TIME-MESSAGE TRADE-OFFS FOR THE WEAK UNISON PROBLEM

Amos Israeli, Evangelos Kranakis,
Danny Krizanc, and Nicola Santoro

TR-222, MARCH 1993

School of Computer Science, Carleton University
Ottawa, Canada, KIS 5B6

TIME-MESSAGE TRADE-OFFS FOR THE WEAK UNISON PROBLEM (EXTENDED ABSTRACT)

Amos Israeli[§¶]
(amos@ee.technion.ac.il)

Evangelos Kranakis[*†]
(kranakis@scs.carleton.ca)

Danny Krizanc[*]
(krizanc@scs.carleton.ca)

Nicola Santoro[*‡]
(santoro@scs.carleton.ca)

Abstract

A set of anonymous processors is interconnected forming a complete synchronous network with sense of direction. Weak unison is the problem where all processors want to enter the same state (in our case "wakeup" state) in the absence of a global start-up signal. As measure of complexity of the protocols considered we use the "bits" times "lag" measure, i.e. the total number of (wakeup) messages transmitted throughout the execution of the protocol times the number of steps which are sufficient in order for all the processors to wakeup. We study trade-offs in the complexity of such algorithms under several conditions on the behavior of the processors (oblivious, non-oblivious, balanced, etc) and provide tight upper and lower bounds on the *time* × *#messages* measure.

1980 Mathematics Subject Classification: 68Q99
CR Categories: C.2.1
Key Words and Phrases: Anonymous network, Balanced, Chordal rings, *t*-step protocol, Non-oblivious, Oblivious, Time-message complexity, Unbalanced, Unison, Wakeup protocol, Weak unison.
Carleton University, School of Computer Science: SCS-TR-222

[§] Technion, Department of Electrical Engineering, Haifa, 32000, Israel

[¶] Partially supported by NWO through NFI Project ALADDIN under contract number NF 62-376

[*] Carleton University, School of Computer Science, Ottawa, ON, K1S 5B6, Canada

[†] Research supported in part by NSERC grant #907002.

[‡] Research supported in part by NSERC grant #A2415.

1 Introduction

An important problem in the current distributed computing literature is the study of the performance of networks when the processors may awaken spontaneously. This paper is concerned with achieving weak unison for all the processors of a synchronous complete network with a sense of direction [12] in the absence of a global start-up signal. The processors have no distinct identities, but the links of the complete network are labeled. Any set of these processors may wake up at any time during the execution of the algorithm. By executing identical protocols and transmitting messages via the network links these "awakened" processors must wake up the entire network. The set of initial processors is arbitrary and may range in size from a single processor to the set of all processors. However, the important point is that the wakeup protocol sought should be such that

- regardless of the set of processors which are awakened by themselves, eventually all the processors in the network should wake up, and
- the number of messages times the number of steps required for the completion of the algorithm should be optimal.

A weak-unison protocol will specify what the action of each processor will be based on the previous history and the given data at that processor. This means that during the execution of the protocol a set of processors is specified to whom an arbitrary processor, say p, (which already has received wakeup message) should send wakeup messages. Assume that a set of processors is initially awakened. These processors will initiate wakeup messages and will each transmit them to a specified set of processors; in turn the recipient processors retransmit wakeup messages to a new set of processors, and so on until eventually all processors in the network wakeup.

In measuring the quality of the resulting protocol we see an interesting interplay between time required to wake up all the processors in the network and the total number of messages transmitted throughout the execution of the protocol. Thus, a protocol that is time- and message-efficient when many processors initiate wakeup messages may fail to be efficient when there is a single initiator. For example, consider the following protocol in an oriented N-node ring: an initiator sends wakeup message to its left and dies; if a processor is awakened by another processor then it sends wakeup message to its left and dies. It is easy to see that if there is a single initiator the time required to wakeup the whole network is N and the total number of messages is N. However, if all processors are initiators then it takes one time unit for the algorithm to execute and the total number of messages is N. It appears that a fair measure of complexity of this protocol is the time required for all processors to wakeup times the number of bits transmitted throughout the execution of the protocol. Thus the complexity for the former schedule is $\Theta(N^2)$ and for the latter is $\Theta(N)$. In general, we are interested in protocols that have the optimal overall complexity performance regardless of the schedule of initiators.

