Randomized Initialization Protocols for Packet Radio Networks *

Tatsuya Hayashi† Koji Nakano† Stephan Olariu‡

Abstract

The main contribution of this work is to propose efficient randomized leader election and initialization protocols for Packet Radio Networks (PRN, for short). As a result of the initialization protocol, the n stations of a PRN are assigned distinct integer IDs from 1 to n. The results include protocols to: (1) initialize the single-channel PRN with the collision detection (CD) capability in $O(n)$ rounds with probability at least $1 - \frac{1}{2^n}$; (2) initialize the k-channel PRN with CD capability in $O(\frac{n}{k})$ rounds with probability at least $1 - \frac{1}{n}$, whenever $k \leq \frac{n}{3\log n}$; (3) elect a leader on the single-channel PRN with no CD in $O((\log n)^2)$ broadcast rounds with probability at least $1 - \frac{1}{n}$; (4) initialize the single-channel PRN with no CD in $O(n)$ rounds with probability at least $1 - \frac{1}{2^{\sqrt{n}}}$; (5) initialize the k-channel PRN with no CD in $O(\frac{n}{k})$ broadcast rounds with probability at least $1 - \frac{1}{n}$, whenever $k \leq \frac{n}{4(\log n)^2}$.

1 Introduction

A Packet Radio Network (PRN, for short) $\mathcal{S}$ consists of n radio transceivers, henceforth referred to as *stations*. The stations are identical and cannot be distinguished by serial or manufacturing number. We refer the reader to Figure 1 depicting a 7-station PRN.

As customary, the time is assumed to be slotted and all the stations have a local clock that keeps (synchronous) time. Also, all broadcast operations are performed at time slot boundaries. The stations are assumed to have the computing power of a usual laptop computer; in particular, they all run the same protocol and can generate random bits that provide "local data" on which the stations may perform computations. Further, we assume that no station knows the number n of stations.

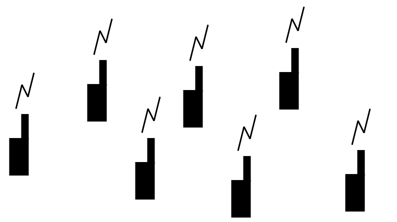

Figure 1. Illustrating a 7-station PRN.

The stations communicate using k radio channels denoted by $\mathrm{C}(1), \mathrm{C}(2), \ldots, \mathrm{C}(k)$. We assume that in any time slot, a station can tune to one such channel and/or broadcast on at most one (possibly the same) channel. A broadcast operation involves a data packet whose length is such that the broadcast operation can be completed within one time slot.

We employ two assumptions in terms of the capability of the system. In the PRN with *collision detection* (CD, for short), the status of a radio channel is: *NULL:* if no station broadcasts on the channel in the current time slot; *SINGLE:* if exactly one station broadcasts on the channel in the current time slot; and *COLLISION:* if two or more stations broadcast on the channel in the current time slot.

In the PRN with no collision detection the status of a radio channel is: *NOISE:* if either no station broadcasts or two or more stations broadcast on the channel in the current time slot; and *SINGLE:* if exactly one station broadcasts on the channel in the current time slot.

The classical *leader election* problem asks to designate one of the stations as *leader*. The leader election problem is fundamental, for many other problems rely directly or indirectly on the presence of a leader in a network [6, 7]. In a previous paper [6], we have presented an $O(\log\log n)$-round randomized leader election protocol in the single-channel PRN with CD.

The *initialization* problem is to assign to each of the n stations in $\mathcal{S}$ an integer ID number in the range $[1, n]$ such that no two stations are assigned the same ID. The

*Work supported in part by the National Science Foundation under grant CCR-9522093, by ONR grant N00014-91-1-0526, and by Grant-in-Aid for Scientific Research (C) (10680351) from Ministry of Education, Science, Sports, and Culture of Japan.

