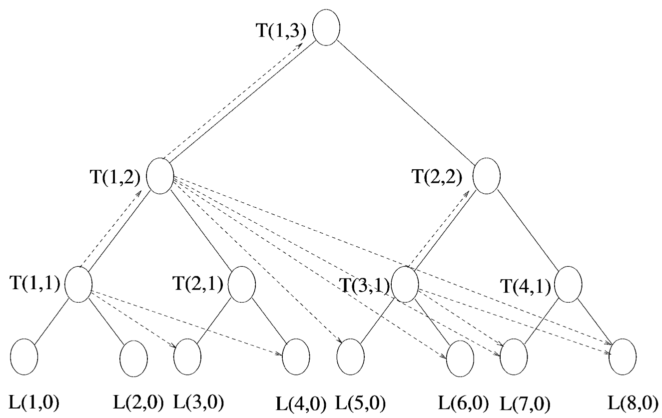

Fig. 4. Illustrating the update of the various leaves in $\mathcal{T}$.

Similarly, if none of the stations $L(2i-1, j-1)$ and $L(2i, j-1)$ exists, then $L(i,j)$ does not exist, either. Otherwise,

$$L(i,j) = \begin{cases} L(2i-1, j-1) & \text{if } L(2i-1, j-1) \text{ exists} \\ L(2i, j-1) & \text{otherwise.} \end{cases} \quad (29)$$

Stage 2 involves $\log k$ steps that we describe next. In Step j, $(1 \leq j \leq \log k)$, in two time slots, taking turns, the stations $L(2i-1, j-1)$ and $L(2i, j-1)$, $(1 \leq i \leq \frac{k}{2^j})$, transmit the values $A(2i-1, j-1)$ and $A(2i, j-1)$ on channel $C(2i-1)$. Of course, if none of the stations $L(2i-1, j-1)$ and $L(2i, j-1)$ exists, nothing gets transmitted.

The reader should be in a position to confirm that for every j, $(1 \leq j \leq \log k)$, and for every i, $(1 \leq i \leq \frac{k}{2^j})$:

- every station $L(i,j)$ is associated with the root of the subtree $T(i,j)$;
- station $L(i,j)$ exists if and only if *any* of these stations exists;
- an easy inductive argument shows that the value $A(i,j)$ associated with $L(i,j)$ is:

$$A(i,j) = |P((i-1)2^{j-1}+1)| + |P((i-1)2^{j-1}+2)| + \cdots + |P(i2^j)|. \quad (30)$$

- the value $A(1, \log k)$ associated with the root of the tree $\mathcal{T} = T(1, \log k)$ is $\sum_{i=1}^{k} |P(i)|$.

Rather interestingly, the task of updating the various sums can be performed concurrently with that of computing the values $L(i,j)$ and $A(i,j)$ defined above. The idea is simple and is illustrated in Fig. 4. Specifically, while station $L(2i-1, j)$ is transmitting the value $A(2i-1, j)$, all the stations that are leaves of the subtree $T(2i, j)$ pick up the value transmitted and use it to update the partial sum they store. A simple inductive argument based on (30) shows that at the end of $2\log k$ time slots all the leaders $L(i, 0)$, $(1 \leq i \leq k)$, have computed the correct sums $Z(i)$. An additional transmission allows each station in $P(i)$, $(2 \leq i \leq k)$, AHN to update their local ID by adding $Z(i-1)$ to it. Consequently, Stage 2 terminates in $2\log k$ time slots.

Notice that if k satisfies 20, we can write

$$\log k \in O\left(\sqrt{\frac{n \ln n}{k}}\right)$$

and, consequently, Stage 2 does not have an impact on the overall complexity of the protocol.

To summarize, we have proved the following result.

Theorem 4.3. *Even if the number n of stations is not known beforehand, an n-station, k-channel AHN with CD can be initialized with probability exceeding $1 - \frac{1}{n}$, in $\frac{10n}{3k} + O\left(\sqrt{\frac{n \ln n}{k}}\right)$ time slots, provided that k satisfies 20.*

5 Initializing the AHN with No-CD and Unknown n

Up to now, we have dealt with the case where the AHN has a collision detection capability. In many practical situations, especially in the presence of noisy channels, collision detection is rather hard to perform with any degree of accuracy. Moreover, collision detection requires special hardware that may not be available, or may be impractical to provide in the case of small hand-held devices.