1.1 Definitions and notations

The set of processors is denoted by $\{0, 1, \ldots, N-1\}$. The network is synchronous and the processors anonymous. The links are labeled, i.e. for any pair $\{x, y\}$ of nodes we associate a label $\ell(\{x, y\})$ such that for any x, if $y \neq y'$ then $\ell(\{x, y\}) \neq \ell(\{x, y'\})$. The orientation we will assume in this paper is the following: for $x, y < N$, the label of the edge $\{x, y\}$ is the integer $y - x \bmod N$. This orientation is called in the literature [12] *sense of direction.*

During the execution of a protocol processors may attain one of three possible states: "dormant", "awake", and "dead". A processor in a dormant state may either wake up on its own or be awakened by another processor, transmit a wakeup message and enter either the awake or the dead state. A processor in the awake state may transmit a wakeup message and enter either the dead state or (again) the awake state. A processor in the dead state always remains in that state (i.e. never transmits a message). Initially (at the beginning of the execution of the protocol) all processors are in the dormant state.

The number of steps of a wakeup protocol on a given set of initiators is the difference between the time that all processors are awakened and the time the first processor wakes up. The number of messages transmitted in this protocol for a given set of initiators is the number of messages transmitted during this time interval. A protocol is called t-step if this difference never exceeds t regardless of the set of initiators.

An important property of a (correct) wakeup protocol is that all processors must eventually wake up regardless of the schedule (i.e. the configuration of processors which initiate wakeup messages). The nature of a protocol may be such that during its execution a processor may receive, for example, more than one message. In this case the action of the processor involved may or may not depend on its current state. We call the protocol oblivious if the action of each processor is independent of its current state. If for a given protocol, the size of the set to whom each processor transmits messages (during the execution of the protocol) is a constant independent of the processor then the protocol is called balanced.

We are interested in protocols P which minimize the $time \times \#messages$ complexity measure. More precisely, the problem we investigate is the following: "Determine the size of $time \times \#messages$ for various kinds of wakeup protocols, e.g. oblivious, non-oblivious, balanced, etc."

1.2 Results of the paper

The following table summarizes our bounds for protocols with arbitrary number of steps.

Type of Protocol	Complexity
Arbitrary	$\Omega(N \log\log N)$
Balanced	$\Omega(N \log N)$
Oblivious	$\Omega(N \log^2 N)$
Oblivious	$O(N \log^2 N)$

Clearly, lower bounds valid for arbitrary (respectively, balanced) protocols are also valid for balanced (respectively, oblivious) protocols. Thus our bounds are tight for oblivious protocols (theorems 12, 11 and corollary 13) with arbitrary number of steps. Moreover for such protocols the optimal wakeup algorithms are obtained on the hyper-ring architecture (see example 8).

We also make a detailed analysis of the complexity of t-step protocols and prove the following bounds.

Type of Protocol	# of Steps	Complexity
Arbitrary	t	$\Omega(tN^{2^t/(2^t-1)})$
Balanced	t	$\Omega(tN^{(t+1)/t})$
Oblivious	$t = dN^{1/d}/m$	$O(mdtN)$

The first column describes the type of protocol, the second column gives the number of steps needed by the protocol P, and the last column gives the corresponding $time \times \#messages$, complexity (henceforth refered to as complexity). As before we see that our bounds are tight for balanced protocols when the number of steps is a constant $t = O(1)$ independent of N. The denominator m in the number of steps of the oblivious case may be any integer $\leq N^{1/d}$, moreover it is easy to see that $O(mdtN) = O(d^2N^{(d+1)/d})$.

1.3 Related work

Even and Rajsbaum [5, 6] have studied the problem of initialization of computation in (synnchronous as well as asynchronous) distributed networks in the absence of a global start-up signal. They consider the unison problem and study the number of beats it takes for all the processors to be in "unison" (as if they all started the computation at the same time) from the time some processor wakes up. They have shown that a synchronous network of N processors can reach unison within $2N$ beats of the clock [5]. They also show how to achieve unison in $2N$ bits when the network is asynchronous by use of Awerbuch's synchronizer, provided that all local clocks have the same rate [6] (see also [7]). Gouda and Herman [9] present a solution to the unison problem for synchronous systems which is also stabilizing, i.e. the system is guaranteed to reach unison starting from any state. Arora, Dolev and Gouda [1] give a stabilizing solution which for an N-node system uses N registers of $2 \log N$ bits and is guaranteed to converge within N^2 triggers. Covreur, Francez and Gouda [3] give bounded as well as unbounded solutions to unison for asynchronous systems.

Attiya, Snir and Warmuth [2] consider the "processor synchronization problem" on anonymous, synchronous rings in order to reduce input collection and orientation algorithms to the case where all processors start simultaneously and solve this problem in $O(N \log N)$ messages. Fischer, Moran, Rudich and Taubenfeld [8] consider implementations of a related problem, the wakeup problem on a shared register. They give upper and lower bounds on the number of values of the shared register under various assumptions characterizing the resilience of the protocol.

2 Lower Bounds for Arbitrary Protocols

Next we consider arbitrary t-step protocols. Usually it is more difficult to prove lower bounds on such protocols because the transmission set of a processor may well depend on the input (e.g. the number of wakeup messages received by the processor). Before discussing the general lower bound we present a simpler result for 2-step protocols that better illustrates our proof technique.

