Optimal Distributed Algorithms in Unlabelled Tori and Chordal Rings[*][†]

Bernard Mans[‡]
(bmans@mpce.mq.edu.au)

[*]This research supported by Macquarie University Research Grant "Structural Information in Interconnection Networks".

[†]This paper is to appear in **Journal of Parallel and Distributed Computing**. Some of the results of this work have been presented at the 3rd International Colloquium on Structural Information and Communication Complexity (SIROCCO '96), Certosa di Pontignano (Siena), Italy, June 6-8, 1996.

[‡]Dept. of Computing, School of MPCE, Macquarie University, Sydney, NSW 2109, Australia.

Unlabelled Tori and Chordal Rings

Author to contact:

Bernard Mans,
Dept. of Computing,
School of Mathematics, Physics, Computing and Electronics,
Macquarie University,
Sydney, NSW 2109,
Australia.

Phone: 61 02 9850 9574
Fax: 61 02 9850 9551
E-mail: bmans@mpce.mq.edu.au

Abstract

We study the message complexity of distributed algorithms in Tori and Chordal Rings when the communication links are unlabelled, which implies that the processors do not have "Sense of Direction".

We introduce the paradigm of *handrail* which allows messages to travel with a consistent direction. We give a distributed algorithm which confirms the conjecture that the Leader Election problem for unlabelled Tori of N processors can be solved using $\Theta(N)$ messages instead of $O(N \log N)$. Using the same *handrail* paradigm, we solve the Election problem using $\Theta(N)$ messages in unlabelled chordal rings with one chord (of length approximately $\sqrt{N}$). This solves a long-standing open problem of the minimal number of unlabelled chords required to decrease to decrease the $O(N \log N)$ message complexity.

For each topology, we give an algorithm to compute the Sense of Direction in $\Theta(N)$ messages (improving the $O(N \log N)$ previous results). This proves the more fundamental result that any global distributed algorithm for these labelled topologies can be used with a similar asymptotic complexity in the respective unlabelled class.

1 Introduction

One of the main themes of investigation in distributed computation concerns the identification of the factors which are significant for the computability and the communication complexity of problems. Recently, it has become clear that an important role is played by Structural Information; that is, a priori knowledge available to the entities about the structure of the system.

In particular, *Sense of Direction* (denoted as $\mathcal{SD}$) has been identified as being significant for the computability and the communication complexity [8]. Informally, $\mathcal{SD}$ refers to the capability of a processor to distinguish between adjacent communication links, according to some globally consistent scheme [20]. More formally, it represents the ability of the processors to adequately combine the relationship between: the labelling, the topological structure, and the local view that an entity has of the system [7].

In this paper (extended from [14]), we address the message complexity of distributed algorithms in tori and in chordal rings when the communication links are unlabelled, which implies that processors do not have a globally consistent labelling of the communication links. They have no "Sense of Direction" but have only a topological awareness.

We first study the impact of structural information limited to a topological awareness on the **Leader Election** problem (the problem of moving the system from an initial situation where the nodes are in the same computational state, to a final situation where exactly one node is in a distinguished computational state, called *leader* and all the others are in the same state, called *defeated*). The election process may be independently started by any subset of the processors. It is assumed that every processor P_i has a distinct identity id_i chosen from some infinite totally ordered set ID, and is only aware of its own identity (in particular, it does not know those of its neighbours).

The leader election is a widely used solution for distributed algorithms requiring one process to act as a coordinator, initiator, sequencer, or otherwise to perform some special role. The election problem occurs, for instance, in token-passing when the token is lost or the owner has failed; in such a case, the remaining processors elect a leader to issue a new token. Several other problems encountered in distributed systems can be solved by election; for example: *crash recovery* (a new server should be found to continue the service when the previous server has crashed), *mutual exclusion* (where values for election can be defined as the last time the process entered the critical section), *group server* (where the choice of a server for an incoming request is made through an election between all the available servers managing a replicated resource), etc.

The lack of Sense of Direction is known to increase the message complexity of the Election problem. For instance, in arbitrary networks, the Election problem requires $\Theta(e + N \ \log N)$ messages [9], instead of $\Theta(N \ \log N)$ [15]. In the complete graph, the complexity increases from $\Theta(N)$ [13] to $\Theta(N \log N)$ messages [11]. In the torus, an $O(N)$ messages algorithm has been given [18] with Sense of Direction. It was conjectured that the problem can be solved with linear complexity when Sense of Direction is not available.

For chordal rings, the main open problem was to determine the minimal sets of links that must be added to the ring (which requires $\Theta(N \log N)$ messages) in order to achieve a linear election algorithm. Along the years, a succession of papers [2, 10, 17] finally lead to prove that a unique chord was sufficient [22], but without Sense of direction no result better than $O(N \log N)$ messages was known.

In section 2, we give a distributed algorithm which confirms the conjecture suggested in [18] that the Leader Election problem for unlabelled tori of N processors can be solved using $\Theta(N)$ messages instead of $O(N \log N)$. In section 3, we solve the Election problem using a $\Theta(N)$ message complexity for the unlabelled chordal ring with one chord (of length approximately $\sqrt{N}$). Finally, in section 4, we give a distributed algorithm to compute the Sense of Direction in $\Theta(N)$ messages for each topology, improving the $O(N \log N)$ previous results, and section 5 concludes by showing the relationship of solutions between labelled and unlabelled networks.

2 Distributed Algorithms for Unlabelled Tori

The labelled square torus of size N is a two-dimensional (wrap-around) array of $\sqrt{N} \times \sqrt{N}$ processors. Each processor is labelled $P_{i,j}$ with $0 \le i, j \le (\sqrt{N} - 1)$ and is connected by a bidirectional link to four processors: $P_{i,j+1 \bmod \sqrt{N}}$, $P_{i,j-1 \bmod \sqrt{N}}$, $P_{i+1 \bmod \sqrt{N},j}$, and $P_{i-1 \bmod \sqrt{N},j}$, through a link labelled *east*, *west*, *north*, and *south* respectively.

The same construction can extend to rectangular tori and higher dimensions. In the following, unless specified, we will denoted torus any $\sqrt{N} \times \sqrt{N}$ processors topology with the previous definition.

2.1 Sense of Direction in tori

Before proceeding with the unlabelled torus, we provide an intuitive description of the fundamental properties used as sense of direction in labelled tori.

For example, consider a portion of a torus depicted in Figure 1, the communication topology is a 2-dimensional torus where the edge labels $\{north, south, east, west\}$ are assigned in the natural globally consistent way. This labelling is a Sense of Direction [7]. Consider for instance the three paths, starting from node O, whose associated labels are $c_1 = [north, east, south, north]$, $c_2 = [east, north, west, east]$, and $c_3 = [east, east]$. Using the rules of the globally consistent labelling, it is trivial to deduce that the two paths corresponding to c_1 and c_2 will end in the same node NE, while the one corresponding to c_3 will end in a different node EE. With Sense of Direction, there is a relationship between labelling and capability of distinguishing among paths. Intuitively, the labelling is a Sense of Direction if it is possible to understand, from the labels associated to the edges, whether different paths from any given node x end in the same node or in different nodes.

