DESIGN AND DEVELOPMENT OF ASYNCHRONOUS TERMINATION DETECTION ALGORITHMS FOR DISTRIBUTED COMPUTATIONS

By

MOHINDER NATH GUPTA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

THESIS SUBMITTED
IN FULFILMENT OF THE REQUIREMENTS OF
THE DEGREE OF
DOCTOR OF PHILOSOPHY

TO THE

INDIAN INSTITUTE OF TECHNOLOGY, DELHI

Dec , 1988

DEDICATED TO MY PARENTS

CERTIFICATE

Certified that the thesis entitled **'DESIGN AND DEVELOPMENT OF ASYNCHRONOUS TERMINATION DETECTION ALGORITHMS FOR DISTRIBUTED COMPUTATIONS'** being submitted by Mr. M.N. Gupta, is a record of work carried out by him under my supervision and guidance. The matter embodied in this thesis has not been submitted for the award of any other degree.

Prof. R. K. ARORA

Professor, Department of
Computer Science and Engineering
Head, Computer Services Center
Indian Institute of Technology,
New Delhi.

Dec 15, 1988

ACKNOWLEDGEMENTS

I will always be grateful to my supervisor Professor R. K. Arora. His constructive criticism, personal interest and constant encouragement has really been invaluable during all phases of this work.

I am also extremely grateful to Dr. S. P. Rana for initiating me in this field and for providing constant encouragement and guidance.

My thanks are due to Prof. M. K. Jain, Prof. P. C. P. Bhatt, Prof. S. N. Maheshwari and Prof. S. I. Ahson for help and encouragement. My thanks are also due to the faculty and the other staff of the Deptt. of Computer Science & Engineering and Computer Services Center for their help in many ways.

I would also like to express my gratitude to Prof. H. M. Gupta, Dr. Pramod Kumar, Dr. S. K. Gupta and Dr. Mukul Sinha for fruitful discussions. I am also thankful to my wife KAMAL for her tolerance and understanding during the 'ups and downs' of this work.

M. N. GUPTA

ABSTRACT

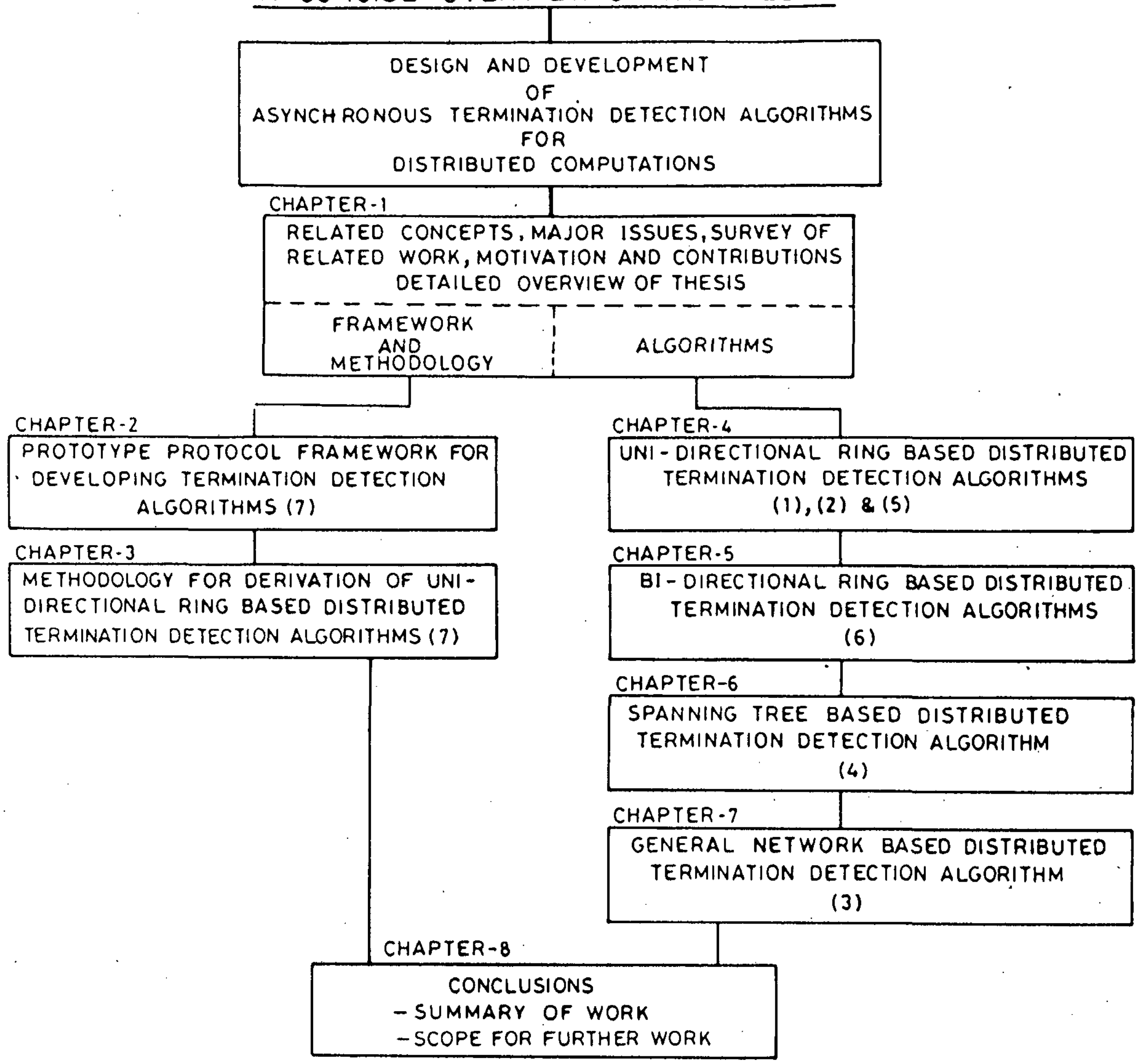
The present thesis deals with the design and development of distributed termination detection algorithms for distributed computations.