2.1 2-step protocols

We will prove an $N^{4/3}$ lower bound for arbitrary 2-step protocols. First we prove the following lemma.

LEMMA 1 *Assume that $K \subseteq \{0,1,\ldots,N-1\}$ is a set of size k. There exists a set I of size $\geq \lfloor N/k^2 \rfloor$ such that the sets $K+x$, for $x \in I$, are pairwise disjoint.*

PROOF (OUTLINE) We construct the set I by induction. Assume we have constructed the first s elements $x_0 = 0, x_1, x_2, \ldots, x_{s-1}$ such that the disjointness condition is satisfied. We show how to find x_s. Assume on the contrary that for all $x \leq sk^2 + 1$,

$$\exists i < s(K + x \cap K + x_i \neq \emptyset).$$

Then for all $x \leq sk^2 + 1$ there exists an $i < s$ and $k_i, k_i' \in K$ such that $x = k_i - k_i' + x_i$. Clearly, there are $\leq k^2$ possible differences of elements of K; hence, the number of elements represented by these last equations is $\leq sk^2$, which is a contradiction. It follows that there exists an $x \leq sk^2 + 1$ (call this x, x_s) such that

$$\forall i < s(K + x_s \cap K + x_i = \emptyset),$$

as desired. ■

A set I of processors satisfying the disjointness condition for a set K is called a set of **independent initiators** for K.

Now we are ready to prove an $N^{4/3}$ lower bound on arbitrary 2-step protocols. We have the following theorem.

THEOREM 2 *$\Omega(N^{4/3})$ is a lower bound on the complexity of any 2-step wakeup protocol.*

PROOF (OUTLINE) Let us consider a 2-step wakeup protocol. Assume that the protocol is such that it wakes up the whole system under any schedule. By anonymity of the network all initiator processors must execute precisely the same instruction. Hence without loss of generality we may assume that processors initiating wakeup messages transmit to a set K of processors of size k. Now we consider several schedules for waking up the network and study their corresponding complexity.

IF ONLY ONE PROCESSOR WAKES UP:

The processor reaches a set K of k other processors. All these k processors are in the same state and must wake up the remaining $N-k-1$ processors. Assume that each of them transmits to a set K' of size k' (it must be the same set for all processors since all processors are in the same state). Since all processors must wakeup following this schedule we obtain

$$kk' \geq N-k-1. \tag{1}$$

IF ALL PROCESSORS WAKE UP:
The total number of messages transmitted will be

$$\geq Nk. \tag{2}$$

IF $\lfloor N/k^2 \rfloor$ INDEPENDENT INITIATORS WAKE UP:
Using the same argument as before, each (independent) initiator reaches k other processors. This wakes up a total of $\frac{N}{k^2}k = \frac{N}{k}$ processors. Since the initiators are independent all these $\frac{N}{k}$ processors are in the same state and will therefore each transmit k' wakeup messages. It follows that the total number of messages transmitted is

$$\frac{N}{k^2}kk' \geq \frac{N(N-k-1)}{k^2}. \tag{3}$$

The maximum of the quantities in (2), (3) represents a lower bound on the number of messages, namely

$$\max\left\{Nk, \frac{N(N-k-1)}{k^2}\right\} \geq N^{4/3}.$$

This proves the desired lower bound. ■

2.2 t-step protocols

Next we give a result on the more general case of multistep protocols. This result will require a generaliztion of lemma 1 regarding the number of independent initiators in an arbitrary wakeup protocol.. As before we can prove the following result.

LEMMA 3 *Assume that $K_1, K_2, \ldots, K_{t-1} \subseteq \{0,1,\ldots,N-1\}$ are sets of corresponding sizes $k_1, k_2, \ldots, k_{t-1}$. Then there exists a set I of size $\geq \lfloor N/(k_1 \cdots k_{t-1})^2 \rfloor$ such that the sets $K+x$, for $x \in I$, are pairwise disjoint, where $K = K_1 + K_2 + \cdots + K_{t-1}$.*

PROOF (OUTLINE) Consider the sum set $K_1 + K_2 + \cdots + K_{t-1}$ which has size at most $k_1 k_2 \cdots k_d$ and apply lemma 1. ■

This lemma has important implications for the complexity of arbitrary t-step wakeup protocols. We can prove the following theorem.