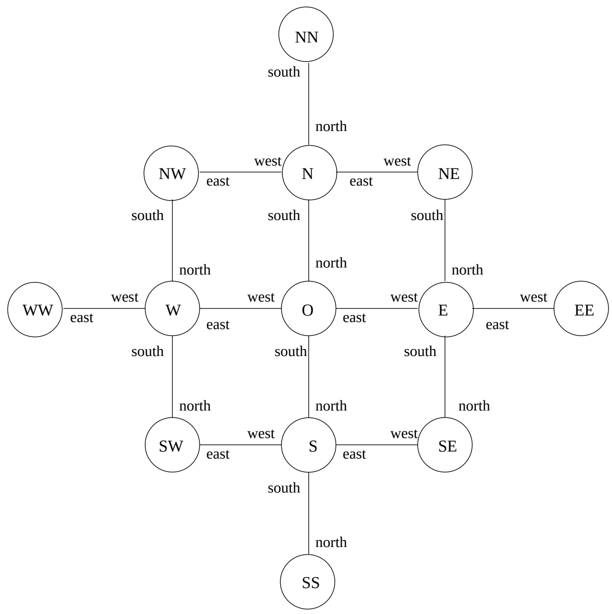

Figure 1: Sense of Direction in a portion of a Torus.

2.2 Electing a Leader in a torus with $\mathcal{SD}$

We briefly recall the main features of Peterson's algorithm [18] on square bidirectional tori. In its preliminary definition, all processors are assumed to be initially active and then process in phases. The number of active processes at each phase is reduced by a constant factor (but more than half since we must avoid an $O(N \log N)$ message complexity).

The basic goal of each active processor on the ith phase is to mark off the boundary of a square distance d on a side ($d = \alpha^i$ for some constant α). This is done by sending a "looking" message at distance d to the *north* then d to the *east*, d *south*, and finally d *west* to the original node (see Figure 2). Marking its boundaries, an active processor can *see* or *been seen* by an other active processor. When the "looking" message encounters the boundary of a processor on the same phase, the message either continues its way after becoming a "SawSmaller" (if the value of the encountered processor is smaller), or becomes a "SeenbySmaller" and continues along the boundary of the encountered processor with a larger value. The active processor is promoted to the next phase $i + 1$ iff either (a) it receives both a "SawSmaller" and a "SeenbySmaller" message, or (b) it did not see an active processor. The efficiency of the algorithm is based on the choice of the size of the search area which must be sufficiently large to cross another square.

The algorithm will terminate when a processor will reach phase $d = \alpha^i \geq \sqrt{N}$, at this time there are at most $(2 - \alpha^2)^{-1}$ active processors surviving. To elect the leader among the remaining processors, a constant number of "wrap-around" phases (sending a

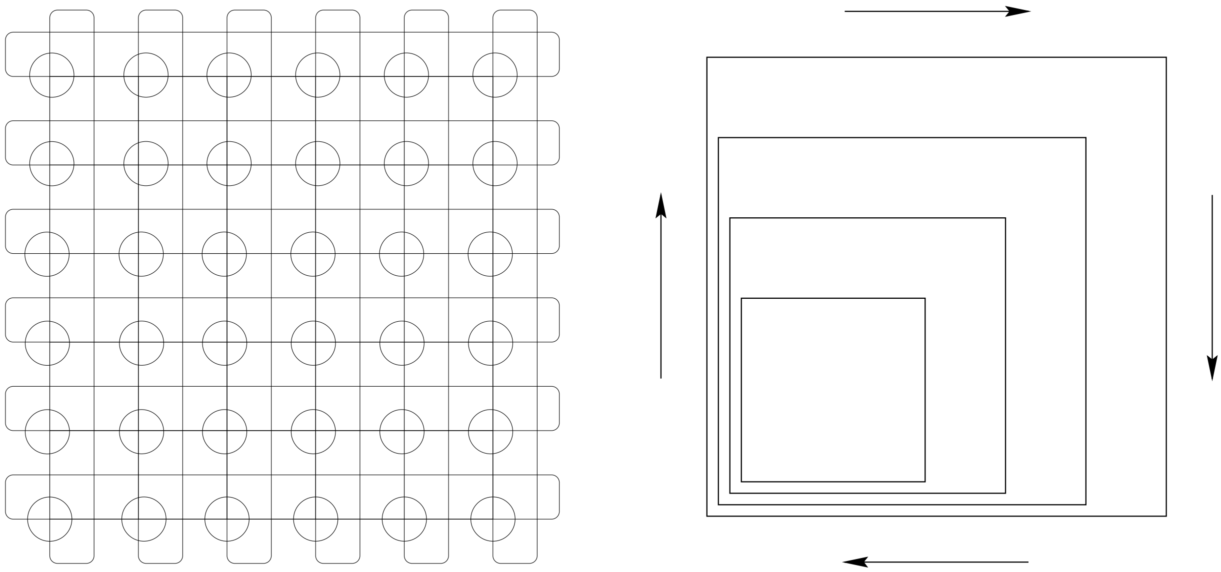

Figure 2: Leader Election Steps (right) in a Torus (left).

"probe" message in one dimension). Each costs $O(\sqrt{N})$ messages. The optimal constant is $\alpha = 1.1795$.

Also, it is conjectured that the algorithm can easily extend to higher dimensions. However, if the torus is not square but rectangular with length l and width w ($l \leq w$), then the algorithm can be adapted to use $\Theta(n + l \log l/w)$ messages.

2.3 Electing a Leader in a torus without $\mathcal{SD}$

When links are unlabelled, Peterson [18] suggested that the same algorithm can be used as is since "the algorithm only needs to mark off a square, the orientation of the square is irrelevant". It is only required to pass a message in straight line or make the "appropriate" turn. To confirm the conjecture we introduce the detailed modifications. We first provide a message with the ability to be forwarded in the same direction (*east/west*, *north/south*), and then introduce the notion of *handrail*, a partial structural information, which allows messages to turn consistently (clockwise or counterclockwise).

Definition 2.1 *(a) A* district labelling *of x is a function which maps the sequences of labels associated to the paths from x to any node y at distance at most 2, to the local name $\beta_x(y)$ used by x to refer to y.*
(b) A district labelling associated to β_x for x is consistent *iff $\forall y, z$ at distance at most 2 from x, $\beta_x(y) = \beta_x(z)$ implies that $y = z$.*

procedure district labelling (MyId)
begin
Mystatus := Asleep; $count = 0$
$\forall$ link r, $H1_r := \emptyset$, $H2_r := \emptyset$

If Mystatus = Asleep:
(0) Upon receipt of WAKEUP or any other message on any link
  Mystatus := Awake
  $\forall$ link k, send ONE(MyId) on arc labelled k
  if not (WAKEUP) process message as awake
end WAKEUP

If Mystatus = Awake:

(1) Upon receipt of ONE($Id1$) on link r
  $H1_r := Id1$
  $Id2 := Id1$; $Id1 := MyId$
  $\forall$ link $k \neq r$ send TWO($Id1, Id2$) on link labelled k
  $count = count + 1$
end ONE

(2) Upon receipt of TWO($Id1, Id2$) on link r
  $H2_r := H2_r \cup \{Id2\}$
  $count = count + 1$
  if $count = 16$ then Terminate
end TWO
end district labelling

Figure 3: District Labelling Algorithm in an Unlabelled Torus.