The first part of the thesis deals with the development of a generalized prototype framework for the development of distributed termination detection algorithms. Thereafter, based upon the framework, a methodology for the derivation of a class of such algorithms, employing uni-directional control communication around a ring, has been discussed. During the development of the framework, certain issues have been raised in the form of questions and correctness requirements (which the answers to these questions have to meet) stated. Subsequently, stated questions have been resolved by providing sample answers meeting the stated correctness requirements. A combination of an answer from each of the questions represents a distributed termination detection algorithm. Each question is amenable to many different answers; thus making it possible to synthesize a large number of distributed termination detection algorithms.

In the second part of the thesis, several distributed termination detection algorithms, in various processor

network topologies such as ring (uni-directional as well as bi-directional), spanning tree and general network, have been described with correctness arguments. The algorithms do not incur any clock-synchronization overheads. In most of the algorithms, the issuance of a control message, employed for termination detection, from a process has been based upon the concept of neighbouring processes i.e. the set of processes with which the process enters into basic communication for solving the distributed termination problem. The message is issued from a process only when the process itself and all its neighbours have become passive. In addition to this, the majority of the algorithms, developed in the thesis, do not require prior knowledge of any pre-designated process, the size or any global knowledge about the number of processes in the network. Further, the algorithms do not require the determination of any global knowledge prior to the initiation of termination detection procedure.

A CONCISE OVERVIEW OF THE THESIS



(1) Arora, R. K., S. P. Rana, and M. N. Gupta Distributed termination detection algorithm for distributed computations, Information Processing Letters, North-Holland, Vol. 22, No. 6, (1986), pp. 311-314.

(2) Arora, R. K., S. P. Rana, and. M. N. Gupta, Ring based termination detection algorithm for distributed computations, EUROMICRO Journal of Microprocessing and Microprogramming, North-Holland, Vol. 19, No. 3, (1987), pp. 219-226.

(3) Arora R. K. and M. N. Gupta, A general network based algorithm for distributed computations, Journal of Microcomputer Applications, Academic Press, UK, Vol. 10, No. 4, (1987), pp. 297-305.

(4) Arora, R. K. and M. N. Gupta, An algorithm for solving distributed termination problem, EUROMICRO Journal of Microprocessing and Microprogramming, North-Holland, Vol. 22, No. 4, (1988), pp. 263-271.

(5) Arora, R. K. and M. N. Gupta, More comments on "distributed termination detection algorithm for distributed computations", Information Processing Letters, North-Holland, Vol. 29, No. 1, (1988), pp. 53-55.

(6) Arora, R. K. and M. N. Gupta, Bi-directional ring based termination detection algorithms for distributed computations, accepted for publication in IEE Proceedings Part E, Computers and Digital Techniques, UK, 1989.

(7) Arora, R. K. and M. N. Gupta, A generalized framework for deriving ring based termination detection algorithms for distributed computations, to appear in Journal of Microcomputer Applications, Academic Press, UK, Vol. 12, (1989).

