Lecture Notes in Computer Science 1499

Shay Kutten (Ed.)

Distributed Computing

12th International Symposium, DISC'98
Andros, Greece, September 1998
Proceedings

Springer

Shay Kutten (Ed.)

Distributed Computing

12th International Symposium, DISC'98
Andros, Greece, September 24-26, 1998
Proceedings

Springer

Volume Editor

Shay Kutten
IBM T.J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598, USA
and
Faculty of Industrial Engineering and Management
The Technion
Haifa 32000, Israel
E-mail: kutten@ie.technion.ac.il

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Distributed computing : 12th international symposium ; proceedings / DISC '98, Andros, Greece, September 24 - 26, 1998. Shay Kutten (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Budapest ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 1998
(Lecture notes in computer science ; Vol. 1499)
ISBN 3-540-65066-0

CR Subject Classification (1991): C.2.4, C.2.2, F.2.2, D.1.3, F.1, D.4.4-5

ISSN 0302-9743
ISBN 3-540-65066-0 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10638936 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

Preface

The name of WDAG was changed to DISC (The International Symposium on DIStributed Computing) to reflect the expansion from a workshop to a symposium, as well as the expansion of the area of interest. DISC'98 builds on, and expands, the tradition of WDAG as a truly international symposium on all aspects of distributed computing. It aims to reflect the exciting and rapid developments in this field, and to help lead its continued exploration, by serving as a meeting point for researchers from various sub-fields.

Following 11 WDAGs, DISC'98 is the 12^{th} in the series. This volume of DISC'98 contains the 28 extended abstracts presented at the symposium, held on September 24-26, 1998, in Andros, Greece. Opening the volume are three extended abstracts based on three keynote lectures given at the symposium by Yoram Moses, Nicola Santoro, and Santosh Shrivastava.

The contributed papers were selected from 87 submissions, at a meeting of the program committee held in Athens, following an on-line discussion. The papers were read and evaluated by the program committee members, with the additional very helpful assistance of external reviewers when needed. It is expected that papers based on most of these extended abstracts will appear in scientific journals.

The program committee consisted of: Yair Amir (Johns Hopkins University), Tushar Chandra (IBM T.J. Watson Research Center), Bernadette Charron-Bost (CNRS, Ecole Polytechnique), Alfredo de Santis (University of Salerno), Faith Fich (University of Toronto), Ajei Gopal (IBM, Pervasive Computing), Yuh-Jzer Joung (National University of Taiwan), Shay Kutten – Chair (IBM T.J. Watson Research Center and Technion), Marios Mavronicolas (University of Cyprus), Louise Moser (University of California, Santa Barbara), Noam Nisan (Hebrew University and IDC), Boaz Patt-Shamir (Tel Aviv University), Peter Ruzicka (Comenius University), Nicola Santoro (Carleton University), Marc Shapiro (INRIA), Alex Shvartsman (University of Connecticut), and Paul Spirakis (Patras University and Computer Technology Institute).

The local arrangements (both for the committee meeting, and for the conference) were made by a team headed by Paul Spirakis (local arrangements chair) and Panagiota Fatourou (local arrangements co-chair) from the Computer Technology Institute in Greece. They also took care of advertising the symposium, and the program committee chair would like to thank them for a wonderful job.

We would like to thank all the authors who submitted extended abstracts. There were interesting papers among those we could not accept, and their authors also contributed towards enhancing DISC. We hope to see the continuing support of the next DISC, by everybody. We would also like to thank the invited speakers for joining us in Andros. The program chair would like to thank the program committee members who invested a lot of work and thought in the symposium.

We would like to thank ACM SIGACT for letting us use its automated submission server and its program committee server at sigact.acm.org, and Springer-Verlag for following the tradition and publishing our proceedings. It is a special pleasure to thank

Esther Jennings and Rinat Regev for helping with operating the SIGACT server and Dana Fruchter for her help in putting together the final proceedings.

DISC thanks the following for their generous support: The Greek Ministry of Education, the Greek General Secretary of Research and Technology, the University of Patras, Greek PTT (OTE), Intracom, and Intrasoft.