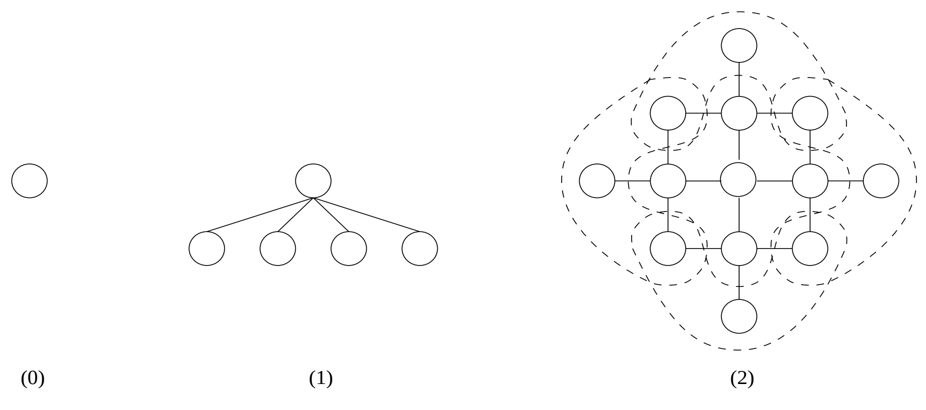

Figure 4: View of a Processor at distance 0 to 2 in an Unlabelled Torus.

Lemma 2.1 *The algorithm presented in Figure 3 computes a consistent district labelling for every node using exactly* $16N$ *messages in unlabelled square tori of* N *nodes with* $\sqrt{N} > 4$.

Proof An arbitrary subset of processors can spontaneously start the execution of the algorithm; this is modelled by the reception of a WAKEUP message.

The goal of the algorithm is to acquire the identity and the position of each processor at distance 2. Each processor sends its identity to each of its neighbours, which forwards it to its three remaining neighbours. Overall, each processor has received 4 messages for its immediate neighbours and 4×3 for nodes at distances 2, that is $16N$ messages overall. Note that this algorithm suggested in [18] is similar to the pre-labelling used in [22] for computing $\mathcal{SD}$ in $O(N \log N)$, and has been independently discovered in [16].

For the sake of the explanation, and without loss of generality, we applied the algorithm in the portion of a torus depicted in Figure 1, although the edge labelling does not contain any particular information in this case (it is only a local naming). When the algorithm terminates, the node O knows:

- that N, E, S, W are its immediate neighbours,
- which link must be used to reach them, respectively $north, east, south, west$,
- for each link, which node at distance 2 can be reached, $H2_{north} = \{NW, NN, NE\}$, $H2_{east} = \{NE, EE, SE\}$, $H2_{south} = \{SE, SS, SW\}$, $H2_{west} = \{SW, WW, NW\}$ respectively.
- (if $\sqrt{N} > 4$), that two sets $H2_i$ and $H2_j$ which intersection is empty are in the same dimension ($north/south$ or $east/west$), perpendicular if not.

To sum it up, the corresponding figure 1 is deduced by O (as shown in Figure 4). □

The restricted cases of electing a Leader in unlabelled tori in $O(N)$ messages when $\sqrt{N} \leq 4$ are simple to design and are not presented here for brevity. It is worth noting that the consistent district labelling algorithm does not need to be started simultaneously at all nodes, it can be initiated by any subset of processors (each sleeping processor will be awaken by the first incoming message).

By construction of the consistent district labelling in Lemma 2.1, a message can be forwarded easily in the same dimension ("straightforward"): the message received on link i is sent through the link j which has an empty intersection set, $H2_i \cap H2_j = \emptyset$. By contradiction a "turn" can be easily done too ($H2_i \cap H2_j \neq \emptyset$), but the exact direction is unknown.

With the next Lemma, we show that if the message carries a specific information, a *handrail*, it is possible to make the appropriate turn. The processors do not have sense of direction but the message forwarded has a globally consistent orientation (thanks to a

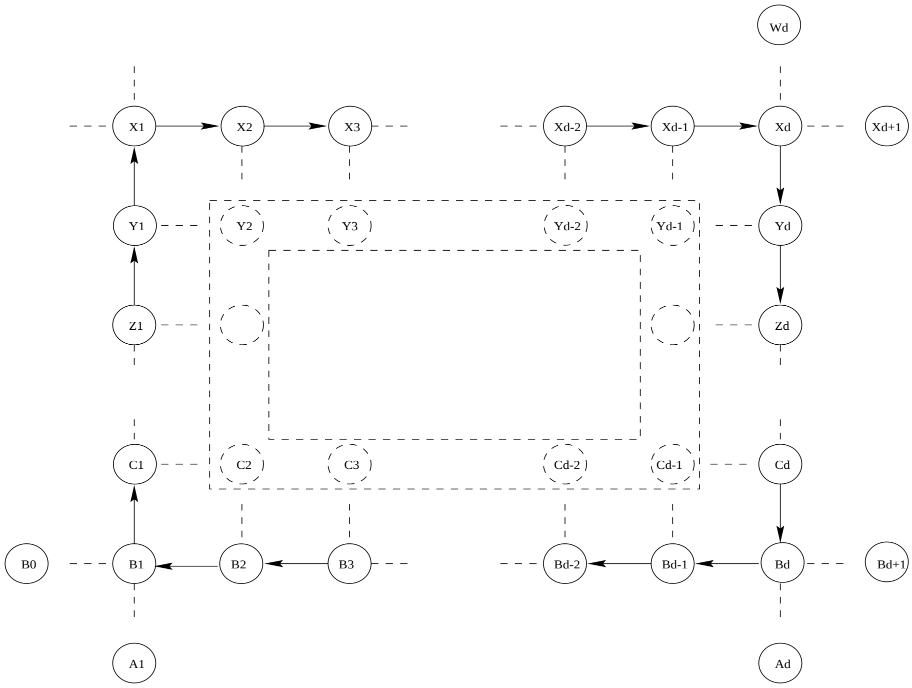

Figure 5: Handrail in the Unlabelled Torus.

handrail). Indeed, since the orientation of the square is irrelevant for the correctness of the algorithm, the choice of the first direction and the first "turn" does not matter for the correctness, but the choice must be consistently applied. Moreover it can be done by the initiator of the message.

