

A better lowerbound for distributed leader  
finding in bidirectional asynchronous rings  
of processors

Hans L. Bodlaender

RUU-CS-87-13

August 1987



Rijksuniversiteit Utrecht

**Vakgroep informatica**

Bodepastoor 6 3504 CD Utrecht  
Corr. adres: Postbus 80112 3508 TA Utrecht  
Telefoon 030-53 1454  
The Netherlands

**A better lowerbound for distributed leader  
finding in bidirectional asynchronous rings  
of processors**

**Hans L. Bodlaender**

**Technical Report RUU-CS-87-13**

**August 1987**

**Department of Computer Science  
University of Utrecht  
P.O. Box 80.012  
3508 TA Utrecht  
the Netherlands**



# A better lower bound for distributed leader finding in bidirectional asynchronous rings of processors\*

Hans L. Bodlaender

Dept. of Computer Science, University of Utrecht  
P.O. Box 80.012, 3508 TA Utrecht, the Netherlands

## Abstract

In this note we prove a lower bound of  $\frac{1}{2}nH_n \approx 0.346n \log n$  for the average number of messages for distributed leader finding in asynchronous, bidirectional rings of processors.

**Keywords:** distributed computing, lower bounds, leader finding on rings of processors

## 1 Introduction

In this note we consider the problem of finding a leader in an asynchronous, bidirectional ring of processors. Each processor is distinguished by a unique identification number. We assume that the size  $n$  of the ring is not known to the processors. There is no central controller. The problem is to design a distributed algorithm that “elects” a unique processor as leader, (e.g., the highest numbered processor,) using a minimum number of messages.

We assume that the processors work fully asynchronous and cannot use clocks or timeouts. As observed in [11], from this assumption it follows that we can assume that the algorithm is message-driven: except for the initialization phase of an election, a processor can only perform actions upon receipt of a message. We also assume that processors and the communication subsystem work error-free.

---

\*This research was done while the author was visiting the Laboratory for Computer Science of the Massachusetts Institute of Technology, with financial support by the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

Much work has been done in order to obtain good upper and lower bounds for the election problem. For unidirectional rings one has an upper bound of  $1.356n \log n + \mathcal{O}(n)$  messages worst case [5,12], and an exact bound of  $nH_n$  messages average case [4,11]. ( $H_n$  is the  $n$ 'th harmonic number, i.e.,  $H_n = \sum_{i=1}^n \frac{1}{i} \approx 0.69 \log n$ .) For bidirectional rings without a sense of orientation one has a worst case upper bound of  $1.44n \log n$  messages [10,13], and an average case upper bound of  $\frac{\sqrt{2}}{2}nH_n \approx 0.48n \log n$  messages [1,7,9].

For the lower bound results for bidirectional rings, one usually assumes that processors do have a sense of orientation, i.e., each processor has the same idea about "left" and "right". This only strengthens the results. The first lower bound for this problem was obtained by Burns [3], who obtained a worst case lower bound of  $\frac{1}{4}n \log n$  messages. Pachl, Korach and Rotem [11] proved a lower bound of  $\frac{1}{8}n \log n$  message for the average case.

Frederickson and Lynch [8] prove a lower bound of  $\frac{1}{2}n \log n + \mathcal{O}(n)$  on the worst case number of messages for *synchronous* bidirectional algorithms, that run in time, bounded by some constant  $t_n$  on all rings with size  $n$ . This lower bound is also valid for the asynchronous case: every algorithm that runs on an asynchronous ring can also run on a synchronous ring and the time that is used by the algorithm is bounded, for instance by the number of messages that is sent.

In this note we improve the lower bound of [11], and prove a lower bound of  $\frac{1}{2}nH_n \approx 0.346n \log n$  messages for the average number of messages for the leader finding problem in asynchronous, bidirectional rings, with sense of orientation. For this lower bound result, we will assume that the asynchronous algorithm works on a synchronous ring. In this way we lose the implicit non-determinism, introduced by the fact that we have to deal with *asynchronous, bidirectional* algorithms.

## 2 Main results

We give a number of definitions. Most of them can be found in [11]. We will assume that all identification numbers are chosen from  $Z$ .

For  $X \subseteq Z$ , we define  $D(X)$  to be the set of finite, non-empty sequences of distinct elements of  $X$ , i.e.,  $D(X) = \{\langle s_1 \dots s_k \rangle \mid k \geq 1; 1 \leq i \leq k \Rightarrow s_i \in X; i \neq j \Rightarrow s_i \neq s_j\}$ . We abbreviate  $D(Z) = D$ .

The concatenation of two strings  $s = \langle s_1 \dots s_k \rangle$  and  $t = \langle t_1 \dots t_l \rangle$  is denoted by  $s \cdot t = \langle s_1 \dots s_k t_1 \dots t_l \rangle$ .

A string  $s$  is said to be a prefix of  $t$ , (denoted by  $s \prec t$ ) if there is a

$v \in D \cup \{\epsilon\}$ , with  $t = sv$ . ( $\epsilon$  denotes the empty string.)