THEOREM 4 *$\Omega(tN^{2^t/(2^t-1)})$ is a lower bound on the complexity of any t-step wakeup protocol.*

PROOF (OUTLINE) Consider the case where only one processor is initiator of a wakeup message. Recall that the network is synchronous and the processors are anonymous. It follows that for each $i = 1, \ldots, t$ there is a set $K_i \subseteq \{0, 1, \ldots, N-1\}$ such that processors awakened at step $i-1$ transmit wakeup messages to all their neighbors labeled with labels in the set K_i. Let k_i be the size of the set K_i. However the initiator processor must wake up the whole network. This implies that

$$k_1 k_2 \cdots k_t \geq N. \tag{4}$$

Let $i = 1, \ldots, t$ be fixed and I_i be a set of independent initiators "for the first i steps of the protocol" of size $\frac{N}{(k_1 \cdots k_{i-1})^2}$. Such a set exists by Lemma 3 (applied to the sets $K_1, \ldots, K_{i-1}$). If this set of initiators transmits messages following the protocol we obtain the lower bound

$$\frac{N}{(k_1 \cdots k_{i-1})^2} k_1 \cdots k_{i-1} k_i = \frac{N}{k_1 \cdots k_{i-1}} k_i \tag{5}$$

We claim that inequalities (4) and (5) imply a lower bound $N^{2^t/(2^t-1)}$ on the number of messages. To prove this we argue as follows. If for some $i = 1, 2, \ldots, t$,

$$\frac{k_i}{k_1 \cdots k_{i-1}} \geq N^{1/(2^t-1)}$$

then the claim is proved. Assume on the contrary that for all $i = 1, 2, \ldots, t$ we have that

$$\frac{k_i}{k_1 \cdots k_{i-1}} < N^{1/(2^t-1)}$$

Using induction we show that $k_i < N^{2^{i-1}/(2^t-1)}$. Indeed,

$$\begin{aligned} k_i &< N^{1/(2^t-1)} k_1 \cdots k_{i-1} \\ &< N^{1/(2^t-1)} N^{1/(2^t-1)+2/(2^t-1)+\cdots+2^{i-2}/(2^t-1)} \\ &= N^{2^{i-1}/(2^t-1)}. \end{aligned}$$

However this easily contradicts inequality (4). Indeed,

$$k_1 \cdots k_t < N^{(2^0+2^1+\cdots+2^{t-1})/(2^{t-1}-1)} = N^{2^t/(2^{t-1}-1)} < N.$$

The previous proof rests on the fact that for all $i = 1, \ldots, t$ we have that $\frac{N}{(k_1 \cdots k_{i-1})^2} \geq 1$. If this is not true then we need to make some trivial adjustements to the proof. Let i be minimal such that $(k_1 \cdots k_{i-1})^2 > N$. It follows that $(k_1 \cdots k_j)^2 < N$, for $j = 1, \ldots, i-2$. For $j \leq i-2$ consider a set of $\frac{N}{(k_1 \cdots k_j)^2}$ independent initiators. As in the previous proof, the theorem would be proved if for some $j \leq i-2$, $\frac{k_{j+1}}{k_1 \cdots k_j} \geq N^{1/(2^t-1)}$. Hence, without loss of generality we may assume that for all $j \leq i-2$, $\frac{k_{j+1}}{k_1 \cdots k_j} < N^{1/(2^t-1)}$. It follows as before that $k_1 \cdots k_{i-1} < N^{2^{i-1}/(2^t-1)}$, $k_{j+1} < N^{1/(2^t-1)} k_1 \cdots k_j \leq N^{2^j/(2^t-1)}$. This implies that $k_1 \cdots k_{i-1} \leq N^{(2^{i-1}-1)/(2^t-1)} < N^{1/2}$, which contradicts $(k_1 \cdots k_{i-1})^2 > N$. ■

As a corollary we obtain the following result.

THEOREM 5 *The complexity of any wakeup protocol is* $\Omega(N \log\log N)$. ■

3 Lower Bounds for Balanced Protocols

In this section we give lower bounds which are valid only for balanced protocols. These are protocols for which each processor broadcasts wakeup messages to a set of fixed size. First we consider the case of balanced protocols with fixed number of steps. Note that since oblivious protocols are balanced the lower bound of theorem 6 is also valid for oblivious protocols. We can prove the following lower bound on the complexity; our proof takes advantage of the the limitations imposed on the problem by the topology.

THEOREM 6 *The complexity of every t-step, balanced wakeup protocol is* $\Omega(tN^{(t+1)/t})$.

PROOF (OUTLINE) Consider an arbitrary (non-oblivious) balanced t-step protocol such that in each iteration of the algorithm processors wake up exactly k other processors. If only one processor wakes up then the whole network must also wake up. Hence

$$N \leq k + k^2 + \cdots + k^t \leq 2k^t.$$

It follows that $k \geq \frac{1}{2^{1/t}} N^{1/t}$. If all processors wakeup then a lower bound on the number of wakeup messages is $\Omega(Nk) = \Omega(N^{(t+1)/t})$. Since the time required is t the required complexity is $\Omega(tN^{(t+1)/t})$. This proves the theorem. ■

As a corollary we obtain the following result.