Lemma 2.2 *Using the* handrail *algorithm presented in Figure 6, a message can travel along the boundaries of a square of a given size d with "globally consistent turns" in unlabelled tori.*

Proof The algorithm is presented in Figure 6. Before sending a message, a processor, say X_1 in Figure 5, chooses arbitrarily two perpendicular communication links locally labelled r and k and labels them accordingly, without loss of generality say *east* and *south* respectively. Enough information should be provided for the message to travel in the "world according to X_1".

The message contains its own *handrail.* The *handrail* is the immediate neighbour according to the other arbitrarily chosen perpendicular direction (Handrail := $H1_k$). The processor can then initiate the message on one of these two links (say *east*) to be forwarded at distance d in the same dimension (*east*/*west*). In our example, the message heading to the *east* contains the name of the processor at distance 1 at *south* (X_1 sets the handrail's value to Y_1 in the message).

Upon receipt of the message, the neighbour X_2 deduces from the handrail's value Y_1 that, according to X_1, its *south* is the link heading to Y_2: i.e., the perpendicular link

```
procedure Handrail algorithm
H1_r, H2_r are the sets of nodes accessible by link r (at distance 1 and 2 respectively).
begin
If Mystatus = Initiator :

(0) Choose arbitrarily k, r s.t.  H2_r ∩ H2_k ≠ ∅ /* perpendicular edges */
    Handrail := H1_k
    send TOKEN(MyId,Handrail,d,0) on link r
    Upon receipt of TOKEN(MyId,Handrail,ToGo,SoFar) on link k
           Terminate

If Mystatus ≠ Initiator :

(1) Upon receipt of TOKEN(Id1,Handrail,ToGo,SoFar) on link r
    SoFar := SoFar + 1
    find k, k ≠ r s.t.  H2_r ∩ H2_k = Handrail
    if SoFar ≠ ToGo then /* must be straight forwarded */
      Handrail := H1_k
      find s s.t.  H2_r ∩ H2_s = ∅ /* the same dimension */
      send TOKEN(MyId,Handrail,ToGo,SoFar) on link s
    else
      Handrail := H1_r
      send TOKEN(MyId,Handrail,ToGo,0) on link k /* turn */
    endif
end TOKEN
end Handrail algorithm
```

Figure 6: Consistent Square Travel with the Handrail Algorithm.

which can lead to the processor Y_1 in 2 hops, $((1)k,\ H2_r \cap H2_k = \text{Handrail})$. It updates the *handrail* value to Y_2 (its own relative *south* now) and forwards the updated message. As it can easily be guessed at this stage, the set of nodes used as a handrail is the set of immediate "inside" neighbours of the nodes of the visited path (as shown with the dashed area inside the travelling path in Figure 5).

The message is forwarded with the same principle until reaching the processor at distance d, say X_d. To turn, the message uses the handrail as a pivot. X_d helps the message to find out which way to turn. Using the idea described above, X_d detects the relative *south* and forwards the message to this direction after updating the *handrail* value to X_{d-1} as the relative node at *west* (the new inside face). The straightforward process is repeated d times to reach a node, say B_d, where the same turn-and-forward process is repeated to B_1 and finally to X_1. □

The following Theorem can be immediately deduced.

Theorem 2.1 *The algorithm elects a* leader *using $O(N)$ messages in an unlabelled two-dimensional torus of $\sqrt{N} \times \sqrt{N}$ nodes.*

Proof For brevity we only present the modifications to the original algorithm regarding the modifications due to the handrail algorithm.

By construction of the consistent district labelling in Lemma 2.1 as a preprocessing phase, all processors are initially active and therefore can process in phases.

Each processor chooses arbitrarily two perpendicular communication links and labels them accordingly, and sends a "looking" message (instead of a token) around the square boundary using the *handrail* algorithm (Figure 6) as presented in Lemma 2.2. If the message becomes a "SeenbySmaller" and must continue along the boundary of a processor with the larger value, it will update its *handrail* value to the one stored locally in the visited node and which corresponds to the processor with the larger value. (Each processor stores the handrail of the processor's boundary it belongs to.)

After $\sqrt{N}$ phases, the "wrap around" phases are executed as in [18] (they required only straightforward communications). Therefore, the complexity of the algorithm remains unchanged which proves the Theorem. □

Corollary 2.1 *The size of the messages in the Election algorithm in an unlabelled torus requires exactly* $\log m$ *extra bits, where m is the largest value in ID.*

Proof Immediate. Only the handrail's value, a processor identity, is added in the message. □

2.4 Generalisation of the handrail

Using the same *handrail* paradigm, one should be able to solve other distributed problems in unlabelled tori with the same asymptotic complexity than in the labelled case. Indeed, the *handrail* provides more than just allowing to turn always "clockwise" or "counter-clockwise". We have seen that, at each processor, the message has the choice to either keep on the same direction or to turn consistently. By elimination of choices, the message can turn in the opposite direction, if desired, by choosing the remaining link and can update the handrail accordingly. Moreover, the message will be able to repeat this as many times as necessary and still being able to know its relative position from its initiator: the message must contain the number of hops in each direction (this can be compressed to a shortest path notation by elimination of combination or application of modulo, e.g., as presented in [6]).

Extending the result to d-dimensional unlabelled tori or grids ($d > 2$) is based on the same consistent district labelling algorithm. To acquire the identity of each processor at distance 2, the algorithm costs $8\,d$ messages. Again, the sets of nodes accessible by a link which have no intersection are in the same dimension, perpendicular otherwise. The handrail is chosen according to d arbitrarily chosen perpendicular links and corresponds to as many node identities (at most $d-1$) which are necessary to turn consistently: one processor identity is sufficient to do consistent turns in a 2-dimensional sub-torus, two are necessary for a 3-dimensional,...

3 Unlabelled Chordal Rings

A common technique to improve reliability of ring networks is to introduce link redundancy; that is, to have each node connected to two or more additional nodes in the network. With alternate paths between nodes, the network can sustain several nodes and links failures. The overall topological structure of these *redundant rings* is always highly regular, [1]; in particular, the set of ring edges (*regular*) and additional edges (*bypass*) form a *Circulant Graph*. Because of an uncoordinated literature, numerous terms to name this topology (e.g., *Chordal Ring* or *Distributed Loop Computer Networks*) are commonly used. A detailed survey of these topologies is presented in [3]. For sake of simplicity, we will use the term *chordal ring* in this paper.