The main goal of this section is to address the problem of initializing an n-station AHN that does not have the CD capability. We further assume that the number n of stations in the AHN is not known beforehand.

Our approach to initializing the AHN in the no-CD case is to first elect a leader in the AHN. The motivation for this is that, as we also show, once a leader is elected, a no-CD AHN can simulate in two time slots the AHN with CD. This result is quite surprising and novel. In addition, it is interesting in its own right.

5.1 Electing a Leader in the Single-Channel AHN with No-CD and Unknown n

We assume an n-station, single-channel AHN with no collision detection capability and where the number n of stations is not known beforehand. In the worst case, the stations are identical and cannot be distinguished by serial or manufacturing number.

The well-known *leader election* problem asks to designate one of the stations of the AHN as *leader*. This is a fundamental task in group collaboration and has been addressed before in multiprocessor systems [5], [28], [33], [41], [42], [44].

In a previous paper, Willard [44] showed that a leader can be elected in an n-station, single-channel AHN with CD in $O(\log\log n)$ expected time. Recently, we [33] have extended Willard's result by showing that the same task can be performed in $O(\log\log n)$ time slots with probability exceeding $1 - O(\frac{1}{\log n})$ or in $O(\log n)$ time slots with probability exceeding $1 - O(\frac{1}{n})$.

However, to the best of our knowledge, the important problem of electing a leader in an AHN that does not have the collision detection capability was open. The lack of collision detection makes the problem significantly more difficult. The protocols of [33], [44] rely in a crucial way on collision detection and, consequently, cannot be used in systems lacking this capability. Not surprisingly, our protocol is vastly different.

As a preamble, we check to see if the AHN contains exactly one station. If such is the case, a leader is elected and there is nothing left to do. Otherwise, the following protocol is executed.

Protocol `Leader-election-with-no-CD`
for $i \leftarrow 1$ to ∞ do
  for $j \leftarrow 1$ to i do
    each station transmits on the channel with probability $\frac{1}{2^j}$;
    if the status of the channel is SINGLE `then`
      the station that transmitted is declared the leader and the protocol terminates;
    endif
  end for
end for

It is clear that protocol `Election-with-no-CD` terminates with the correct election of a leader. We therefore turn to the task of evaluating the number of time slots it takes the protocol to terminate.

Let s be the unique integer satisfying $2^s \leq n < 2^{s+1}$. We say that a time slot j is *promising* if $j = s + 1$. Note that by Lemma 2.2, a promising time slot succeeds to find a leader with probability at least

$$\binom{n}{1}\left(\frac{1}{2^{s+1}}\right)^1\left(1-\frac{1}{2^{s+1}}\right)^{n-1} \geq \frac{2^s}{2^{s+1}}\left[\left(1-\frac{1}{2^{s+1}}\right)^{2^{s+1}-1}\right]^{\frac{n-1}{2^{s+1}-1}} \geq \frac{1}{2}e^{-\frac{n-1}{2^{s+1}-1}} \geq \frac{1}{2}\cdot e^{-1} \geq \frac{1}{2e}. \quad (31)$$

Now, Lemma 2.1 implies that the probability that the first t promising time slots fail to find a leader is less than

$$\left(1-\frac{1}{2e}\right)^t < e^{-\frac{t}{2e}}. \quad (32)$$

Put differently, (32) tells us that with probability exceeding

$$1 - e^{-\frac{t}{2e}},$$

the first t promising time slots in the protocol will succeed in electing a leader.

Notice that the first $s + t$ iterations of the outer `for`-loop, corresponding to $i = 1, 2, \ldots, s, s+1, \ldots, s+t$ are guaranteed to contain t promising time slots. It follows that the first $1 + 2 + \cdots + (s+t)$ time slots must contain t promising ones.

By selecting $t = 2e \ln n < 3.769\ldots \log n$ and by recalling that $s \leq \log n$ we have proven the following result:

Theorem 5.1. *With probability exceeding $1 - \frac{1}{n}$, one can elect a leader in an n-station, single-channel AHN in fewer than $11.37(\log n)^2 + 2.39 \log n$ time slots, even if n is not known beforehand.*

For later reference, we note that by selecting $t = 2e \ln kn < 3.769\ldots \log kn$, we have the following result.