THEOREM 7 *$\Omega(N \log N)$ is a lower bound on the complexity of any balanced, wakeup protocol.* ■

4 Lower Bounds for Oblivious Protocols

Here we derive a tight lower bound for the special class of oblivious protocols (of arbitrary number of steps). Many of the results on wakeup protocols can be described very naturally within the framework of chordal rings. For this reason next we give some useful notation and definitions on chordal rings.

4.1 Chordal rings

Let Z_N be the set of integers modulo N. Let S be an arbitrary subset of Z_N. The circulant graph $Z_N[S]$ has the nodes $0, 1, \ldots, N-1$ and an edge between the nodes x, y if and only if $x - y \bmod N \in S$. The name "circulant" arises from the fact that the adjacency matrix of $Z_N[S]$ is a circulant matrix.

Another class of graphs are the so-called chordal rings. These are the rings R_N on N nodes, say, such that the set of vertices is $\{0, 1, \ldots, N-1\}$ and there exists a set S such that two nodes x, y are adjacent if and only if $x - y \bmod N \in S$. We denote such a chordal ring by $R_N[S]$. We observe that the well-known ring R_N itself is the chordal ring $R_N[\emptyset]$ and that in general $R_N[S] = Z_N[S \cup \{1\}]$ (this is because the ring structure of R_N assumes the generator 1). Here are two examples of chordal rings [4, 10].

EXAMPLE 8 *For $N \leq k^n$ the chordal ring $R_N[k, k^2, \ldots, k^{n-1}]$ has diameter $\leq k \log_k N$ (use the fact that every $x < N$ can be represented in the basis k) and degree $\log_k N$. If $k = 2$ we call this network the hyper-ring.*

EXAMPLE 9 *For $N < n!$ the chordal ring $R_N[2!, 3!, \ldots, (n-1)!]$ has diameter $O(n^2)$ (use the fact that every $x < N$ can be represented in the mixed basis $1!, 2!, \ldots, (n-1)!$ as $x = x_1 + x_2 2! + \cdots + x_{n-1}(n-1)!$, with $0 \leq x_i \leq i$, for $i \geq 1$) and degree $n \leq \log N / \log\log N$.*

4.2 Lower bound

Our result is based on the following lemma which gives an $\Omega(\log^2 N)$ lower bound on the product of the diameter times the degree of arbitrary chordal rings. More precisely we have the following result.

LEMMA 10 *If the chordal ring $R_N[k_1, k_2, \ldots, k_{d-1}, k_d]$ has diameter δ then*

$$d \cdot \delta = \Omega\left(\log^2 N\right). \tag{6}$$

PROOF (OUTLINE) Let $x < N$ and suppose that s is the distance of x from 0 in the given chordal ring. A minimal length path connecting x to 0 consists of a_i edges each labeled with k_i, $i = 0, 1, \ldots, d$, where $s = a_0 + a_1 + \cdots + a_d$. It follows that every vertex $x < N$ of the chordal ring corresponds to a partition $i = a_0 + a_1 + \cdots + a_d$ of an integer $i \leq \delta$ into $\leq d+1$ parts; hence the number N of vertices of the chordal ring cannot exceed the number of partitions of an integer $i \leq \delta$ into $t+1$ parts. This latter number is majorized by

$$\sum_{i=1}^{\delta} \binom{i+d}{d}.$$

To see this, notice that the mapping

$$(a_0, a_1, a_2, \ldots, a_d) \rightarrow 1^{a_0}01^{a_1}01^{a_2}0\cdots01^{a_d}$$

is a $1-1$ correspondence between "partitions of i into d parts" and "d-element subsets of a set with $i+d$ elements". Using the well-known identity [11]

$$\sum_{r=0}^{m} \binom{m+r}{r} = \binom{n+m+1}{m}$$

it follows that

$$N \leq \binom{\delta+d+1}{\delta}. \tag{7}$$

However the right-hand side of (7) can be majorized using the inequality

$$\binom{x+y}{x} \leq 4^{\sqrt{xy}}. \tag{8}$$

This is is proved by induction on x, for all $y \leq x$. Indeed, it is trivially true for $x = 0$. Assume it is true for x. To prove it for $x + 1 \leq y$ we note that by the induction hypothesis

$$\binom{y+x+1}{x+1} = \frac{y+x+1}{x+1}\binom{y+x}{x} \leq \frac{y+x+1}{x+1} \cdot 4^{\sqrt{xy}}.$$

The right-hand side of this last inequality is easily shown to be $\leq 4^{\sqrt{(x+1)y}}$. Indeed, since $2^{\sqrt{y/(x+1)}} \leq 4^{\sqrt{(x+1)y}-\sqrt{xy}}$ it is enough to show that

$$1 + \frac{y}{x+1} = \frac{y+x+1}{x+1} \leq 2^{\sqrt{y/(x+1)}}.$$

This is equivalent to showing that $1+t \leq 2^{\sqrt{t}}$, for $0 < t \leq 1$, which in turn is equivalent to proving that $(1+t)^{\sqrt{t}} \leq 2^t$. But this is trivial since $0 < t \leq 1$. Hence inequality (8) is proved.