The  $l$ 'th element of a string  $s$  is denoted by  $s_l$ . The length of a string  $s = \langle s_1 \dots s_k \rangle$  is denoted by  $length(s) (= k)$ .

We denote  $D_k(X) = \{s \in D(X) \mid length(s) = k\}$ . For  $s \in D$ , the set of cyclic permutations of  $s$  is denoted by  $C(s)$ .

We say that a ring  $r$  is labeled with  $s = \langle id_1 \dots id_k \rangle \in D$  if the successive processors in  $r$  have identities successively  $id_1, \dots, id_k$ . (Note that it also follows that the size of  $r$  is  $k$ .)

Let  $A$  be an arbitrary bidirectional, asynchronous leader finding algorithm. We now will consider the behavior of  $A$ , when it works on a *synchronous* ring and all processors start the algorithm on the same moment. We assume that the algorithm can deal with the case that it receives messages of both its neighbors on the same moment. (If not, we can assume for instance that it always first processes the message received from the left neighbor, and then the message from the right neighbor.)

Note that we now have a deterministic behavior: executing  $A$  on rings with identical length, and identical labelings of the rings will give the same messages that are transmitted.

We ignore the time, necessary for internal computations in processors. We assume all processors start at time 0.

We now consider the variant of the problem, where after termination of the algorithm, not only a unique processor is designated as leader, but also each processor knows the identity of the leader. Note that this latter requirement costs at most  $n$  extra messages.

The following lemma follows directly from the fact that  $A$  is message-driven.

**Lemma 2.1** *If no processor sends a message on time  $k$ , then also no processor sends a message on time  $k + 1$ .*

**Lemma 2.2** *When  $A$  is executed on a ring with size  $n$ , and  $k \leq n/2 - 1$ , then there must be at least one processor that sends a message on time  $k$ .*

**Proof.** Suppose there exists a ring  $r$  with size  $n$ , such that no processor sends a message at time  $k \leq n/2 - 1$ . It follows from lemma 2.1 that no processor sends a message after time  $k$ . Suppose processor  $p$  is designated as leader. Let  $q$  be a processor with distance  $\lfloor n/2 \rfloor$  to  $p$ . By assumption,  $q$  must know the identity of  $p$  upon termination, i.e., at time  $k$ . Let  $r'$  be a ring, obtained from  $r$  by changing the identity of  $p$ . As  $q$  will receive the

same messages when  $A$  is executed on  $r$  or on  $r'$ , it will terminate incorrectly in at least one of these two executions. **Q.E.D.**

Now note that it can only depend on (a subset of) the  $2k + 1$  identities of the processors with distance at most  $k$  to a processor  $p$ , whether  $p$  will send (a) message(s) on time  $k$  or not and what message(s) it will send. (This can be proved easily with induction on  $k$ . The status of processor  $p$  just before time  $k$  depends on the id's of the processors with distance at most  $k - 1$  to  $p$  (induction hypothesis). It depends on the id's of the processors with distance at most  $k - 1$  to the neighbors of  $p$ , what, if any, messages  $p$  will receive at time  $k$  (use again the induction hypothesis). Hence, by induction, the result follows.)

**DEFINITION 2.1** Let  $k \in N$ .  $X_k = \{u \in D \mid \text{length}(u) = 2k + 1$ , and if we have a ring labeled by  $s \in D$ , and  $u \prec s$ , then the processor with identity  $u_{k+1}$  will send at least one message on time  $k\}$ .

From the above discussion, it follows that the total number of messages, sent on a ring  $r$ , labeled with  $s$  and with size  $n$ , on time  $k \leq n/2$ , is at least  $|\{v \in X_k \mid \exists w \in C(s) : v \prec w\}|$ . Now we proceed with an argument, which is similar to the argument in [11].

**Lemma 2.3** Let  $k \leq n/2 - 1$ . Then  $\forall s \in D : \text{length}(s) = 2k + 1 \Rightarrow C(s) \cap X_{2k+1} \neq \emptyset$ .

**Proof.** Consider a ring labeled with  $s$ . By lemma 2.2, there will be at least one message sent at time  $k$ . This message will correspond to an element in  $C(s) \cap X_{2k+1}$ . **Q.E.D.**

Consider a fixed ring size  $n$ , and all  $(n - 1)!$  rings with all identification numbers taken from some fixed subset  $X$  of  $Z$  with size  $n$ , i.e., we consider all rings with labelings in  $D_n(X)$ . (Labelings, which are a cyclic shift of each other, give the same ring, i.e., each ring corresponds with  $n$  elements of  $D_n(X)$ .) The total number of messages sent at time  $k$ , ( $k \leq n/2 - 1$ ) over all these rings is at least:

$$\frac{1}{n} \cdot \sum_{s \in D_n(X)} |\{v \in X_k \mid \exists w \in C(s) : v \prec w; \text{length}(v) = 2k + 1\}|.$$

Now note that:

$$\sum_{s \in D_n(X)} |\{v \in D(X) \mid \exists w \in C(s) : v \prec w; \text{length}(v) = 2k + 1\}| = n \cdot n!.$$