†Dept. of Electrical and Computer Engineering, Nagoya Institute of Technology, Showa-ku, Nagoya 466-8555, JAPAN, nakano@elcom.nitech.ac.jp

‡Department of Computer Science, Old Dominion University, Norfolk, Virginia 23529, USA, olariu@cs.odu.edu

initialization problem is fundamental in both network design and in multiprocessor systems [5]

The main contribution of this work is to propose efficient randomized initialization protocols for Packet Radio Networks. To begin, we show that the single-channel PRN with CD can be initialized in $O(n)$ broadcast rounds with probability at least $1 - \frac{1}{2^n}$. We then generalize this result by showing that the k-channel PRN with CD can be initialized in $O(\frac{n}{k})$ broadcast rounds with probability at least $1 - \frac{1}{n}$, whenever $k \leq \frac{n}{3 \log n}$.

Next, we design a *leader election* protocol for the single-channel PRN with no CD that terminates in $O((\log n)^2)$ broadcast rounds with probability at least $1 - \frac{1}{n}$. This leader election protocol is key in designing an initialization protocol for the single-channel PRN that terminates in $O(n)$ broadcast rounds with probability at least $1 - \frac{1}{2^{\sqrt{n}}}$. Using this protocol, we design an initialization protocol for the k-channel PRN with no CD that runs in $O(\frac{n}{k})$ broadcast rounds with probability at least $1 - \frac{1}{n}$, provided that $k \leq \frac{n}{4(\log n)^2}$.

2 Initializing the PRN with CD

We begin by presenting an initialization protocol for the single-channel PRN that terminates in $O(n)$ broadcast rounds with probability at least $1 - \frac{1}{2^n}$. Next, we generalize this result to the case of the k-channel PRN with CD.

Throughout, $\Pr[A]$ will denote the probability of event A. 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, $(0 \leq r \leq n)$,

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

Further, the expected value of X is given by

$$E[X] = \sum_{r=0}^{n} r \cdot \Pr[X = r] = np. \quad (2)$$

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

$$\Pr[X \leq (1-\epsilon)E[X]] \leq e^{-\frac{\epsilon^2}{2}E[X]} (0 \leq \epsilon \leq 1). \quad (3)$$

$$\Pr[X \geq (1+\epsilon)E[X]] \leq e^{-\frac{\epsilon^2}{3}E[X]} (0 \leq \epsilon \leq 1). \quad (4)$$

2.1 Initializing the single-channel PRN

In outline, our protocol partitions the stations into non-empty subsets $\mathcal{P}_1$ and $\mathcal{P}_2$. In turn, $\mathcal{P}_2$ is also partitioned into two non-empty subsets $\mathcal{P}_2$ and $\mathcal{P}_3$. The same partition is then applied to $\mathcal{P}_3$. In general, $\mathcal{P}_{i-1}$ is partitioned into non-empty subsets $\mathcal{P}_{i-1}$ and $\mathcal{P}_i$. This procedure is repeated until, at some stage, some $\mathcal{P}_l$ contains a single station. This station receives the ID of 1 and quits the protocol. After that, the same partition procedure is applied to $\mathcal{P}_{l-1}$ and so on. This is then repeated until all stations in the PRN have been assigned IDs. In order to perform the partitioning above, we use the protocol `Partition-with-CD`, which partitions the stations into two non-empty subsets. The details are spelled out as follows:

Protocol `Partition-with-CD`
repeat
  each station selects 0 or 1 with probability $\frac{1}{2}$;
  all the stations that selected 1 broadcast on channel C(1);
  let $Status(1)$ be the resulting status of C(1);
  all the stations that selected 0 broadcast on channel C(1);
  let $Status(0)$ be the resulting status of C(1)
until neither $Status(1)$ nor $Status(0)$ is NULL.

It is easy to see that protocol `Partition-with-CD` correctly partitions the set of stations into two non-empty subsets. We now present the details of our initialization protocol using `Partition-with-CD`. Each station maintains local variables l, L and N. Let $\mathcal{P}_i$ denote the set of stations whose local variable l has value i. Notice that the collision detection capability allows us to determine whether $|\mathcal{P}_i| = 0$, $|\mathcal{P}_i| = 1$, or $|\mathcal{P}_i| \geq 2$. This is done, simply, by mandating the stations in $\mathcal{P}_i$ to broadcast and by recording the resulting status of the channel.