It is now clear that inequalities (7) and (8) imply $(\delta+1)d = \Omega(\lfloor \log N \rfloor^2)$, which in turn implies the lower bound stated in inequality (6). This proves the lemma. ■

THEOREM 11 $\Omega(N \log^2)$ *is a lower bound on the complexity of any oblivious, wakeup protocol.*

PROOF (OUTLINE) This follows easily from theorem 10. Since the protocol is oblivious and balanced every processor transmits a fixed number of messages in each iteration of the wakeup protocol, say d. The graph resulting from such a protocol is the chordal ring $R_N[K]$, where K is a set of size d. The time required for the wakeup message to reach all the processors is at least the diameter δ of the chordal ring $R_N[K]$. Eventually all N processors are awakened. Since the protocol is oblivious every processor that receives a wakeup message must transmit to all its d neighbors. Hence the number of messages transmitted during the execution of the protocol is Nd. It follows that the complexity is at least $Nd\delta = \Omega(N \log^2 N)$. ■

5 Upper Bounds

In this section we give wakeup algorithms and prove the upper bounds discussed in subsection 1.2. The protocols we consider here are oblivious. The main theorem is the following.

THEOREM 12

1. *For any d, if $N = k^d$, for some k, then there is an oblivious d-step wakeup protocol whose complexity is $O(d^2 N^{(d+1)/d})$.*

2. *More generally, for any d and any $m \leq N^{1/d}$, if $N = k^d$, for some k, then there is an oblivious $dN^{1/d}/m$-step wakeup protocol whose complexity is $O(mdtN) = O(d^2 N^{(d+1)/d})$.*

PROOF (OUTLINE) To prove the theorem we observe that we can view the t-dimensional mesh as a chordal ring $R_N[K]$, for some set K of links. For example, the 2-dimensional mesh is the chordal ring $R_N[\sqrt{N}]$, while the t-dimensional mesh is the chordal ring $R_N[N^{1/d}, N^{2/d}, \ldots, N^{(d-1)/d}]$. This indicates that wakeup algorithms can be implemented as follows.

First we consider the case of a d-step protocol. For each processor $p = (p_1, p_2, \ldots, p_d)$ let

$$K_p^i = \{(p_1, \ldots, p_{i-1}, x_i, p_{i+1}, \ldots, p_d) : x_i < N^{1/d}\}.$$

In this case if we define $K^i = \{(0, \ldots, 0, x_i, 0, \ldots, 0) : 0 \leq x_i < N^{1/d}\}$ then we see easily that $K_p^i = p + K^i$. Let $K_p = K_p^1 \cup \cdots \cup K_p^d$, and $K = K^1 \cup \cdots \cup K^d$. The protocol is such that processor p transmits wakeup wakeup messages to all processors in the set $p + K$. Formally the d-step protocol is as follows.

d-step Wakeup Algorithm
Algorithm for processor p:

1. If p is an initiator then it sends a wakeup message to all its neighbors in the set $p + K$ and dies.
2. If p receives a wakeup message from another processor and is not dead then it sends wakeup messages to all processors in the set $p + K$ and dies.

The size of each broadcast is $dN^{1/d}$. By definition of the protocol a broadcast from processor $p = (p_1, \ldots, p_d)$ will reach all processors of the form $p' = (p_1, \ldots, p_{i-1}, p_i', p_{i+1}, \ldots, p_d)$, where $0 \leq p_i' < N, i = 1, 2, \ldots, d$. Therefore it is clear that every processor will be reached after d steps. The complexity is easily seen to be as in the statement of the theorem.

The $t = dN^{1/d}/m$-step protocol is exactly as before. Each processor p transmits to the set $p + K$, where $K = K_1 \cup \cdots \cup K_d$ and $K_i = \{(0, \ldots, 0, x_i, 0, \ldots, 0) : 0 \leq x_i < m\}$. Details are left to the reader. ■

As an immediate corollary we obtain the special case of the hyper-ring. This is the chordal ring $R_N[2, 2^2, \ldots, 2^{n-1}]$ described in example 8 for $N = 2^n$.

COROLLARY 13 *There is an oblivious* $\log N$*-step wakeup protocol (implemented on the hyper-ring) whose complexity is* $O(N \log^2 N)$. ■

We observe that in view of the lower bound for oblivious protocols the result of corollary 13 is optimal. In addition, theorem 7 shows that the result of theorem 12 is also optimal for balanced wakeup protocols with a constant number of steps.