LIST OF FIGURES

NOTATIONS ADOPTED IN THE THESIS

B_i	predicate associated with the process p_i.
$bcm(p_i)$	a bit flag employed by a process p_i for keeping track of basic communication of process p_i with that of the process p_j.
$cdmstatus_i(p_j, p_k)$	status associated with the channel through which the detection-message of p_j has been received by the process p_i from a process p_k $(i \neq j,k)$.
$\{CH_i\}$	a set representing the children processes of the process p_i.
$child_i(p_j)$	used for recording the state of a child process p_j (of the process p_i.) in the process p_i.
$\{COMMN_i\}$	the set of communicating neighbours of p_i.
$cstatus_i(p)$	status associated with a channel connecting the process p_i with the process p.
$\{DM_i\}$	the set of identifiers of processes whose detection-messages have been received by the process p_i.

$dmrphn_i(p_j)$	denotes the process identifier to which the process p_i has to convey its own response for the detection-message of p_j.
$dmstatus(p_i)$	status of the detection-message initiated by the process p_i.
$dmstatus_i(p_j)$	used by a process p_i for recording the status of the detection-message of p_j after the process p_i has generated its own response for the message.
DTC	distributed termination condition.
DTP	distributed termination protocol.
f_1	a bit flag employed for accumulation of responses of the processes.
f_2	a bit flag employed for keeping track of an active process encountered by a detection-message during propagation.
$kfm(p_i)$	a boolean variable employed by a process p_i for keeping record about the receipt of I-am-passive/I-am-active messages.
$list(p_i)$	list of process identifiers of neighbour processes (successor and predessor / communicating) of a process p_i.

P	a distributed program comprising of n processes.
p	an arbitrary process out of $p_1, p_2, \ldots, p_n$; p with subscripts i,j,k etc. also would refer the processes out of $p_1, p_2, \ldots, p_n$.
$p_1, p_2, \ldots, p_n$	processes of the distributed program.
p_{i+1}	successor process of the process p_i on the ring.
p_{i-1}	predecessor process of the process p_i on the ring.
p_{n+1}	refers the process p_1 on the ring.
parent(p_i)	denotes parent of the process p_i (in a spanning tree).
PAS	a boolean variable employed by a process p_1 for keeping track of the state when the process becomes passive first time.
{$PHYN_i$}	the set of physical neighbours of p_i.
$pmseq_i(p_j)$	represents the sequence number of the process p_j whose probe-message has been received by the process p_i.

predecessor(p_i)	predecessor process of the process p_i on the ring.
prevpm(p_i)	a bit flag employed by a process p_i for recording status of the forwarded probe message.
procflg(p_i)	a bit flag employed for recording a change of state of the process p_i.
pseq(p_i)	denotes the sequence number of the probe-message already forwarded /issued by the process p_i.
pstatus (p_{ij})	used for recording the change of state of the process p_i since the time p_i receives a detection-message of the process p_j till the time the process p_i conveys its own response for the detection-message of p_j.
RCBDM(p_i)	a boolean variable employed by a process p_i for indicating the truth or falsity of DTC.
$RCBDM_i$(p)	used by a process p_i, for keeping the record of receipt of valid detection-messages from a child process p.
RCTM(p_i)	a boolean variable employed by a process p_i for keeping record of receipt of termination-message.

$rdmrtnd_i(p_j)$	used for recording the information whether p_i has conveyed or still to convey its own response for the detection-message of p_j.
$S1_i$	a set denoting half of the process identifiers in clock-wise direction(on the ring) from the process p_i.
$S2_i$	a set denoting half of the process identifiers in anti-clockwise direction (on the ring) from the process p_i.
$S(p_i)$	a bit flag employed by a non-root process p_i for recording whether the process itself, all its children processes and their respective communicating neighbours have become passive and whether this information has been conveyed by the process p_i to its parent on the tree.
$seq(p_i)$	represents the sequence number maintained in the process p_i.
$seq_i(p_j)$	used by a process p_i for keeping record of sequence number associated with the detection-message of p_j.
$seqdm(p_i)$	denotes the sequence number to be carried along with the detection-message of p_i.

successor(p)	successor process of the process p on the ring.
SNDTM(p_i)	a boolean variable employed by a process p_i for keeping a record of whether a termination-message has been sent by the process.
state(p_i)	denotes the state of the process p_i.
$state_i(p_j)$	employed for storing the state of p_j in the process p_i.
$status_i(p_j)$	used by the process p_i for recording the status of the probe-message received from the process p_j ($i \neq j$).
	does not belong to

NOTE:- Other notations that are only of local significance have been described as and when they appear in the thesis.

CONTENTS