Protocol `Initialization-with-CD`
every station sets $l \leftarrow L \leftarrow N \leftarrow 1$;
while $L \geq 1$ **do**
  if $|\mathcal{P}_L| = 1$ **then**
    the unique station in $\mathcal{P}_L$ is declared the N-th station and quits the protocol;
    every station sets $N \leftarrow N + 1$ and $L \leftarrow L - 1$;
  else
    use protocol `Partition-with-CD` to partition $\mathcal{P}_L$ into two non-empty sets $\mathcal{P}_L$ and $\mathcal{P}_{L+1}$;
    every station sets $L \leftarrow L + 1$;
    every station in $\mathcal{P}_L$ sets $l \leftarrow L$;
  endif
endwhile

The task of checking whether $|\mathcal{P}_L| = 1$ can be done in one broadcast. Let us estimate how many times the protocol checks if $|\mathcal{P}_L| = 1$. Note that in case $|\mathcal{P}_L| = 1$ the unique station in $\mathcal{P}_L$ is assigned an ID and leaves the protocol. Thus, exactly n times the condition $|\mathcal{P}_L| = 1$ must evaluate to "true" in the **if** statement. The protocol `Partition-with-CD` partitions a set of stations into two non-empty subsets. Hence, protocol `Partition-with-CD` must be executed exactly $n - 1$

times. Thus, exactly $n-1$ times the condition $|\mathcal{P}_L| = 1$ is "false" in the **if** statement. Therefore, the task of checking $|\mathcal{P}_L| = 1$ in the **if** statement requires $2n-1$ broadcasts.

Next, we evaluate the number of broadcast rounds involved in `Partition-with-CD`. Suppose that m, ($m \geq 2$), stations are to be partitioned into two non-empty sets using protocol `Partition-with-CD`. We say that an iteration of the **repeat-until** loop in protocol `Partition-with-CD` is *successful* if it succeeds in partitioning the set into two non-empty subsets. Let X be the random variable denoting the number of stations that selected a 1. Then, since $m \geq 2$, the probability of a successful iteration is

$$\Pr[1 \leq X \leq m-1] = 1 - \Pr[X = 0] - \Pr[X = m] \geq \frac{1}{2}.$$

Since a successful iteration produces two non-empty sets, it is clear that `Initialization-with-CD` must perform, overall, $n-1$ successful iterations. Let Y be the random variable denoting the number of successes among $8n$ Bernoulli trials, with parameter $p = \frac{1}{2}$. It is clear that $E[Y] = 4n$. By virtue of (3) with $\epsilon = \frac{3}{4}$,

$$\Pr[Y < n-1] < \Pr[Y \leq (1 - \frac{3}{4})E[Y]] < 2^{-\frac{9}{8}n} < \frac{1}{2^n}.$$

We just proved that with probability at least $1 - \frac{1}{2^n}$, among the first $8n$ iterations of the `repeat-until` loop in protocol `Partition-with-CD` there must exist at least $n-1$ successful iterations. Thus, we have,

Theorem 2.1 *An n-station PRN with CD can be initialized in $O(n)$ rounds with probability at least $1 - \frac{1}{2^n}$.*

Now, consider a PRN with m stations, where $m \leq n$. For later reference, we now evaluate the probability that protocol `Initialization-with-CD` will take $O(n)$ broadcast rounds to initialize the PRN. Let Y be the random variable defined above. Then, it is clear that

$$\Pr[Y < m-1] < \Pr[Y < n-1] \leq e^{-\frac{9}{8}n} < 2^{-\frac{9}{8}n} < \frac{1}{2^n}.$$

Thus, we have the following result.