6 Conclusion

We have considered the problem of constructing efficient wakeup protocols on an anonymous, synchronous, complete network. We have constructed oblivious protocols with arbitrary number of steps and shown their optimality, i.e. complexity $\Theta(N \log^2 N)$. For balanced protocols we have given an $\Omega(N \log N)$ lower bound, while for arbitrary protocols an $\Omega(N \log\log N)$ lower bound. This leaves a $\log N$ (respectively, $\log^2 N / \log\log N$) gap for balanced (respectively, arbitrary) protocols from the optimal value for oblivious protocols.

Another interesting question concerns the gap between the $\Omega(N^{4/3})$ lower bound on the complexity of arbitrary 2-step wakeup protocols and the $O(N^{3/2})$ upper bound for oblivious 2-step protocols. A similar question applies to the corresponding lower bound $\Omega(tN^{2^t/(2^t-1)})$ for arbitrary t-step protocols and the upper bound $O(t^2 N^{(t+1)/t})$ for oblivious t-step protocols.

Acknowledgements

Many thanks to J. D. Dixon for suggesting inequality (8) and B. Mans and R. B. Tan for useful conversations.

References

[1] A. Arora, S. Dolev and M. Gouda, "Maintaining Digital Clocks in Step", preprint.

[2] H. Attiya and M. Snir and M. Warmuth, "Computing on an Anonymous Ring", Journal of the ACM, 35 (4), 1988. (Short version has appeared in proceedings of the 4th Annual ACM Symposium on Principles of Distributed Computation, 1985, 845 - 875.)

[3] J.-M. Couvreur, N. Francez and M. Gouda, "Asynchronous Unison", preprint.

[4] P. J. Davis, "Circulant Matrices", John Wiley and Sons, 1979.

[5] S. Even and S. Rajsbaum, "Unison in Distributed Networks", in "Sequences, Combinatorics, Compression, Security and Transmission", R. M. Capocelli, ed., Advanced Workshop held in Naples, Italy, June 6 - 11, 1988, Springer Verlag.

[6] S. Even and S. Rajsbaum, "Lack of a Global Clock Does Not Slow Down the Computation in Distributed Networks", TR522, Computer Science Department, Technion, 1988.

[7] S. Even and S. Rajsbaum, "The Use of Synchronizer Yields Maximum Computation Rate in Distributed Networks", STOC 1990, pages 95 - 105.

[8] M. J. Fischer, S. Moran, S. Rudich and G. Taubenfeld, "The Wakeup Problem", STOC 1990, pages 106 - 116.

[9] M. Gouda and T. Herman, "Stabilizing Unison", Information Processing Letters, Vol. 35, No. 4, pages 171 - 175, 1990.

[10] Ki Hang Kim, "Boolean Matrix Theory and Applications", Marcel Dekker Inc., New York, 1982.

[11] D. Knuth, "The Art Computer Programming: Fundamental Algorithms", Addison Wesley, 1973.

[12] N. Santoro, "Sense of Direction, Topological Awareness and Communication Complexity", ACM SIGACT News, Number 16, pages 50 - 56, 1984.

School of Computer Science, Carleton University
Recent Technical Reports

TR-179 **Parallel Algorithms for Determining K-width-Connectivity in Binary Images**
Frank Dehne and Susanne E. Hambrusch, September 1990

TR-180 **A Workbench for Computational Geometry (WOCG)**
P. Epstein, A. Knight, J. May, T. Nguyen, and J.-R. Sack, September 1990

TR-181 **Adaptive Linear List Reorganization under a Generalized Query System**
R.S. Valiveti, B.J. Oommen and J.R. Zgierski, October 1990

TR-182 **Breaking Substitution Cyphers using Stochastic Automata**
B.J. Oommen and J.R. Zgierski, October 1990

TR-183 **A New Algorithm for Testing the Regularity of a Permutation Group**
V. Acciaro and M.D. Atkinson, November 1990

TR-184 **Generating Binary Trees at Random**
M.D. Atkinson and J.-R. Sack, December 1990

TR-185 **Uniform Generation of Combinatorial Objects in Parallel**
M.D. Atkinson and J.-R. Sack, January 1991

TR-186 **Reduced Constants for Simple Cycle Graph Separation**
Hristo N. Djidjev and Shankar M. Venkatesan, February 1991

TR-187 **Multisearch Techniques for Implementing Data Structures on a Mesh-Connected Computer**
Mikhail J. Atallah, Frank Dehne, Russ Miller, Andrew Rau-Chaplin, and Jyh-Jong Tsay, February 1991

TR-188 **Generating and Sorting Jordan Sequences**
Alan Knight and Jörg-Rüdiger Sack, March 1991

TR-189 **Probabilistic Estimation of Damage from Fire Spread**
Charles C. Colbourn, Louis D. Nel, T.B. Boffey and D.F. Yates, April 1991