These  $n \cdot n!$  instances of strings  $v \in D(X)$  (with  $\exists w \in C(s) : v \prec w$ ;  $length(v) = 2k + 1$ , taken over all  $s \in D_n(X)$ ) can be grouped together in groups of  $2k + 1$  elements, such that each group consists of all cyclic permutations of one sequence. (Note that each string  $v \in D_{2k+1}(X)$  yields the same number of instances.) The number of groups now is  $(n \cdot n!)/(2k+1)$ . It follows from lemma 2.3 that each of these groups contains an element of  $X_{2k+1}$ . So it follows that

$$\sum_{s \in D_n(X)} |\{v \in X_k \mid \exists w \in C(s) : v \prec w; length(v) = 2k + 1\}| \geq \frac{n \cdot n!}{2k + 1}.$$

Hence the average number of messages, sent in a ring with size  $n$  is at least

$$\frac{1}{n} \cdot \frac{1}{(n-1)!} \sum_{k=1}^{\lfloor n/2 \rfloor - 1} \frac{n \cdot n!}{2k + 1} = n \cdot \sum_{k=1}^{\lfloor n/2 \rfloor - 1} \frac{1}{2k + 1} = \frac{1}{2} n H_n + \mathcal{O}(n).$$

**Theorem 2.4** *Every asynchronous, bidirectional leader finding algorithm will use at least  $\frac{1}{2}nH_n + \mathcal{O}(n)$  messages on the average, assuming the algorithm runs on a synchronous ring.*

### 3 Discussion

The given lower bound is of the type, where we average over all possible labelings of the processors with identity numbers, and where the delay time of the messages can be chosen by an “adversary”. Here we choose the delay time of all messages equal. The lower bound result for the average case on bidirectional rings of Pachl, Korach and Rotem [11] also is of the type, where an adversary may chose the delay times of messages, in order to have a large number of messages sent. It is an interesting open problem to find a lower bound for the average case number of messages in a bidirectional ring, where we also average over all possible delay times of all messages (where the delay time of a message is described by some given probability distribution). However, we remark that it is not difficult to modify the proof in section 2 in order to obtain a  $\frac{1}{2c}n \log n$  lower bound for the average number of messages sent, if the maximum and the minimum delay time of a message differ at most a constant factor  $c$ .

The same lowerbound can be obtained (with minor changes in the proof) for the average expected number of messages for *probabilistic* algorithms.

Very recently, Duris and Galil [6] obtained  $\Omega(n \log n)$  lower bounds for the average number of messages on rings, where *processors know the ring*



size. With a powerful new technique, these bounds were improved in [2] to  $\frac{1}{2}n \log n + \mathcal{O}(n)$  for the unidirectional case, and  $(\frac{1}{4} - \epsilon)nH_n$  for the bidirectional case.

## References

- [1] Bodlaender, H.L., *Distributed Computing, Structure and Complexity*, Ph.D. Thesis, Univ. of Utrecht, Utrecht, 1986.
- [2] Bodlaender, H.L., A new lower bound technique for distributed extrema finding on rings of processors, Techn. Rep. RUU-CS-87-11, Dept. of Comp. Science, Univ. of Utrecht, Utrecht, 1987. (To appear.)
- [3] Burns, J.E., A formal model for message passing systems, Techn. Rep. 91, Computer Science Dept., Indiana Univ., Bloomington, IN., 1980.
- [4] Chang, E., and R. Roberts, An improved algorithm for decentralized extrema-finding in circular configurations of processes, *C. ACM* 22 (1979) 281-283.
- [5] Dolev, D., M. Klawe, and N. Rodeh, An  $\mathcal{O}(n \log n)$  unidirectional distributed algorithm for extrema finding in a circle, *J. Algorithms* 3 (1982) 245-260.
- [6] Duris, P., and Z. Galil, Two lower bounds in asynchronous distributed computation, to appear in: proceedings FOCS '87, 1987.
- [7] Flajolet, P., personal communication, 1986.
- [8] Frederickson, G.N., and N.A. Lynch, Electing a leader in a synchronous ring, *J. ACM* 34 (1987) 98-115.
- [9] Korach, E., D. Rotem, and N. Santoro, A probabilistic algorithm for decentralized extrema-finding in a circular configuration of processors, Res. Rep. CS-81-19, Dept. of Computer Science, Univ. of Waterloo, Waterloo, 1981.
- [10] Moran, S., M. Shalom, and S. Zaks, An algorithm for distributed leader finding in bidirectional rings without common sense of direction, 1985.
- [11] Pachl, J., E. Korach, and D. Rotem, Lowerbounds for distributed maximum-finding algorithms, *J. ACM* 31 (1984) 905-918.

- [12] Peterson, G.L., An  $\mathcal{O}(n \log n)$  unidirectional algorithm for the circular extrema problem, *ACM Trans. Prog. Lang. & Syst.* 4 (1982) 758-762.
- [13] van Leeuwen, J., and R.B. Tan, An improved upperbound for decentralized extrema-finding in bidirectional rings of processors, Tech. Rep. RUU-CS-85-23, Dept. of Computer Science, Univ. of Utrecht, Utrecht, 1985.