Corollary 2.2 *Protocol* `Initialization-with-CD` *initializes an m-station, ($m \leq n$) PRN with CD in $O(n)$ rounds with probability at least $1 - \frac{1}{2^n}$.*

2.2 Initializing the k-channel PRN with CD

The main purpose of this subsection is to present an efficient initialization protocol for the k-channel PRN. Our initialization protocol involves two stages as described next.

Protocol `Initialization-k-channel-PRN`

Stage 1 Each station selects one of the channels at random. Let P_i, ($1 \leq i \leq k$), denote the set of stations that selected channel C(i). For every i, ($1 \leq i \leq k$), dedicate channel C(i) to P_i and use the protocol `Initialization-with-CD` to initialize P_i independently. Let $p_{i,j}$, ($1 \leq i \leq k, 1 \leq j \leq |P_i|$), denote the j-th station in P_i. At the end of this stage, each station $p_{i,j}$ knows i and its local ID number j.

Stage 2 Compute the prefix-sums of $|P_1|, |P_2|, \ldots, |P_k|$. That is, compute for every i, ($1 \leq i \leq k$), the sum $|P_1|+|P_2|+\cdots+|P_i|$. Next, for every i, ($1 \leq i \leq k-1$), broadcast the value $|P_1|+|P_2|+\cdots+|P_i|$ on channel $C(i+1)$. Now, each station $p_{i,j}$, ($2 \leq i \leq k$), computes its ID whose value is $|P_1|+|P_2|+\cdots+|P_i|+j$.

It is worth noting that Stage 1 uses the randomized protocol developed in Subsection 2.1, while Stage 2 uses a deterministic protocol inspired by the well-known prefix sums algorithm for the PRAM [3]. We begin by evaluating the number of rounds involved in completing Stage 1. For this, we show that no set P_i is likely to contain too many stations.

Fix a channel and let X be the random variable denoting the number of stations that selected that channel. Clearly, $E[X] = \frac{n}{k}$. By using the Chernoff bound in (4) with $\epsilon = 1$, we can bound the probability $\Pr[X \geq \frac{2n}{k}]$ as follows:

$$\Pr[X \geq \frac{2n}{k}] \leq e^{-\frac{2n}{3k}}.$$

Thus, the probability that some channel is chosen by at least $\frac{2n}{k}$ stations is less than $k \cdot e^{-\frac{2n}{3k}}$. It follows that, with probability at least $1 - k \cdot e^{-\frac{2n}{3k}}$, every channel is chosen by at least $\frac{2n}{k}$ stations.

Let us assume that each of the k channels was selected by fewer than $\frac{2n}{k}$ stations. Corollary 2.2 guarantees that protocol `Initialization-with-CD` initializes an m-station, single-channel PRN, where $m \leq \frac{2n}{k}$, in $O(\frac{n}{k})$ rounds with probability at least $1 - e^{-\frac{9n}{4k}}$.

Note, however, that before the protocol can advance to Stage 2, it must check whether each of the k instances of `Initialization-with-CD` running in the k channels has finished executing. For this purpose, the execution of `Initialization-with-CD` is suspended every, say, 10 rounds. Using one broadcast in this suspended round, every station that has not yet been assigned a local ID number broadcasts on channel C(1). If the status of C(1) is NULL, we know that Stage 1 has ended and Stage 2 may begin.

The above discussion implies that with probability at least

$$1 - k \cdot e^{-\frac{2n}{3k}} - ke^{-\frac{9n}{4k}} \quad \geq \quad 1 - 2k \cdot e^{-\frac{2n}{3k}}$$

Stages 2 is ready to begin at the end of $O(\frac{n}{k})$ broadcast rounds.

Recall that at the end of Stage 1, every station $p_{i,1}$, ($1 \leq i \leq k$), knows the number $|P_i|$ of stations in P_i. Thus, Stage 2 can, in principle, use the stations $p_{i,1}$, ($1 \leq i \leq k$), to compute the prefix-sums of $|P_1|, |P_2|, \ldots, |P_k|$. If every set P_i, ($1 \leq i \leq k$), is non-empty, the prefix sums are computed by a trivial implementation of the prefix-sums algorithm for the PRAM [3]. However, in our case, some of the P_is may be empty and, consequently, some of the stations $p_{i,1}$ may not exist. Thus, we need to adapt the prefix-sums algorithm for the PRAM to work even if some of the stations do not exist.