Continuation of the DISC events is supervised by the DISC Steering Committee, which, for 1998, consists of: Özalp Babaǧlu (U. Bologna), Bernadette Charron-Bost (E. Polytechnique), Vassos Hadzilacos (U. Toronto), Shay Kutten (IBM and Technion), Marios Mavronicolas (U. Cyprus), Sam Toueg (Cornell) – Chair, and Shmuel Zaks (Technion) - Vice Chair.

July 1998

Shay Kutten
DISC'98 Program Committee Chair

External Reviewers

The Program Committee wishes to thank the following persons, who acted as referees for DISC'98:

Y. Afek
D. Agrawal
M. Aguilera
A. Allist
J. H. Anderson
A. Baggio
G. Banavar
C. Bastarrica
A. Basu
Y. Ben-Asher
K. Berket
X. Blondel
P. Boldi
R. Borgstrom
S. Chari
O. Cheiner
M. S. Chen
P. Dasgupta
A. K. Datta
S. Dandamudi
O. Dedieu
C. Delporte-Gallet
S. Demurjian
S. Dolev
N. Dorta
T. Eilam
B. Englert
O. Etzion
H. Fauconnier
P. Flocchini
B. Folliot
C. Fournet
P. Fraigniaud
J. A. Garay
C. Gavoille
M. Greenwald
R. Guerraoui
M. Herlihy
L. Higham
G. Hunt
J. James
E. Jennings
M. Kalantar
V. Kalogeraki
D. Kaminsky
S. Katz
I. Keidar
C. Khoury
R. R. Koch
E. Kushilevitz
F. Le Fessant
C. L. Lei
G. Le Lann
S. Leonardi
G. Lindstrom
T. J. Liu
D. Malkhi
B. Mans
C. Mascolo
F. Mattern
P. M. Melliar-Smith
M. Moir
B. Mukherjee
P. Narasimhan
L. Narayanan
G. Neiger
R. Netzer
S. Onn
J. Parameswaran
D. Peleg
J. Pershing
F. Petit
G. Pierre
K. Potter-Kihlstrom
S. Rajsbaum
A. Roncato
E. M. Royer
E. Santos
A. Schiper
H. Shachnai
D. Shaw
G. Singh
J. Stanton
R. Strom
D. Sturman
R. B. Tan
G. Taubenfeld
G. Tel
L. A. Tewksbury
S. Toueg
D. Touitou
Y. K. Tsay
G. Tsudik
M. Tuttle
R. VanRenesse
V. Villain
D. W. Wang
F. Wang
M. J. Ward
S. Weiss
A. Wool
D. Wu
L. C. Wuu
S. Zaks

Table of Contents

Keynote Lectures

Contributed Papers

Sense of Direction in Distributed Computing

Paola Flocchini[1], Bernard Mans[2], Nicola Santoro[3]

[1] Département d'Informatique Université du Quebec à Hull
(flocchini@uqah.uquebec.ca)
[2] Department of Computing, School of MPCE, Macquarie University, Sydney,
(bmans@mpce.mq.edu.au)
[3] School of Computer Science, Carleton University, Ottawa
(santoro@scs.carleton.ca)

1 Introduction

1.1 Distributed Model

There are numerous models for distributed systems, differing from one another on a large number of important factors and parameters. We shall restrict ourselves to systems based on the point-to-point message-passing model: there is no common memory, and a node may communicate directly only with its direct neighbors by exchanging messages. The communication topology of the system can be described by an edge labeled graph where nodes correspond to entities and edges correspond to direct communication links between entities [32]. Let $G = (V, E)$ be the graph; let $E(x)$ denote the set of edges incident to node $x \in V$, and $d(x) = |E(x)|$ the degree of x. Each node has a local label (port number) associated to each of its incident edges. Given a set Σ of labels, a *local orientation* of $x \in V$ is any injective function $\lambda_x : E(x) \to \Sigma$ which associates a distinct label $l \in \Sigma$ to each edge $e \in E(x)$. The set $\lambda = \{\lambda_x : x \in V\}$ of local labeling functions will be called a *labeling* of G, and by (G, λ) we shall denote the corresponding *(edge-)labeled graph.*