Corollary 5.2. *With probability exceeding $1 - \frac{1}{kn}$, one can elect a leader in an n-station, single-channel AHN in fewer than $11.37(\log(nk))^2 + 2.39 \log(nk)$ time slots, even if n is not known beforehand.*

5.2 Simulating Collision Detection

The reader should be aware by now that, in essence, the difference between collision detection and its absence is that given a subset $\mathcal{P}$ of stations, the AHN with CD can detect in one time slot whether:

- (cd1) $|\mathcal{P}| = 1$,
- (cd2) $|\mathcal{P}| = 0$, or
- (cd3) $|\mathcal{P}| \geq 2$,

whereas the AHN with no-CD can detect only Condition cd1 and cannot distinguish between cd2 and cd3.

Somewhat surprisingly, once a leader is elected, the AHN with no-CD can simulate the AHN with CD in two time slots. In other words, in one additional time slots the AHN with no-CD can discern between Conditions cd2 and cd3.

To see that this is the case, assume that a leader p was elected in the AHN. The leader receives the ID of 1 and will play a special role in the AHN. In particular, p is not considered to be part of any subset $\mathcal{P}$ of stations.

To see how the simulation proceeds, the stations in $\mathcal{P}$ transmit on the channel. If the status of the channel is SINGLE, Condition cd1 holds and no further action is necessary. Suppose, therefore, that the status of the channel is NOISE. Now p together with all the stations in $\mathcal{P}$ transmit on the channel again. If the status is SINGLE, then it is clear that $|\mathcal{P}| = 0$ and condition (cd2) holds. Finally, if the status of the channel is NOISE, then it must be that $|\mathcal{P}| \geq 2$ holds.

Thus, if a leader is elected beforehand, every time slot of a single-channel AHN with CD can be simulated in two time slots of the AHN with no-CD. To summarize, we state the following important result.

Theorem 5.3. *Assuming that a leader has been elected, any protocol running in T time slots on an n-station, single-channel AHN with CD will take at most $2T$ time slots on an n-station, single-channel AHN with no-CD.*

On the other hand, if a leader is not available, then we can use our randomized leader election protocol. Of course, in this case the overhead of electing a leader must be added to the overall complexity of the protocol at hand.

Theorem 5.4. *Consider a protocol that terminates with probability exceeding $1 - \frac{1}{n}$, in T time slots on an n-station, single-channel AHN. Then the same protocol will terminate with probability exceeding $1 - \frac{1}{n}$, in fewer than $2T + E(n)$ time slots on the n-station, single-channel AHN with no-CD, where $E(n) = 11.37(\log n)^2 + 2.39 \log n$.*

5.3 Initializing the AHN with No-CD and Unknown n

In this section, we show how all the pieces fit together, resulting in efficient initialization protocols for the AHN with no-CD capability. We begin by electing a leader in the AHN, as discussed in Section 5.1.

Observe that the availability of the leader allows us to simulate every time slot of the initialization protocol for the single-channel AHN discussed in Section 4.1 in at most two time slots. One can make a small improvement, however. Recall that the complexity of protocol `Initialization-with-CD` has two major components: processing internal nodes of the partition tree, and processing the leaves. Clearly, exactly n of these leaves must be single stations in the AHN. As a consequence, when detecting such a leaf, the second step of the simulation described in Section 5.1 is not necessary. With this observation, the overall number of time slots required drops from $\frac{20}{3}n + E(n) + O(\sqrt{n \ln n})$ to $\frac{17}{3}n + E(n) + O(\sqrt{n \ln n})$. In addition, notice that

$$E(n) \in O(\sqrt{n \ln n}).$$

Consequently, we have the following result.

Theorem 5.5. *Even if n is not known beforehand, an n-station, single-channel AHN with no-CD can be initialized with probability exceeding $1 - \frac{1}{n}$, in fewer than $5.67n + O(\sqrt{n \ln n})$ time slots.*

We now turn to the initialization of the n-station, k-channel AHN with no-CD. Recall that the protocol `Initialize-k-channel-AHN` discussed in Section 4.2 involves two stages. The first stage involved setting up k instances of the initialization problem in each of the k channels.

Having set up these k instances, we begin by electing a leader in the group of stations assigned to each channel, incurring a time penalty of $E(\frac{n}{k} + O(\sqrt{\frac{n \ln n}{k}})) < E(\frac{2n}{k})$. Once a leader is elected in each group, we simulate the protocol `Initialization-with-CD` discussed in Section 4.1 with the above suggested improvement.