The protocol to compute the prefix-sums is executed in a recursive manner. After the execution of the prefix-sums, for every i, ($1 \leq i \leq k$), the following conditions are satisfied: (ps1) Every existing station $p_{i,1}$, knows the prefix-sum $|P_1| + |P_2| + \cdots + |P_i|$, (ps2) A leader is elected in $P_1 \cup P_2 \cup \cdots \cup P_i$ if at least one of $P_1, P_2, \ldots, P_i$ is non-empty, and (ps3) The elected leader knows the value of the sum $|P_1| + |P_2| + \cdots + |P_i|$.

Our prefix-sums protocol is as follows. If $k = 1$, elect station $p_{i,1}$ as the leader of P_1. Clearly, station $p_{i,1}$ knows the prefix sum $|P_1|$ if it exists. Thus above conditions are satisfied.

If $k \geq 2$, partition $G = \{P_1, P_2, \ldots, P_k\}$ into two groups $G_1 = \{P_1, P_2, \ldots, P_{\frac{k}{2}}\}$ and $G_2 = \{P_{\frac{k}{2}+1}, P_{\frac{k}{2}+2}, \ldots, P_k\}$. Recursively compute the prefix sums in G_1 and G_2 using channels $C(1), C(2), \ldots, C(\frac{k}{2})$ and $C(\frac{k}{2}+1), C(\frac{k}{2}+2), \ldots, C(k)$, respectively.

By the inductive hypothesis, when these prefix sums protocols terminate the conditions (ps1)–(ps3) above are satisfied in both G_1 and G_2. The leader in G_1 broadcasts the value $S_1 = |P_1| + |P_2| + \cdots + |P_{\frac{k}{2}}|$ on channel $C(1)$. Every station in G_2 monitors channel $C(1)$ and updates the value of its prefix sum by adding the value broadcast to the value it stores (i.e its local prefix sum within G_2). It is easy to see that the value obtained corresponds to the correct prefix sum within G. Finally, the leader in G_2 broadcasts the sum $S_2 = |P_{\frac{k}{2}+1}| + |P_{\frac{k}{2}+2}| + \cdots + |P_k|$ on channel $C(1)$. The leader in G_1 adds together the sums S_1 and S_2 and is elected the leader of G. If no station belongs to G_1, every station in G_2 can detect it, because the status of $C(1)$ is NULL when the leader in G_1 broadcasts S_1. In this case, the leader of G_2 becomes that of G. Since the depth of recursion is $\lceil \log k \rceil$, the protocol performs $O(\log k)$ broadcast rounds.

After executing the prefix sums protocol, each station $p_{i,1}$ broadcasts the value $|P_1| + |P_2| + \cdots + |P_i|$ on channel $C(i+1)$. Every station $p_{i+1,j}$ monitors channel $C(i+1)$ and computes $|P_1| + |P_2| + \cdots + |P_i| + j - 1$, which is its ID number. Consequently, Stage 2 terminates in $O(\log k)$ rounds. Thus, we have proved the following result.

Theorem 2.3 *The k-channel PRN with CD can be initialized in $O(\frac{n}{k} + \log k)$ rounds with probability at least $1 - 2k \cdot e^{-\frac{2n}{3k}}$.*

Assuming $k \leq \frac{n}{3 \log n}$, we have

Corollary 2.4 *Even if the number n of stations is not known, the k-channel PRN with CD can be initialized in $O(\frac{n}{k})$ rounds with probability at least $1 - \frac{1}{n}$, whenever $k \leq \frac{n}{3 \log n}$.*