A chordal ring $C_N\langle d_1, d_2, ..., d_k\rangle$ of size N and k-chord structure $\langle d_1, d_2, ..., d_k\rangle$, with $d_1 = 1$, is a ring R_N of N processors $\{p_0, p_1, ..., p_{N-1}\}$, where each processor is also directly connected to the processors at distance d_i and $N - d_i$ by additional incident chords. The link connecting two nodes is labelled by the distance which separates these two nodes on the ring, i.e. following the order of the nodes on the ring: the node p_i is connected to the node $p_{i+d_j \bmod N}$ through its link labelled d_j (as shown in figure 7(a)). In particular, if a link, between two processors p and q, is labelled by distance d at processor p, this link is labelled by $N - d$ at the other incident processor q, where N is the number of processors. Note that both *rings* and *complete graphs* are chordal rings, denoted $C_N\langle 1\rangle$ and $C_N\langle 1, 2, 3, \cdots, \lfloor N/2\rfloor\rangle$ respectively. This link labelling, known as *distance* or *chordal* labelling, is a Sense of Direction [7]. Some instances of the chordal ring are reminiscent of

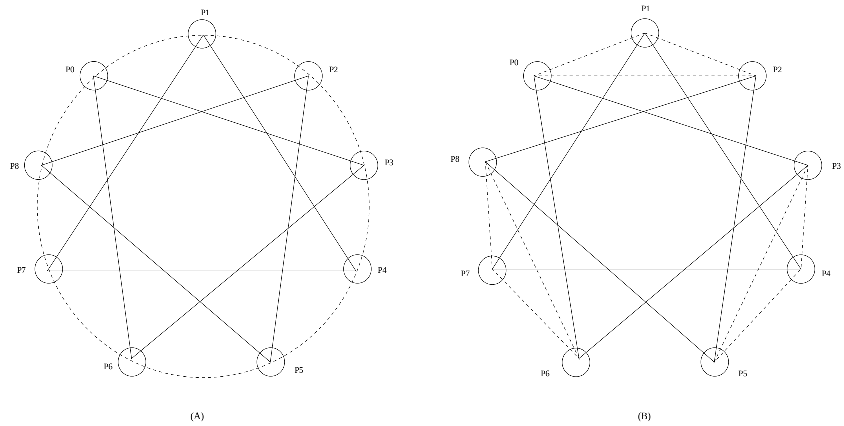

Figure 7: (a) Chordal Ring $C_9\langle 1, 3\rangle$ and (b) Torus.

the torus, as shown in Figure 7. (It is worth pointing out that some designs for *redundant tori* and *redundant hypercubes* are also chordal rings, e.g. [4].)

3.1 Electing a Leader in a Chordal Ring with $\mathcal{SD}$

For this class of network topology, the fundamental problem of Leader Election has been extensively studied. The main problem is to determine the minimal sets of links that must be added to the ring in order to achieve a linear election algorithm. Along the years, the minimal number of necessary chords decreased from $O(\log N)$ [2] to $O(\log \log N)$ [10], and to $O(\log \log \log N)$ [17]. The ultimate case is the chordal ring $C_N\langle 1, n\rangle$, where n is approximately $\sqrt{N}$. Using Peterson's algorithm, Fabri and Tel presented a $\Theta(N)$ solution for this case [22].

Fabri and Tel's algorithm. We briefly recall the main features of their algorithm. Each processor has four communication links labelled $\{1, N-1, n, N-n\}$, conveniently denoted as $\{+1, -1, +n, -n\}$ in the following. All the processors are assumed to be initially active and then process in phases. The basic goal of each active process on the ith phase is to mark off the boundary of a d-square ($d = \alpha^i$ for some constant α). This is done by sending a "looking" message at distance d through links $+1$, d through links $+n$ then d through links -1, and finally d through links $-n$ to the original node. Marking its boundaries, an active processor can *see* or *been seen* by an other active processor.

The promotion rules applied and the modifications of the message boundary are as in Peterson's algorithm. The algorithm will terminate when a processor will reach phase $d = \alpha^i \geq \sqrt{N}$, at this time there are at most $(2-\alpha^2)^{-1}$ active processors surviving. In this case, instead of "wrap-around" phases, an election is initiated along the ring (on link labelled $+1$ and -1) to distinct the leader among the constant number of survivors. This termination phase costs $O(N)$ messages.

3.2 Electing a Leader in a Chordal Ring without $\mathcal{SD}$

Again, the previous observation that the orientation of the square is irrelevant holds. In Tel's algorithm, if marking the boundary is initiated through $+1$ edges, it terminated through $-n$ chords. Obviously, marking the boundary by reversing the path will have no impact on the correctness of the algorithm. Similarly, a sequence of turns $[+n, +1, -n, -1]$ or any permutation which respects the alternate dimensional sequence is still valid. The algorithm is only based on the ability of the search area to be sufficiently large to cross another boundary. Using the same *handrail* paradigm, we introduce the necessary modifications.

Lemma 3.1 *The algorithm presented in Figure 3 computes a consistent district labelling for every node using exactly* $16N$ *messages in an unlabelled chordal ring* $C_N\langle 1, n\rangle$*, (*$n \approx \sqrt{N}$ *and* $n > 3$*).*

Proof Same as in Lemma 2.1 using the algorithm presented in Figure 3. Each processor sends its identity to each of its neighbours, which forwards it to its three remaining neighbours. The case $n \leq 3$ avoids consistency, but obvious solutions for this case can be designed to obtain a linear complexity. □

For the sake of the explanation we applied the algorithm in a system depicted in Figure 8, using the notation as in Figure 1. Again, when the algorithm terminates, the node O knows that N, E, S, W are its immediate neighbours; which link must be used to reach them, respectively $north, east, south, west$; for each link, which node at distance 2 can be reached, $H2_{north} = \{NW, NN, NE\}$, $H2_{east} = \{NE, EE, SE\}$, $H2_{south} = \{SE, SS, SW\}$, $H2_{west} = \{SW, WW, NW\}$ respectively; that two sets $H2_i$ and $H2_j$ which intersection is empty are in the same dimension ($north/south$ or $east/west$), perpendicular if not.

In this case, however, the problem is different from the torus where both dimensions are of same length and play a symmetric role in the algorithm. In the present setting, a node does not know which dimension between $north/south$ and $east/west$ corresponds to the ring labelling $+1/-1$. This means that after this preprocessing phase a node cannot distinguish between a ring edge and a bypass chord.

Lemma 3.2 *Using the* handrail *algorithm, a message can travel along the boundaries of a square of a given size* d *(*$d \leq \sqrt{N}$*) with "globally consistent turns" in an unlabelled chordal ring* $C_N\langle 1, n\rangle$*, (*$n \approx \sqrt{N}$*).*

Proof Similar to Lemma 2.2 using the algorithm presented in Figure 6. First the consistent district labelling is computed as shown in Lemma 3.1. Second, a processor chooses arbitrarily two perpendicular communication links and labels them accordingly, without loss of generality say *east* and *south*, as shown on Figure 8. The processor then initiates the message on one of these two links (say *east*) to be forwarded at distance d

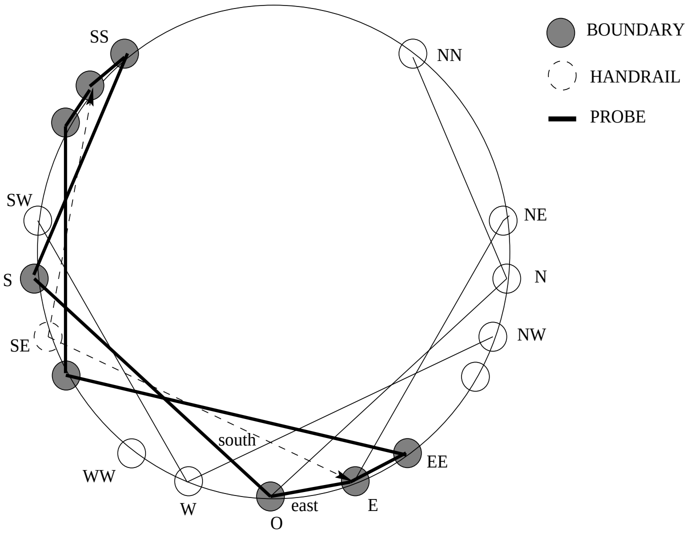

Figure 8: Handrail in the Unlabelled Chordal Ring $C_N\langle 1, n\rangle$.

in the same dimension (*east/west*). The message heading to the *east* contains the name of the processor at distance 1 at *south* (O sets the handrail's value to S in the message).