Any property of the labeling λ can be exploited to improve the performance of the system, e.g., by reducing the amount of communication required to perform some distributed tasks. The most basic property is *Local Orientation*: the capacity at each node to distinguish between the incident links; by definition, we only consider labelings having this property. Another interesting property is *Edge Symmetry*: there exists a bijection $\psi : \Sigma \to \Sigma$ such that for each $\langle x, y\rangle \in E$, $\lambda_y(\langle y, x\rangle) = \psi(\lambda_x(\langle x, y\rangle))$; ψ will be called the edge-symmetry function. The particular case of edge symmetry called *Coloring* where the edge-symmetry function is the identity (i.e., the labels on the two sides of each edge are the same) is also of interest. For specific labelings in specific topologies, some other properties have been extensively studied; e.g., *Orientation* in ring networks with "left-right" labeling or in tori with the compass labeling.

Without any doubt, the property of labeled graphs which has been shown to have a definite impact on computability and complexity, and whose applicability ranges from the analysis of graph classes to distributed object systems, is *Sense of Direction.* In the rest of the paper, we will provide some introduction and pointers to the relevant results and literature.

1.2 Sense of Direction

Informally, in a labeled a graph (G, λ). the labeling is a (weak) sense of direction if it is possible to understand, from the labels associated to the edges, whether different walks from a given node x end in the same node or in different nodes. A *walk* π in G is a sequence of edges in which the endpoint of one edge is the starting point of the next edge. Let $P[x]$ denote the set of all the walks starting from $x \in V$, $P[x, y]$ the set of walks starting from $x \in V$ and ending in $y \in V$. Let $\Lambda_x : P[x] \to \Sigma^+$ and $\Lambda = \{\Lambda_x : x \in V\}$ denote the extension of λ_x and λ, respectively, from edges to walks; let $\Lambda[x] = \{\Lambda_x(\pi) : \pi \in P[x]\}$, and $\Lambda[x, y] = \{\Lambda_x(\pi) : \pi \in P[x, y]\}$.

A *coding function* f of a graph (G, λ) be any function such that: $\forall x, y, z \in V$, $\forall \pi_1 \in P[x, y]$, $\pi_2 \in P[x, z]$ $f(\Lambda_x(\pi_1)) = f(\Lambda_x(\pi_2))$ iff $y = z$.

Definition 1 *[15]* - Weak Sense of Direction
A system (G, λ), has weak sense of direction *iff there exists a coding function f.*

A *decoding function* h for f is a function that, given a label and a coding of a string (sequence of labels), returns the coding of the concatenation of the label and the string. More precisely, given a coding function f, a decoding function h for f is such that $\forall x, y, z \in V$, such that $\langle x, y \rangle \in E(x)$ and $\pi \in P[y, z]$, $h(\lambda_x(\langle x, y \rangle), f(\Lambda_y(\pi)) = f(\lambda_x(\langle x, y \rangle) \circ \Lambda_y(\pi))$, where $\circ$ is the concatenation operator. We can now define sense of direction:

Definition 2 *[15]* - Sense of Direction
A system (G, λ), has a sense of direction *($\mathcal{SD}$) iff the following conditions hold:*
1) there exists a coding function f,
2) there exists a decoding function h for f.

We shall also say that (f, h) is a sense of direction in (G, λ).

Several instances of sense of direction (*contracted, chordal, neighboring, cartographic*) have been described in [15]; a class of sense of direction called *group* $\mathcal{SD}$ that comprises contracted, chordal, and cartographic $\mathcal{SD}$s has been defined in [48].

It has been shown that $\mathcal{WSD}$ does not imply $\mathcal{SD}$; in fact there are labeled graphs which have $\mathcal{WSD}$ but do not have any $\mathcal{SD}$ [4].