In this case, however, only Stage 1 has to be simulated, as Stage 2 is deterministic and does not use simultaneous transmissions on any channel.

Since in typical situations the number k of channels is much smaller than the number n of stations in the AHN, it is reasonable to assume that

$$k \leq \frac{n}{(\log n)^3}.$$

With this assumption, both $E(\frac{2n}{k}) \in O\left(\sqrt{\frac{n \ln n}{k}}\right)$ and $k \in O\left(\sqrt{\frac{n \ln n}{k}}\right)$. This observation, along with Corollary 5.2, imply the following important result.

Theorem 5.6. *Even if the number n of stations is not known beforehand, an n-station, k-channel AHN with no-CD can be initialized with probability exceeding $1 - \frac{1}{n}$, in fewer than $5.67\frac{n}{k} + O\left(\sqrt{\frac{n \ln n}{k}}\right)$ time slots, provided that $k \leq \frac{n}{(\log n)^3}$.*

6 Concluding Remarks and Open Problems

An ad hoc network (AHN, for short) is a distributed system with no central arbiter, consisting of n mobile radio transceivers, referred to as *stations*. Typically, these stations are small, inexpensive, commodity hand-held devices running on batteries that are deployed on demand in support of various events: collaborative computing, disaster-relief, search-and-rescue, or law enforcement operations.

Our main contribution was to address one of the fundamental tasks in setting up an AHN, namely that of initializing the network both in the case where collision detection is available and when it is not. To the best of our knowledge, these are the first initialization results reported for ad hoc networks.

There are a number of problems that remain open. First, we do not know how to take advantage of the fact that in some cases the number n of stations is known at the outset. This is the case in collaborative environments—for example, in a brain-storming session in which the participants cooperate to attain a certain goal. The problem is that in the presence of k radio channels, once we assign the stations to channels, the number of stations per channel is no longer known exactly. This is, in fact, the approach we took in this work. Can one do better? Yet another important avenue is to realize that stations in the AHN often have severe power constraints [7], [19], [22]. It is important, therefore, to devise protocols that are as energy-efficient as possible. Ideally, the protocols should allow the end user to specify the amount of power available and seamlessly integrate power restrictions into the protocol.

The initialization protocols presented in this work assign *consecutive* ID numbers to the stations of the AHN. While this is very useful for the purpose of designing efficient protocols, the consecutive numbering of the stations is only suitable for low-mobility environments. As an illustration, consider an ad hoc network set up in a conference room where the participants engage in a collaborative multimedia session. Clearly, in this case mobility is low and, once established, the consecutive ID numbering is easily maintained. An interesting open problem to address in the future is a mechanisms for handling (highly) dynamic group membership.

Finally, we note that the protocols discussed in this paper rely on the fact that the stations can keep synchronous time by interfacing with a GPS system. It is important to look at various less stringent synchronization scenarios and to adapt our protocols for these environments. This promises to be a fascinating area for future work.

Acknowledgments

This work was supported by ONR grant N00014-97-1-0526, and was performed while K. Nakano visited old Dominion University on a grant from the Telecommunication Advancement Foundation of Japan. The authors wish to express their appreciation to the three anonymous referees for their constructive comments that have greatly contributed to enhancing the overall quality of the work.

References

[1] N. Abramson, ed., *Multiple Access Communications: Foundations for Emerging Technologies.* New York: IEEE Press, 1993.

[2] N. Abramson, "Multiple Access in Wireless Digital Networks," *Proc. IEEE,* vol. 82, pp. 1,360–1,370, 1994.

[3] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, "Single-Round Simulation on Radio Networks," *J. Algorithms,* vol. 13, pp. 188–210, 1992.

[4] D. Angluin, "Global and Local Properties in Networks of Processors," *Proc. 12th ACM Symp. Theory of Computing,* pp. 82–93, 1980.