3 Initializing the PRN with no CD

3.1 Leader election on the PRN with no CD

This subsection presents a leader election protocol on the PRN with no CD. The details of the protocol follow.

```
Protocol Election-with-no-CD
for i ← 0 to ∞ do
  for j ← 0 to i do
    each station broadcasts on channel C(1) with
      probability 1/2^j;
    if the status of C(1) is SINGLE then the station
      broadcasting is declared the leader
    endif
  end for
end for
```

Let s be the unique integer satisfying $2^s \leq n < 2^{s+1}$. We say that a broadcast round is *good* if $j = s$, that is, if $2^j \leq n < 2^{j+1}$. A good round succeeds in finding a leader with probability at least

$$\binom{n}{1} \left(\frac{1}{2^s}\right)^1 \left(1 - \frac{1}{2^s}\right)^{n-1} \geq \left(1 - \frac{1}{2^s}\right)^{2^{s+1}-1} > \frac{1}{e^2}.$$

Thus, the first t good rounds fail with probability at most

$$\left(1 - \frac{1}{e^2}\right)^t \leq e^{-\frac{t}{e^2}}. \qquad \text{(since } 1 + x \leq e^x\text{)}$$

On the other hand, the first $s + t + 1$ iterations of the outer **for**-loop, corresponding to $i = 0, 1, \ldots, s, s+1, \ldots, s+t$ contain t good rounds. It follows that the first $1 + 2 + \cdots + (s + t + 1) = O((s+t)^2)$ broadcast rounds contain t good rounds. Since $s = \lfloor \log n \rfloor$ we have proved the following result.

Theorem 3.1 *Protocol* `Election-with-no-CD` *elects a leader in $O(t^2 + (\log n)^2)$ broadcast rounds with probability at least $1 - e^{-\frac{t}{e^2}}$.*

Choosing $t = e^2 \log n$ and $t = e^2 \sqrt{n}$, we obtain the following important result.

Corollary 3.2 *Protocol* `Election-with-no-CD` *elects a leader in $O((\log n)^2)$ broadcast rounds with probability at least $1 - \frac{1}{n}$ or in $O(n)$ broadcast rounds with probability at least $1 - \frac{1}{2^{\sqrt{n}}}$.*

3.2 Initializing the PRN with no CD

Suppose that the stations in a subset $\mathcal{P}$ of $\mathcal{S}$ broadcast on channel C(1). If the PRN can detect collisions, then every station can determine whether $|\mathcal{P}| = 0$, $|\mathcal{P}| = 1$, or $|\mathcal{P}| \geq 2$. On the other hand, if the PRN does not have this capability, then one can only determine whether $|\mathcal{P}| = 1$ or $|\mathcal{P}| \neq 1$.

However, once a leader is elected in $\mathcal{S}$, the PRN with no CD can simulate that with CD in $O(1)$ rounds. In other words, the PRN with no CD can determine whether $|\mathcal{P}| = 0$, $|\mathcal{P}| = 1$, or $|\mathcal{P}| \geq 2$. The details follow.

First, the leader p broadcast to all stations whether $p \in \mathcal{P}$ or not. After that, the following protocol is executed:

Case 1: $p \in \mathcal{P}$**.** Since $|\mathcal{P}| \geq 1$, it is sufficient to check whether $|\mathcal{P}| = 1$ or $|\mathcal{P}| \geq 2$. The stations in $\mathcal{P}$ broadcast on channel C(1). If the status of C(1) is SINGLE, then $|\mathcal{P}| = 1$, otherwise, it must be that $|\mathcal{P}| \geq 2$.

Case 2: $p \notin \mathcal{P}$**.** By mandating the stations in $\mathcal{P}$ to broadcast on channel C(1), we can determine if $|\mathcal{P}| = 1$ or $|\mathcal{P}| \neq 1$. Similarly, by mandating the stations in $\mathcal{P} \cup p$ to broadcast, we can determine if $|\mathcal{P}| = 0$ or $|\mathcal{P}| \neq 0$. These two results combined can determine whether $|\mathcal{P}| = 0$, $|\mathcal{P}| = 1$, or $|\mathcal{P}| \geq 2$.