Upon receipt of the *east* message, the immediate neighbour E of the node O deduces from the handrail's value S that its *south*, according to O, is the link heading to SE (the perpendicular link which can lead to the processor S in 2 hops). It updates the *handrail* value to its own relative *south* (SE) and forwards the updated message.

The same process is repeated until the message returns to its original node. As shown in Figure 8 for an execution with $d = 2$, SE will be used repeatedly as handrail since it is the centre of a 3×3 grid). $\square$

The Election Theorem follows.

Theorem 3.1 *Using the handrail, the algorithm elects a* leader *using* $\Theta(N)$ *messages in unlabelled chordal rings* $C_N\langle 1, n\rangle$*,* ($n \approx \sqrt{N}$)*.*

Proof By construction of the consistent district labelling (Lemma 3.1) as a preprocessing phase, all processors are initially active and therefore can process in phases.

Each processor chooses arbitrarily two perpendicular communication links and proceeds as presented in Lemma 3.2. Each processor initiates a "looking" message. If the message becomes a "SeenbySmaller" and must continue along the boundary of a processor with the larger value, it updates its *handrail* value to the one stored locally in the visited node and which corresponds to the processor with the larger value. Every node stores the respective handrail's value of the boundary it belongs to.

After $\sqrt{N}$ phases, only a constant number of processors remain active. An election termination phase on the ring between them can be trivially achieved by sending a message in each perpendicular dimension (since the node cannot distinguish between a ring link and a bypass chord). The node receiving its own messages back declares itself as the leader. This technique only doubles the number of termination messages compare to the original algorithm. Each phase, including the termination, costs at most $O(N)$ messages which proves the theorem. □

Corollary 3.1 *During the termination phase of the* Election *algorithm for unlabelled chordal ring $C_N\langle 1, n\rangle$, ($n \approx \sqrt{N}$), the leader distinguishes between the chordal links and the ring links iff n and N are not co-prime.*

Proof During the termination phase, the message coming back after N hops is the one which travels on the ring and allows the processor to distinguish between the $-1/+1$ ring links and $-n/+n$ chords. Both messages used N hops iff n and N are co-prime [12] (i.e., $\gcd(n, N) = 1$). In this case, the processor cannot choose arbitrarily a link to be labelled $+1$ since each local link can be used to build a ring. Indeed, it is then possible to build two isomorphic chordal rings where one has a chord of length n^{-1} (modulo N), with n approximately $\sqrt{N}$ (e.g., $C_{23}\langle 1, 5\rangle$ is isomorphic to $C_{23}\langle 1, 9\rangle$). □

Improvement of the Termination. If n and N are not co-prime, topological observations allow an improvement of the termination phase. First observe that in any phase i ($\alpha^i < \sqrt{N}$), because a message makes at most $\sqrt{N}$ hops in a direction before turning, a message initiated through a given link, say *east*, will always come back through the arbitrarily chosen perpendicular link (i.e. the *handrail*), say *south*.

Second, note that when $N \bmod n = 0$ (i.e., $n = \sqrt{N}$) the chords provides a "wrap-around" of size $\sqrt{N}$. At the last iteration, $d = \sqrt{N}$, two cases may occur:

- The message comes back after only $\sqrt{N}$ hops through link *west* (opposite of *east*). The initiator immediately deduces that *east* is a chord link. It then distinguishes between the $-1/+1$ ring links (*south*/*north*) and the $-n/+n$ chords (*west*/*east*).

- The message comes back after only $3\sqrt{N}$ hops through link *east* (used for sending). The processor immediately deduces that *east* is a ring link. It then distinguish between the $-1/+1$ ring links (*west*/*east*) and the $-n/+n$ chords (*south*/*north*).

In both cases, the termination phase can proceed as in the original algorithm using only the ring.

Corollary 3.2 *The size of the messages in the Election algorithm in the unlabelled chordal ring requires exactly* $\log m$ *extra bits, where the largest value in ID is m.*

Proof Immediate (as in Corollary 2.1). Only the handrail's value is added in the message. □

4 Computing Sense of Direction

The *handrail* can be used as a generic tool to solve many other distributed problems in unlabelled topology, but its use may remain a tedious task for the designer of the distributed algorithm. Other alternatives may be considered.

Since Sense of Direction is known to improve the communication complexity of distributed algorithms, computing $\mathcal{SD}$ as a preprocessing phase in unlabelled topology has been studied [21, 22]. So far, results have not been encouraging: any algorithm computing $\mathcal{SD}$ exchanges at least $\Omega(e - 1/2\,N)$ messages in a network with N nodes and e edges. This result is not attractive for dense topologies: $\Omega(N^2)$ for cliques and $\Omega(N \log N)$ for the hypercube; even if algorithms matching the lower bounds have been proposed. The interest is more relevant for topologies with a linear number of edges such as tori; however the best algorithm known so far requires $O(N \log N)$ [22]. Here, we present a solution to build $\mathcal{SD}$ using $\Theta(N)$ messages for each topology. This proves that solutions for any problem in an unlabelled topology will be immediately deduced from the according solution in the labelled topology without asymptotic overcost.

Theorem 4.1 *A Leader can compute Sense of Direction using exactly $N + \sqrt{N}$ messages in unlabelled two-dimensional tori with $\sqrt{N} \times \sqrt{N}$ nodes.*

Proof The algorithm is executed in two concurrent phases, as shown in Figure 9. First the leader chooses arbitrarily two perpendicular communication links and labels them accordingly, without loss of generality say *east* and *south*. The leader then initiates a message on each of these two links to be forwarded in the same dimension (*east*/*west* and *north*/*south*) as shown in the algorithm presented in Theorem 2.1. Both messages build their own *handrail* (according to the other arbitrarily chosen direction): the message heading to the *east* contains the name of the processor at distance 1 at *south*, the message heading to the *south* contains the processor at distance 1 at *east*.