1.3 Sense of Direction and Local Names

The definition of both coding and decoding (and, thus, of sense of direction) can be restated in terms of "translation" capability of local names.

Each node x refers to the other nodes using local names from a finite set $\mathcal{N}$ called name space; let $\beta_x(y)$ be the name associated by x to y. Let us stress that these local names are *not* necessarily identities (i.e., unique global identifiers); in fact, the system could be *anonymous*. The family of injective functions $\beta = \{\beta_x : V \to \mathcal{N} : x \in V\}$ will be called a *local naming* of G.

Let us restate the definition of coding function: a *coding function* f of a graph (G, λ) endowed with local naming β is any function such that: $\forall x, y \in V$,

$\forall \pi \in P[x,y]$, $f(\Lambda_x(\pi)) = \beta_x(y)$. In other words, a coding function translates the sequence of labels of a path from x to y into the local name that x gives to y. Note that while the resulting name is *local* (i.e. x and z might choose different local names for the same node y), the coding function is *global* (i.e., the same for all the nodes).

Let us restate the definition of decoding in terms of local naming. Given a coding function f, a decoding function h for f is any map such that $\forall x, y, z \in V$, with $\langle x, y\rangle \in E(x)$, $h(\lambda_x(\langle x, y\rangle), \beta_y(z)) = \beta(x(z)$. To understand the capabilities of the decoding function in terms of names, consider the situation of node y sending to its neighbor x a message containing information about a node z (see Figure 1. Node z is known at y as $\beta_y(z)$, thus, the message sent by y will contain

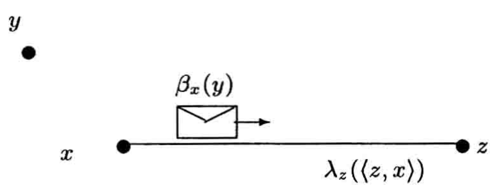

Fig. 1. Communication of information about node y from node x to z.

information about a node called "$\beta_y(z)$". The decoding function allows x to "translate" such a name into its own local name for z $\beta_x(z)$, knowing only the label of the link $\lambda_x(\langle x, y\rangle)$ on which the message is received.

1.4 Sense of Direction and Structural Knowledge

The study of sense of direction is part of the larger investigation on structural knowledge. Informally, the term "structural knowledge" refers to any knowledge about the structure of the system; in general we can distinguish between four distinct types, depending on whether the knowledge is about: 1. the communication topology (T); 2. the labeling (L); 3. the (input) data (D); 4. the status of the entities and of the system (S).

Clearly, sense of direction is of type L. In most cases, its impact, existence and properties will be strongly affected also by the other forms of structural information available in the system. It is thus important to both distinguish among different types and within each type.

Examples of type-T knowledge are: *metric information* (TM): "numeric" information about the network; e.g., size of the network (exact or approximate), number of edges, maximum degree, length of the diameter, girth, etc. *topological properties* (TP): knowledge of some properties of the topology; e.g., the fact that the topology of the network is a regular graphs, a Cayley graph, a (edge-, vertex-, cycle-) symmetric graph, etc. *topological classes* (TC): information about the "class" to which the topology belongs; for example, the fact that the topology is a ring, a tree, a mesh, a tori,etc. *topological awareness* (TA): a knowledge equivalent to the knowledge of the adjacency matrix that describes the topology. *complete knowledge of the topology* (TK): each node knows the adjacency matrix and its position in it (this knowledge is partial when truncated at distance d)

Examples of type-D knowledge are: *unique identifiers*: all input values are distinct; *anonymous network*: all input values are identical; *size*: number of distinct values.

Examples of type-S knowledge are: *system with leader*: there is a unique entity in state "leader"; *reset* all nodes are in the same state.

2 Impact of Sense of Direction on Computability

A large amount of research has been devoted to the study of computability in anonymous systems; i.e., the study of what problems can be solved when there are no distinct identifiers associated to the nodes (e.g., [2, 29, 30, 51]). Clearly, which problems can be solved depend on many factors including the structural properties of the system as well as the amount and type of structural knowledge available to the system entities (the nodes).