[5] H. Attiya, "Distributed Computing Theory," *Parallel and Distributed Computing Handbook.* pp. 127–162, New York: McGraw-Hill, 1996.
[6] D.J. Baker, "Data/Voice Communication over a Multihop, Mobile, High Frequency Network," *Proc. MILCOM '97,* pp. 339–343, 1997.
[7] N. Bambos and J.M. Rulnick, "Mobile Power Management for Wireless Communication Networks," *Wireless Networks,* vol. 3, pp. 3–14, 1997.
[8] R. Bar-Yehuda, O. Goldreich, and A. Itai, "Efficient Emulation of Single-Hop Radio Network with Collision Detection on Multi-Hop Radio Network with No Collision Detection," *Distributed Computing,* vol. 5, pp. 67–71, 1991.
[9] R. Bar-Yehuda, O. Goldreich, and A. Itai, "On the Time-Complexity of Broadcast in Multi-Hop Radio Networks: An Exponential Gap between Determinism and Randomization," *J. Computer and Systems Sciences,* vol. 45, pp. 104–126, 1992.
[10] D. Bertzekas and R. Gallager, *Data Networks,* second ed. Prentice-Hall, 1992.
[11] V. Bhargavan, A Demers, S. Shenkar, and L. Zhang, MACAW: Media Access Protocol for Wireless LANs, *Proc. ACM SIGCOMM '94,* pp. 212–225, 1994.
[12] R. Binder, N. Abramson, F. Kuo, A. Okinaka, and D. Wax, "ALOHA Packet Broadcasting–A Retrospect," *Proc. AFIPS Conf.,* pp. 203–216, May 1975.
[13] U. Black, *Mobile and Wireless Networks.* Upper Saddle River, N.J.: Prentice-Hall, 1996.
[14] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems,* second ed. Addison-Wesley, 1994.
[15] F. Cristian, "Reaching Agreement on Processor-Group Membership in Synchronous Distributed Systems," *Distributed Computing,* vol. 4, pp. 254–264, 1991.
[16] P.H. Dana, The Geographer's Craft Project, Dept. of Geography, UT Austin, Sept. 1999. http://www.utexas.edu/depts/grg/gcraf/notes/gps/gps.html.
[17] D. Duchamp, S.K. Feiner, and G.Q. Maguire, "Software Technology for Wireless Mobile Computing," *IEEE Network Magazine,* pp. 12–18, Nov. 1991.
[18] K. Feher, *Wireless Digital Comm.,* Prentice-Hall, 1995.
[19] W.C. Fifer and F.J. Bruno, "Low Cost Packet radio," *Proc. IEEE,* vol. 75, pp. 33–42, 1987.
[20] V.K. Garg and J.E. Wilkes, *Wireless and Personal Communication Systems,* Prentice-Hall, 1996.
[21] M. Gerla and T.-C. Tsai, "Multicluster, Mobile, Multimedia Radio Network," *Wireless Networks,* vol. 1, pp. 255–265, 1995.
[22] E.P. Harris and K.W. Warren, "Low Power Technologies: A System Perspective," *Proc. Thirrd Int'l Workshop on Multimedia Comm.,* 1996.
[23] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable Routing Strategies for Ad hoc Networks," *IEEE J. Selected Areas in Comm.,* vol. 17, pp. 1,369–1,379, 1999.
[24] E.D. Kaplan, *Understanding GPS: Principles and Applications.* Boston: Artech House, 1996.
[25] P. Krishna, N.H. Vaidya, M. Chatterjee, and D.K. Pradhan, "A Cluster-Based Approach for Routing in Dynamic Networks," *ACM SIGCOMM Review,* vol. 27, no. 2, Apr. 1997.
[26] C.R. Lin and M. Gerla, "Adaptive Clustering in Mobile Wireless Networks," *IEEE J. Selected Areas in Communications,* vol. 16, pp. 1,265–1,275, 1997.
[27] C.R. Lin and M. Gerla, "Real-Time Support in Multihop Wireless Networks," *Wireless Networks,* vol. 5, pp. 125–135, 1999.
[28] R.J. Lipton and A. Park, "The Processor Identity Problem," *Information Processing Letters,* vol. 36, pp. 91–94, 1990.
[29] G.Y. Liu, A. Marlevi, and G.Q. Maguire, Jr.,, "A Mobile Virtual-Distributed System Architecture for Supporting Wireless Mobile Computing and Communications," *Wireless Networks,* vol. 2, pp. 77–86, 1996.
[30] R.M. Metcalfe and D.R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," *Comm. ACM,* vol. 19, pp. 395–404, 1976.
[31] R. Motwani and P. Raghavan, *Randomized Algorithms.* Cambridge Univ. Press, 1995.
[32] K. Nakano, "Optimal Initializing Algorithms for a Reconfigurable Mesh," *J. Parallel and Distributed Computing,* vol. 24, pp. 218-223, 1995.
[33] K. Nakano and S. Olariu, "Randomized $O(\log\log n)$-Round Leader Election Protocols in Packet Radio Networks," *Proc. Int'l Symp. Algorithms and Computation,* pp. 209–218, 1998.
[34] K. Nakano, S. Olariu, and J.L. Schwing, "Efficient Routing, Ranking, and Sorting Protocols for Ad Hoc Radio Networks," *IEEE Trans. Parallel and Distributed Systems,* to appear.
[35] M. Joa-Ng and I.-T. Lu, "A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks," *IEEE J. Selected Areas in Comm.,* vol. 17, pp. 1415–1425, 1999.
[36] B. Parkinson and S. Gilbert, "NAVSTAR: Global Positioning System–Ten Years later," *Proc. IEEE,* pp. 1,177–1,186, 1983.
[37] R. Ramanathan and M. Steenstrup, "Hierarchically-Organized, Multihop Wireless Networks for Quality-of-Service Support," *Mobile Networks and Applications,* vol. 3, pp. 101–119, 1998.
[38] D. Raychaudhuri and N.D. Wilson, "ATM-Based Transport Architecture for Multiservice Wireless PCN," *IEEE J. Selected Areas in Comm.,* vol. 12, pp. 1,401–1,414, 1994.
[39] R. van Renesse personal communication March 2000.
[40] R. Sanchez, J. Evans, and G. Minden, "Networking on the Battlefield: Challenges in Highly Dynamic Multihop Wireless Networks," *Proc. IEEE MILCOM '99,* Oct. 1999.
[41] G. Singh, "Real-Time Leader Election," *Information Processing Letters* vol. 49, pp. 57–61, 1994.
[42] G. Singh, "Leader Election in the Presence of Link Failures," *IEEE Trans. Parallel and Distributed Systems,* vol. 7, pp. 231–236, 1996.
[43] S. Singh and J.F. Kurose, "Electing Good Leaders," *J. Parallel and Distributed Computing,* vol. 21, pp. 184–201, 1990.
[44] D.E. Willard, "Log-Logarithmic Selection Resolution Protocols in a Multiple Access Channel," *SIAM J. Computing,* vol. 15, pp. 468–477, 1986.