Upon receipt of the *east* message, the immediate neighbour of the leader labels its incoming link as *west* and the other link on same dimension as *east*. With the handrail's name in the message, it deduces where is its *south* (the perpendicular link which can lead to the processor mentioned in 2 hops) and initiates a message to the *south* to be forwarded in the same dimension in order to take care of the *north*/*south* labelling in this column. It can then resume its *east*/*west* phase by changing the label of the handrail to its own *south* node and can forward the updated leader message. The second phase (started concurrently upon receipt of the first phase message) corresponds to the *north*/*south* labelling. The *north*/*south* message is just forwarded and does not generate any message in the *east*/*west* dimension. However using the same technique as above, the message must contain the according handrail's name (the processor at distance 1 at *east*) to distinct *east* and *west* locally (and update accordingly). Each message will be stopped upon receipt by its initiator.

The algorithm will terminate in $O(N)$ time and will use exactly $\sqrt{N}$ messages for the first phase and N for the second, which proves the Theorem. $\square$

Corollary 4.1 *A distributed algorithm computes Sense of Direction using* $\Theta(N)$ *messages in unlabelled two-dimensional tori with* $\sqrt{N} \times \sqrt{N}$ *nodes.*

Proof Immediate from Theorem 2.1 and Theorem 4.1. $\square$

Remarks: *Processor naming.* Note that the algorithm can name the processor in the commonly used way ($\{P_{i,j} : 0 \leq i, j \leq (\sqrt{N} - 1)\}$) without overcost: the leader names itself $P_{0,0}$ and initiates the naming of each $\{P_{0,j} : 0 \leq j \leq (\sqrt{N} - 1)\}$ during the first step (*east*), which will initiate the naming of their respective $\{P_{i,j} : 0 \leq i \leq (\sqrt{N} - 1)\}$ during the second step (*south*).
Grid. Without wrap around, in a regular d-dimensional array processors, the algorithm presented in Theorem 4.1 is still valid. For each phase, the processor must initiate a message in both direction of the dimension which will be stopped when encountering the boundary.
Higher Dimension. The algorithm can be designed in a simple manner by extending the handrail as shown in Section 2.

Theorem 4.2 *A Leader can compute Sense of Direction using at most* $N + 2\sqrt{N}$ *messages in unlabelled chordal ring* $C_N\langle 1, n\rangle$*,* $(n \approx \sqrt{N})$*.*

Proof The algorithm is shown in Figure 10. When n and N are not co-prime, the algorithm is executed in two phases. After completion of the election algorithm, by corollary 3.1, the leader knows which pair of link is associated to $-1/+1$. It sends a probe message on one of these two links (thus, labelled arbitrarily $+1$). The message is forwarded in the same dimension n times to be received by an immediate neighbour of the leader which sends back to the leader directly. Upon receipt, the leader can assign the label $+n$ to the link from which it receives the message, and ends this phase.

The rest of the algorithm can be achieved with a similar method to the algorithm used in theorem 4.1 (providing concurrency and achieving an optimal time complexity, $O(\sqrt{N})$). However, for brevity, we give a simpler method based on the ring-based structure (requiring $O(N)$ time though).

The leader initiates the second phase by sending a token through link $+1$, the message contains a *handrail* set to the identity of its neighbour accessible through link $+n$. The message will be forwarded along the same dimension (the ring) while the *handrail* will be updated accordingly. Upon receipt, each processor will be able to label consistently each of its link (knowing that -1 is the receiving link and $+n$ is the corresponding *handrail*). The second phase will terminate when the neighbour of leader (through link -1) receives the token. Overall, the algorithm requires exactly $N + \sqrt{N}$ messages.

procedure Torus SD algorithm
$H1_r, H2_r$ are the sets of nodes accessible by link r (at distance 1 and 2 respectively).
begin
If Mystatus = Leader :

(0) Choose arbitrarily k, r s.t. $H2_r \cap H2_k \neq \emptyset$ /* perpendicular edges */
    $east := r$; $south := k$
    choose r' s.t. $H2_r \cap H2_{r'} = \emptyset$; $west := r'$
    choose k' s.t. $H2_k \cap H2_{k'} = \emptyset$; $north := k'$
    Handrail := $H1_k$
    send ONE(Handrail) on link r
    Handrail := $H1_r$
    send TWO(Handrail) on link k
    Upon receipt of ONE(Handrail) and TWO(Handrail)
        Terminate

If Mystatus $\neq$ Leader :

(1) Upon receipt of ONE(Handrail) on link r'
    find k, $k \neq r'$ s.t. $H2_{r'} \cap H2_k =$ Handrail
    $west := r'$; $south := k$
    choose r s.t. $H2_r \cap H2_{r'} = \emptyset$; $east := r$
    choose k' s.t. $H2_k \cap H2_{k'} = \emptyset$; $north := k'$
    Handrail := $H1_k$
    send ONE(Handrail) on link r
    Handrail := $H1_r$
    send TWO(Handrail) on link k
    Upon receipt of TWO(Handrail)
        Terminate

(2) Upon receipt of TWO(Handrail) on link k'
    find r, $k' \neq r$ s.t. $H2_r \cap H2_{k'} =$ Handrail
    $east := r$; $north := k'$
    choose r' s.t. $H2_r \cap H2_{r'} = \emptyset$; $west := r'$
    choose k s.t. $H2_k \cap H2_{k'} = \emptyset$; $south := k$
    Handrail := $H1_r$
    send TWO(Handrail) on link k
    Terminate

end Torus SD algorithm

Figure 9: Leader constructing Sense of Direction in a Torus.

When n and N are not co-prime, the algorithm may need to repeat the first phase twice. Indeed, the leader does not know which pair of link corresponds to $-1/+1$. It must choose arbitrary and proceed as above. If after n hops the node receiving the message does not match a neighbour of the leader, this node can undoubtedly detect that its perpendicular pair of link corresponds to the dimension $-1/+1$. This node can then process the 2 phases as the leader in the previous case. This modification adds only $\sqrt{N}$ messages to the $(N+\sqrt{N})$ messages required above, which gives a $N+2\sqrt{N}$ in the worst case, proving the theorem. □

Corollary 4.2 *A distributed algorithm computes Sense of Direction using $\Theta(N)$ messages in unlabelled chordal rings $C_N\langle 1, n\rangle$, ($n \approx \sqrt{N}$).*

Proof Immediate from Theorem 3.1 and Theorem 4.2. □

5 Concluding Remarks and Open Problems

This study shows how a limited number of edges, usually considered as a drawback, can be used efficiently to obtain an optimal communication complexity. We built a partial structural information which is proved sufficient to allow the design of linear Election algorithms, and built Sense of direction without asymptotic overcost. Both algorithms use the same fundamental ideas to achieve efficiency, indicating that the technique is both powerful and general. Since, the solution is based on a preprocessing phase which will work for any network with constant degree (i.e., linear number of edges), it can be used for such other topologies, e.g. [19]. The question is still open for topologies such as the hypercube where the degree is logarithmic. In this case such method would cost $O(N \log N)$ preliminary messages and thus is not suitable. The best known complexity is $O(n \log\log n)$ messages [5] compared to $O(n)$ in the labelled case [6].