The computational power of sense of direction in anonymous systems has been studied in [19] and shown to be linked to the notion of surrounding introduced in [18]. The *surrounding* $N(u)$ of a node u in (G, λ) is a labeled graph isomorphic to (G, λ) where the isomorphism maps each node $v \in V$ to the set $\Lambda[u, v]$ of strings corresponding to all walks from u to v. It has been proven in [19] that, in anonymous distributed systems with sense of direction, the surrounding of a node represents the maximum information that an entity can obtain by message transmissions. Moreover, what is computable in such systems depends on the number of distinct surroundings as well as on their multiplicity (i.e., how many nodes have a given surrounding) [19]. It also depends on the amount of type-T knowledge available.

Let $\mathcal{W}_K(P)$ denote the set of graphs with sense of direction where, in presence of type-T knowledge K, problem P can be solved. In [19] it has been shown that, with weak sense of direction, for every type-T knowledge K and for every problem P, $\mathcal{W}_K(P) \supseteq \mathcal{W}_{TK}(P)$. Since $\mathcal{W}_K(P)$ do not depend on K we will denote it as $\mathcal{W}(P)$.

A powerful implication of this result is the formal proof that, *with sense of direction, it is possible to do shortest path routing even in anonymous networks.*

This result also implies that the obvious knowledge-computability hierarchy

$$\mathcal{W}_{ni}(P) \subseteq \mathcal{W}_{ub}(P) \subseteq \mathcal{W}_{size}(P) \subseteq \mathcal{W}_{TA}(P) \subseteq \mathcal{W}_{TK}(P)$$

collapses, where ni, ub, $size$, TA, TK denote no knowledge, knowledge of an upper bound on the size of the network, knowledge of the exact size, topological awareness, and complete knowledge of the topology, respectively.

Another interesting result is about the relationship between the strongest type-T knowledge and the lowest with sense of direction (i.e., $\mathcal{W}(P)$). Let $D_K(P)$

denote the set of graphs with just local orientation where, in presence of type-T knowledge K, problem P can be solved; clearly, $\mathcal{D}_K(P) \subseteq \mathcal{W}_K(P)$ for every type-T knowledge K. In [19] it has been shown that sense of direction and complete knowledge of the topology have the same computational power; in fact $\mathcal{W}(P) = \mathcal{D}_{TK}(P)$ for every problem P.

The relation between *topological awareness* and sense of direction has been studied with respect to several problems including leader and edge election, and spanning tree construction. It has been shown that, for each of these problems, there exist graphs in which they are not solvable with topological awareness (TA); on the other hand, they can be solved with *any* weak sense of direction; that is, *sense of direction is strictly more powerful than topological awareness.*

3 Impact of Sense of Direction on Complexity

The evidence of the impact that specific sense of direction have on the communication complexity of several problems in particular network topologies has been accumulating in the recent years. Most of the distributed algorithms presented in the literature assume and exploit specific, and mostly implicit, instances of sense of direction within their pre-defined set of assumptions on the structural knowledge (held by the entities) and the attributes of the network. Techniques developed for most of the existing applications are *ad hoc*, network- or application-specific; they do not provide seamless mechanisms for allowing processors to inter-operate with all intended assumptions on knowledge.

A large amount of investigations has been dedicated to the study of a small but typical set that recur in most applications: broadcast ($\mathcal{B}$), depth-first traversal ($\mathcal{DFT}$), spanning tree construction ($\mathcal{SPT}$), minimum finding ($\mathcal{MF}$), election (or leader finding)($\mathcal{LF}$). The next sections provide a schematic view of the most important results.

3.1 Arbitrary Network

A network topology is said to be arbitrary if it is unknown to the entities. This can be understood as a lack of topological awareness. The results on the impact of sense of direction in arbitrary networks are listed below.