TR-190 **Coordinators: A Mechanism for Monitoring and Controlling Interactions Between Groups of Objects**
Wilf R. LaLonde, Paul White, and Kevin McGuire, April 1991

TR-191 **Towards Decomposable, Reusable Smalltalk Windows**
Kevin McGuire, Paul White, and Wilf R. LaLonde, April 1991

TR-192 **PARASOL: A Simulator for Distributed and/or Parallel Systems**
John E. Neilson, May 1991

TR-193 **Realizing a Spatial Topological Data Model in a Relational Database Management System**
Ekow J. Otoo and M.M. Allam, August 1991

TR-194 **String Editing with Substitution, Insertion, Deletion, Squashing and Expansion Operations**
B John Oommen, September 1991

TR-195 **The Expressiveness of Silence: Optimal Algorithms for Synchronous Communication of Information**
Una-May O'Reilly and Nicola Santoro, October 1991

TR-196 **Lights, Walls and Bricks**
J. Czyzowicz, E. Rivera-Campo, N. Santoro, J. Urrutia and J. Zaks, October 1991

TR-197 **A Brief Survey of Art Gallery Problems in Integer Lattice Systems**
Evangelos Kranakis and Michel Pocchiola, November 1991

TR-198 **On Reconfigurability of Systolic Arrays**
Amiya Nayak, Nicola Santoro, and Richard Tan, November 1991

TR-199 **Constrained Tree Editing**
B. John Oommen and William Lee, December 1991

TR-200 **Industry and Academic Links in Local Economic Development: A Tale of Two Cities**
Helen Lawton Smith and Michael Atkinson, January 1992

TR-201 **Computational Geometry on Analog Neural Circuits**
Frank Dehne, Boris Flach, Jörg-Rüdiger Sack, Natana Valiveti, January 1992

TR-202 **Efficient Construction of Catastrophic Patterns for VLSI Reconfigurable Arrays**
Amiya Nayak, Linda Pagli, Nicola Santoro, February 1992

TR-203 **Numeric Similarity and Dissimilarity Measures Between Two Trees**
B. John Oommen and William Lee, February 1992

TR-204 **Recognition of Catastrophic Faults in Reconfigurable Arrays with Arbitrary Link Redundancy**
Amiya Nayak, Linda Pagli, Nicola Santoro, March 1992

TR-205 **The Permutational Power of a Priority Queue**
M.D. Atkinson and Murali Thiyagarajah, April 1992

TR-206 **Enumeration Problems Relating to Dirichlet's Theorem**
Evangelos Kranakis and Michel Pocchiola, April 1992

TR-207 **Distributed Computing on Anonymous Hypercubes with Faulty Components**
Evangelos Kranakis and Nicola Santoro, April 1992

TR-208 **Fast Learning Automaton-Based Image Examination and Retrieval**
B. John Oommen and Chris Fothergill, June 1992

TR-209 **On Generating Random Intervals and Hyperrectangles**
Luc Devroye, Peter Epstein and Jörg-Rüdiger Sack, July 1992

TR-210 **Sorting Permutations with Networks of Stacks**
M.D. Atkinson, August 1992

TR-211 **Generating Triangulations at Random**
Peter Epstein and Jörg-Rüdiger Sack, August 1992

TR-212 **Algorithms for Asymptotically Optimal Contained Rectangles and Triangles**
Evangelos Kranakis and Emran Rafique, September 1992

TR-213 **Parallel Algorithms for Rectilinear Link Distance Problems**
Andrzej Lingas, Anil Maheshwari and Jörg-Rüdiger Sack, September 1992

TR-214 **Camera Placement in Integer Lattices**
Evangelos Kranakis and Michel Pocchiola, October 1992

TR-215 **Labeled Versus Unlabeled Distributed Cayley Networks**
Evangelos Kranakis and Danny Krizanc, November 1992

TR-216 **Scalable Parallel Geometric Algorithms for Coarse Grained Multicomputers**
Frank Dehne, Andreas Fabri and Andrew Rau-Chaplin, November 1992

TR-217 **Indexing on Spherical Surfaces Using Semi-Quadcodes**
Ekow J. Otoo and Hongwen Zhu, December 1992

TR-218 **A Time-Randomness Tradeoff for Selection in Parallel**
Danny Krizanc, February 1993

TR-219 **Three Algorithms for Selection on the Reconfigurable Mesh**
Dipak Pravin Doctor and Danny Krizanc, February 1993

TR-220 **On Multi-label Linear Interval Routing Schemes**
Evangelos Kranakis, Danny Krizanc, and S.S. Ravi, March 1993

TR-221 **Note on Systems of Polynomial Equations over Finite Fields**
Vincenzo Acciaro, March 1993