Finally, we showed that extending the result to d-dimensional unlabelled tori or grids ($d > 2$) was a simple matter. For brevity, we did not investigate the case of unlabelled chordal ring with more than one chord. The problem does not appear as simple (even if one of the chords is known to have length $\sqrt{N}$).

Acknowledgments. We like to thank B. Litow for his comments on this paper, and the anonymous referees for pointing out the relevance to a recent paper [16].

procedure Chordal SD algorithm
$H1_r, H2_r$ are the sets of nodes accessible by link r (at distance 1 and 2 respectively).
begin
If Mystatus = Leader or NewLeader /* do not know $\{+n\}$ */:
(1) choose r s.t. $r \in \{-1, +1\}$ if known
    send PROBE(MyId,n) on link r
    Upon receipt of PROBE(MyId,n) on link k
        Mystatus := Initiator
        label(+1) := r; label($+n$) := k
        start(3)

If Mystatus $\neq$ Leader and Mystatus $\neq$ NewLeader :
(2) Upon receipt of PROBE(Id,d) on link r'
    if $d = 0$
      then if $\exists k$, s.t. $H1_k =$ Id
          then send PROBE(Id,n) on link k
          else Mystatus := NewLeader; start(1) with r s.t. $H2_{r'} \cap H2_r \neq \emptyset$
      else
          choose r s.t. $H2_r \cap H2_{r'} = \emptyset$
          send PROBE(Id,(d-1)) on link r

If Mystatus = Initiator: /* knows $\{+1\}$ and $\{+1\}$ */
(3) label(+1) = r; label($+n$) = k
    choose r' s.t. $H2_r \cap H2_{r'} = \emptyset$; label(-1) := r'
    choose k' s.t. $H2_k \cap H2_{k'} = \emptyset$; label($-n$) := k'
    Handrail := $H1_k$
    send TOKEN(Handrail) on link r
    Upon receipt of TOKEN(Handrail)
        Terminate

If Mystatus $\neq$ Initiator :
(4) Upon receipt of TOKEN(Handrail) on link r'
    find k, $k \neq r'$ s.t. $H2_{r'} \cap H2_k =$ Handrail
    label(-1) := r'; label($+n$) := k
    choose r s.t. $H2_r \cap H2_{r'} = \emptyset$; label(+1) := r
    choose k' s.t. $H2_k \cap H2_{k'} = \emptyset$; label($-n$) := k'
    Handrail := $H1_k$
    send TOKEN(Handrail) on link r
    Terminate

end Chordal SD algorithm

Figure 10: Leader constructing Sense of Direction in a Chordal Ring.

References

[1] B.W. Arden and H. Lee. Analysis of chordal ring. *IEEE Transactions on Computers*, C-30(4):291–295, April 1981.

[2] H. Attiya, J. van Leeuwen, N. Santoro, and S. Zaks. Efficient elections in chordal ring networks. *Algorithmica*, 4:437–446, 1989.

[3] J.-C. Bermond, F. Comellas, and D.F. Hsu. Distributed loop computer networks: a survey. *Journal of Parallel and Distributed Computing*, 24:2–10, 1995.

[4] J. Bruck, R. Cypher, and C.-T. Ho. Fault-tolerant meshes and hypercubes with minimal numbers of spares. *IEEE Transactions on Computers*, 42(9):1089–1104, September 1993.

[5] S. Dobrev and P. Ruzicka. Linear broadcasting and n loglog n election in unoriented hypercubes. In *Proc. of the 4th International Colloquium on Structural Information and Communication Complexity, Sirocco'97*, Ascona, Switzerland, July 1997. To appear.

[6] P. Flocchini and B. Mans. Optimal elections in labeled hypercubes. *Journal of Parallel and Distributed Computing*, 33(1):76–83, 1996.

[7] P. Flocchini, B. Mans, and N. Santoro. Sense of direction: formal definition and properties. In *Proc. of 1st Colloquium on Structural Information and Communication Complexity*, Ottawa, Canada, 1994. 9–34.

[8] P. Flocchini, B. Mans, and N. Santoro. On the impact of sense of direction on message complexity. *Information Processing Letters*, 1997. To appear.

[9] R.G. Gallager, P.A. Humblet, and P.M. Spira. A distributed algorithm for minimum spanning tree. *ACM Transactions on Programming Languages and Systems*, 5(1):66–77, 1983.

[10] T.Z. Kalamboukis and S.L. Mantzaris. Towards optimal distributed election on chordal rings. *Information Processing Letters*, 38:265–270, 1991.

[11] E. Korach, S. Moran, and S. Zaks. Optimal lower bounds for some distributed algorithms for a complete network of processors. *Theoretical Computer Science*, 64:125–132, 1989.

[12] B. Litow and B. Mans. On isomorphic chordal rings. In *Proc. of the Seventh Australasian Workshop on Combinatorial Algorithms (AWOCA'96)*, pages 108–111, Magnetic Island, July 1996. Univ. of Sydney, BDCS-TR-508.

[13] M.C. Loui, T.A. Matsushita, and D.B. West. Election in complete networks with a sense of direction. *Information Processing Letters*, 22:185–187, 1986. see also Information Processing Letters, vol.28, p.327, 1988.

[14] B. Mans. Optimal distributed algorithms in unlabeled tori and chordal rings. In *Proc. of the 3rd International Colloquium on Structural Information and Communication Complexity, Sirocco'96*, pages 17–31, Siena, Italy, June 1996. Carleton Press.

[15] B. Mans and N. Santoro. On the impact of sense of direction in arbitrary networks. In *Proc. of 14th International Conference on Distributed Computing Systems*, pages 258–265, Poznan, 1994.

[16] S. Olariu, I. Stojmenovic, and A. Zomaya. On the dynamic initialization of parallel computers. In *Proc. of the 8th IEEE International Parallel Processing Symposium*, Geneva, Switzerland, April 1997. To appear.

[17] Yi Pan. A near-optimal multi-stage distributed algorithm for finding leaders in clustered chordal rings. *Information Sciences*, 76(1-2):131–140, 1994.

[18] G.L. Peterson. Efficient algorithms for elections in meshes and complete networks. Technical Report TR-140, Dept. of Computer Science, Univ. of Rochester, Rochester, NY-14627, 1985. Has been replaced by *Improved algorithms for elections in meshes and complete networks*, Georgia Institute of Tech., Dec. 1986.

[19] F.P. Preparata and J. Vuillemin. The cube-connected cycles: a versatile network for parallel computations. *Communication of the ACM*, 25:479–482, 1981.

[20] N. Santoro. Sense of direction, topological awareness and communication complexity. *SIGACT NEWS*, 2(16):50–56, Summer 1984.

[21] G. Tel. Network orientation. *International Journal of Foundations of Computer Science*, 5(1), 1994.

[22] G. Tel. Sense of direction in processor networks. In M. Bartošek, J. Staudek, and J. Wiedermann, editors, *Proc. of SOFSEM'95, Theory and Practice of Informatics*, volume 1012 of *lncs*, pages 50–82, Milovy (CZ), Nov. 1995. Springer-Verlag.