Thus, three broadcast rounds are sufficient to simulate the PRN with CD if a leader is elected beforehand. By using the leader election protocol in the previous subsection, we have the following result.

Theorem 3.3 *A protocol running in T rounds on the single-channel PRN with CD can be simulated in $O(T + t^2 + (\log n)^2)$ rounds on the single-channel PRN with no CD with probability at least $1 - O(e^{-\frac{t}{e^2}})$.*

Theorems 2.1 and 3.3 combined imply that the single-channel n-station PRN with no CD can be initialized in $O(n + t^2 + (\log n)^2)$ rounds with probability at least $1 - O(\frac{1}{2^n} + e^{-\frac{t}{e^2}})$. Choosing $t = e^2\sqrt{n}$, we have the following

Corollary 3.4 *The single-channel, n-station PRN with no CD can be initialized in $O(n)$ broadcast rounds with probability at least $1 - \frac{1}{2\sqrt{n}}$.*

Next, we discuss an initialization protocol for the k-channel PRN with no CD. For this purpose, we will implement `Initialization-k-channel-PRN` on the k-channel PRN with no CD. Recall that `Initialization-k-channel-PRN` executes k instances of `Initialization-with-CD`, one in each of the k channels.

Instead, we execute the no CD version of `Initialization-with-CD` in Corollary 3.4. Thus, all the P_is, ($1 \leq i \leq k$), can be initialized in $O(\frac{n}{k})$ broadcast rounds with probability at least $1 - \frac{k}{2\sqrt{\frac{n}{k}}}$. Further, we need to check whether the initialization of all the P_is is finished. Similarly to the implementation on the PRN with CD, this can be done by suspending the initialization in every, say, 10 rounds. Using one broadcast round in this suspended round, we will check whether all stations are locally initialized.

For this purpose, we need to find a leader of P in advance. By Corollary 3.2, this can be done in $O((\log n)^2)$ rounds with probability at least $1 - \frac{1}{n}$. Consequently, Stage 1 of `Initialization-k-channel-PRN` can be implemented on the PRN with no CD in $O(\frac{n}{k} + (\log n)^2)$ broadcast rounds with probability at least $1 - O(\frac{k}{2\sqrt{\frac{n}{k}}} + \frac{1}{n})$.

The prefix-sums computation executed in Stage 2 involves no broadcast causing collision, it can also executed on the PRN with no CD in $O(\log k)$ rounds.

As a consequence, Stages 1 and 2 can be implemented on the PRN with no CD to run in $O(\frac{n}{k} + (\log n)^2 + \log k)$ rounds with probability $1 - O(\frac{k}{2\sqrt{\frac{n}{k}}} + \frac{1}{n})$. Assuming $k \leq \frac{n}{4(\log n)^2}$, we have

Theorem 3.5 *The k-channel PRN with no CD can be initialized in $O(\frac{n}{k})$ rounds with probability $1 - O(\frac{1}{n})$, whenever $k \leq \frac{n}{4(\log n)^2}$.*

References

[1] I. Chlamtac and S. Kutten, Tree-based broadcasting in multihop radio networks, *IEEE Transaction on Computers*, C-36, (1987), 1209–1223.

[2] R. Dechter and L. Kleinrock, Broadcast communication and distributed algorithms, *IEEE Transactions on Computers*, C-35, (1986), 210–219.

[3] J. JaJa. *An Introduction to Parallel Algorithms*. Addison-Wesley, 1992.

[4] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.

[5] K. Nakano, Optimal initializing algorithms for a reconfigurable mesh, *Journal of Parallel and Distributed Computing*, 24, (1995), 218-223.

[6] K. Nakano and S. Olariu, Randomized $O(\log \log n)$-round leader election protocols in Packet Radio Networks, *Proc. of International Symposium on Algorithms and Computation*, (LNCS 1533), 209–218, 1998.

[7] D. E. Willard, Log-logarithmic selection resolution protocols in a multiple access channel, *SIAM Journal on Computing*, 15, (1986), 468–477.