Arbitrary Networks

	Broadcast Depth First Traversal	Election Spanning Tree Min Finding
local orientation	$\Theta(e)$	$\Theta(e + n\log n)$ [24, 43]
neighboring $\mathcal{SD}$	$\Theta(\min(e, n^{1+\Theta(1)}))$ [3, 24] Theta(n)	$O(e + n\log n)$
chordal $\mathcal{SD}$	$\Theta(n)$ [36]	$O(e + n\log n)$
any $\mathcal{SD}$	$2n - 2$ [17]	$3n\log n + O(n)$ [17, 36, 35]

Here and in the following n denotes the number of nodes of the network and e the number of edges.

Note that Depth First Traversal can be performed in $O(n)$ with *any sense of direction* even if the graph is *anonymous*. In arbitrarily labeled graphs, this problem requires $\Omega(e)$ messages even if the system is not anonymous; this bound can be easily achieved (e.g., [7]). An improvement in the communication complexity has been shown to exist if each entity has a distinct identity *and* knows the identities of all its neighbors; in this case, this problem can be solved with $O(n)$ messages [44]. This result implies that, in presence of neighboring sense of direction, $\mathcal{DFT}$ can be performed in $O(n)$ messages. Recall that graphs with neighboring labelings are *not* anonymous. A similar reduction has been shown to exist in the presence of chordal sense of direction [16]. It has been proven in [3] that a similar complexity can be obtained for broadcasting without sense of direction only if the size of messages is unbounded.

3.2 Regular Networks

Complete Networks

	Broadcast	Depth First Traversal	Election
local orientation	$(n-1)$	$O(n^2)$	$\Theta(n^2)$ [24]
local orientation with topological awareness	$(n-1)$	$O(n^2)$	$\Theta(n \log n)$ [28]
chordal $\mathcal{SD}$	$(n-1)$	$(n-1)$	$\Theta(n)$ [31, 37, 45]

The results for the Leader Election problem show explicitly that, despite maximum connectivity, there are strict (reachable) lower bounds on what optimal solutions can achieve. A strict and proper classification of the structural information can be deduced. In complete networks also the problem of fault-tolerant leader election has been studied with chordal $\mathcal{SD}$ [35, 38].

Hypercube Network

	Broadcast	Election
local orientation	$O(n)$ [9, 11]	$O(n \log\log n)$ [12]
local orientation and Hamming node-labeling	$O(n)$ [41]	$O(n \log\log n)$
neighboring $\mathcal{SD}$	$O(n)$ [3]	$O(n \log\log n)$
dimensional $\mathcal{SD}$	$O(n)$ (Folklore)	$\Theta(n)$ [14, 42, 47, 50]
chordal $\mathcal{SD}$	$O(n)$ (Folklore)	$O(1)$ [14]

Note that for both Broadcast and Election, a similar complexity can be obtained in all cases (beside neighboring) even if the graph is *anonymous*.

All solutions provided in [14, 47, 42, 50] exploit the implicit region partitioning of the topology and an efficient and implicit scheme to compute and represent shortest paths. Recently, several Broadcast algorithms succeed to exploit the highly symmetric topology without considering processor or link names. Nevertheless, in this case, the best known message complexity for the Election problem is $O(n \log\log n)$ (for large Hypercubes and with a large asymptotic constant) [12].

Chordal Rings

	Broadcast	Election
$\langle 1,2,..,k\rangle_n$ local orientation chordal $\mathcal{SD}$	 $O(n)$ [34] $n-1$ (Folklore)	 $O(n\log n)$ $O(n\log n)$
$\langle S\rangle_n$ $\|S\| < \log n$ local orientation chordal $\mathcal{SD}$	 $O(\|S\|n)$ $n-1$ (Folklore)	 $O(n\log n)$ $O(n)$ [1]
$\langle S\rangle_n$ $\|S\| < \log\log n$ local orientation chordal $\mathcal{SD}$	 $O(\|S\|n)$ $n-1$ (Folklore)	 $O(n\log n)$ $O(n)$ [27]
$\langle S\rangle_n$ $\|S\| < \log^* n$ local orientation chordal $\mathcal{SD}$	 $O(\|S\|n)$ $n-1$ (Folklore)	 $O(n\log n)$ $O(n\log^* n)$ [40]

The results on chordal rings give information on the situation of a complete graph where, at each node, the chordal labeling is known only for a subset (S) of the edges. In this view must also be interpreted the result on double loop [33].