Koji Nakano, received the BE, ME, and PhD degrees from Osaka University, Japan in 1987, 1989, and 1992, respectively. From 1992-1995, he was a research scientist at Advanced Research Laboratory, Hitachi Ltd. Since 1995, he has worked at Nagoya Institute of Technology, Japan. In 1999, he was an adjunct associate professor of Old Dominion University. He is currently an associate professor with the Department of Electrical and Computer Engineering, at the Nagoya Institute of Technology.

Dr. Nakano has published in the following publications: *IEEE Transactions on Parallel and Distributed Systems*, *Journal of Parallel and Distributed Computing*, *Theoretical Computer Science*, *Theory of Computing Systems*, *Parallel Algorithms and Applications*, and *IEICE Transactions*. He serves on the editorial board of *International Journal of Foundations of Computer Science*. He served as an editor of the *Journal of Information Processing Society* of Japan. He has served as a program chair or committee member of several conferences and workshops, including the International Conference on Parallel Processing, the International Conference on Parallel and Distributed Computing and Systems, the Reconfigurable Architecture Workshop, and the Workshop on Advances in Parallel and Distributed Computational Models. His research interests include parallel algorithms and architectures, mobile computing, computational complexity, and graph theory. He is a member of the IEEE Computer Society.

Stephan Olariu received the MSc and PhD degrees in computer science from McGill University, Montreal in 1983 and 1986, respectively. In 1986, he joined the Computer Science Department at Old Dominion University, where he is now a professor.

Dr. Olariu has published extensively in various journals, book chapters, and conference proceedings. His research interests include wireless networks and mobile computing, parallel and distributed systems, performance evaluation, and medical image processing.

Dr. Olariu serves on the editorial board of several archival journals, including *IEEE Transactions on Parallel and Distributed Systems*, *Journal of Parallel and Distributed Computing*, *International Journal of Foundations of Computer Science*, *International Journal of Computer Mathematics*, *VLSI Design*, and *Parallel Algorithms and Applications*. He is a member of the IEEE Computer Society.