Wake-up

	Time × Bits
local orientation	$\Theta(n^2)$ [17]
chordal $\mathcal{SD}$	$O(n\log^2 n)$ [25]

In all previous results, the communication complexity is measured in terms of number of messages. In the case of synchronous systems, the complexity measure is the trade-off between completion time and amount of transmitted information. A strong example of impact of sense of direction in synchronous systems is the *wake-up* (or, *weak unison*) problem in complete graphs:

3.3 Open Problems

The interplay between symmetry of the graph and consistency of the labels is an important open question. As illustrated in the previous results, there is an empirical evidence that highly regular topologies are difficult to exploit efficiently when the edges are not labeled with consistent structural information.

4 Constructing and Testing Sense of Direction

4.1 Deciding and Testing Sense of Direction

Given a labeled graph (G, λ), how can we verify (decide) whether there is sense of direction?

In [4], the authors have shown that there exist polynomial algorithms for deciding both $\mathcal{WSD}$ and $\mathcal{SD}$. Furthermore, they have shown that deciding weak sense of direction can be done very efficiently in parallel. In fact, considering as model of computation a CRCW PRAM, weak sense of direction is in AC^1 for all graphs, where AC^1 denotes the class of problems which are solvable in time $O(\log n)$ using a polynomial number of processors (in this case n^6 processors). This result is based on a characterization of weak sense of direction in purely combinatorial terms. In fact weak sense of direction is shown to be equivalent to a combinatorial condition on the labeling of a graph called *uniformity*, and checking such a condition has then been proved to be in AC^1. Deciding sense of direction is not as efficient in parallel, in fact it is in AC^1 only for some classes of graphs (k-reducible graphs).

4.2 Constructing Sense of Direction

Since Sense of Direction is known to improve the communication complexity of distributed algorithms, computing $\mathcal{SD}$ as a preprocessing phase in unlabeled topology has been studied in [46, 48] showing that any algorithm computing the common sense of direction for cliques, hypercubes, arbitrary graphs, or tori, exchanges at least $\Omega(e - \frac{1}{2}n)$ messages in a network with n nodes and e edges.

This result is not attractive for these dense topologies (e.g., $\Omega(n^2)$ for cliques and $\Omega(n \log n)$ for the hypercube); although natural algorithms matching the lower bounds have been proposed [46], most of the solutions proposed cannot avoid a complete flooding of the communication links in the network.

The interest is more relevant for topologies with a linear number of edges such as tori or chordal rings of constant degree, as shown in the following table presenting the known message complexity.

	Cost	Type of $\mathcal{SD}$
Complete graph	$\Theta(n^2)$ [46]	chordal $\mathcal{SD}$
Hypercube	$\Theta(n \log n)$ [46]	dimensional $\mathcal{SD}$
Torus ($\sqrt{n} \times \sqrt{n}$)	$\Theta(n)$ [33]	compass SD
Double Loop $\langle 1, \sqrt{n} \rangle_n$	$\Theta(n)$ [33]	chordal $\mathcal{SD}$

Interesting constructions of sense of direction have been given in [5], where the authors have concentrated on the problem of constructing senses of direction that use a small number of labels. Such a number always lies between the maximum degree of the graph (in that case it is minimum) and the number of vertices. By exploiting compositional, algebraic and geometrical techniques, they have shown some constructions of large graphs with sense of direction using few labels; the idea of their constructions is to build large networks by appropriately connecting smaller ones. The authors show that with their approach they obtain minimal senses of direction in Cayley graphs. Their result is a first attempt of developing a set of techniques for constructing sense of direction in graphs by using primitive constructions.

4.3 Coping with Lack of Sense of Direction

The results on constructing sense of direction suggested that it may be more efficient to cope with the lack of sense of direction than to use a pre-processing