# Contents

Certificate ii

Acknowledgments v

Preface vii

List of Publications x

Contents xi

List of Figures xiv

List of Tables xvi

Abbreviations xvii

Symbols xviii

1 Introduction 1
- 1.1 Background . . . . . 6
- 1.2 Research Goals and Problem Statement . . . . . 9
- 1.3 Motivations . . . . . 10
- 1.4 Contributions . . . . . 12
- 1.5 Thesis Outline . . . . . 14

2 Preliminaries and Literature Review 18
- 2.1 Preliminaries . . . . . 18
  - 2.1.1 Self-stabilizing Conditions . . . . . 19
  - 2.1.2 Failure Detection . . . . . 19
  - 2.1.3 System Models for Leader Election . . . . . 20
  - 2.1.4 Failure Models . . . . . 21
- 2.2 Literature Review . . . . . 23

2.2.1 Algorithms for Regular Network Topology . . . . . . . . . . . 23
2.2.1.1 Algorithms for Ring Network . . . . . . . . . . . . . 23
2.2.1.2 Algorithms for Mesh Network . . . . . . . . . . . . . 25
2.2.1.3 Algorithms for 2D Torus Network . . . . . . . . . . . 27
2.2.2 Algorithms for Arbitrary Network . . . . . . . . . . . . . . . . 30

**3 FRLLE: A Failure Rate and Load-based Leader Election Algorithm for a Ring Network 33**
3.1 System Model . . . . . . . . . . . . . . . . . . . . . . . . . . 34
3.1.1 Assumptions . . . . . . . . . . . . . . . . . . . . . . . . . 34
3.1.2 Definitions . . . . . . . . . . . . . . . . . . . . . . . . . . 35
3.1.3 Types of Message . . . . . . . . . . . . . . . . . . . . . . 38
3.2 Proposed Algorithm . . . . . . . . . . . . . . . . . . . . . . . 38
3.2.1 Proof of Self-stabilization . . . . . . . . . . . . . . . . . . 43
3.2.2 Complexity Analysis . . . . . . . . . . . . . . . . . . . . 46
3.2.2.1 Message Complexity . . . . . . . . . . . . . . . . . . 46
3.2.2.2 Time Complexity . . . . . . . . . . . . . . . . . . . . 47
3.2.3 An Illustrative Example . . . . . . . . . . . . . . . . . . 47
3.3 Empirical Appraisement . . . . . . . . . . . . . . . . . . . . . 51
3.3.1 Experiment Setup . . . . . . . . . . . . . . . . . . . . . . 51
3.3.2 Performance Comparison and Discussion . . . . . . . . . . . 52
3.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

**4 Lea-TN: A Leader Election Algorithm Considering Node and Link Failures in a Torus Network 58**
4.1 System Model . . . . . . . . . . . . . . . . . . . . . . . . . . 59
4.2 Lower Bound . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
4.3 Proposed Methodology . . . . . . . . . . . . . . . . . . . . . . 63
4.3.1 Assumptions . . . . . . . . . . . . . . . . . . . . . . . . . 63
4.3.2 Definitions . . . . . . . . . . . . . . . . . . . . . . . . . . 63
4.3.3 Types of Message and Message Sending Pattern . . . . . . . 66
4.3.4 Description of Lea-TN . . . . . . . . . . . . . . . . . . . . 69
4.3.5 Proof of Self-stabilization . . . . . . . . . . . . . . . . . . 81
4.3.6 Performance Analysis . . . . . . . . . . . . . . . . . . . . 83
4.3.6.1 Message Complexity . . . . . . . . . . . . . . . . . . 83
4.3.6.2 Time Complexity . . . . . . . . . . . . . . . . . . . . 85
4.3.6.3 Space Complexity . . . . . . . . . . . . . . . . . . . . 86
4.3.7 Illustrative Example . . . . . . . . . . . . . . . . . . . . 86
4.4 Empirical Evaluation . . . . . . . . . . . . . . . . . . . . . . 89
4.4.1 Experiment Setup . . . . . . . . . . . . . . . . . . . . . . 89
4.4.2 Performance Comparison and Discussion . . . . . . . . . . . 89
4.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96

**5 Preselection Based Leader Election in Distributed Systems 97**
5.1 System Model . . . 98
5.1.1 Assumptions . . . 99
5.1.2 Definitions . . . 99
5.1.3 Types of Message . . . 100
5.2 Proposed Leader Election Algorithm . . . 101
5.2.1 Illustrative Example . . . 104
5.2.2 Complexity Analysis . . . 107
5.2.2.1 Message complexity . . . 107
5.2.2.2 Time complexity . . . 108
5.2.3 Proof of Self-stabilization . . . 108
5.3 Experimental Analysis and Discussion . . . 110
5.4 Summary . . . 113

**6 Multi-attribute based self-stabilizing algorithm for leader election in distributed systems 114**
6.1 System Model . . . 115
6.1.1 Definitions . . . 116
6.1.2 Assumptions . . . 116
6.2 The Proposed Leader Election Method . . . 117
6.2.1 Identification and prioritization of the quality attributes . . . 117
6.2.2 The leader election algorithm . . . 119
6.2.3 Proof of Self-stabilization . . . 124
6.2.4 Complexity analysis . . . 130
6.2.4.1 Message complexity . . . 130
6.2.4.2 Time complexity . . . 131
6.2.4.3 Bit complexity . . . 131
6.2.5 Illustrative Example . . . 132
6.3 Experiment and result analysis . . . 137
6.4 Summary . . . 143

**7 Conclusion and Future Directions 145**
7.1 Conclusion . . . 145
7.2 Future Research Directions . . . 148

**Bibliography 150**

# List of Figures

1.1 The classical architecture of a distributed system . . . . . . . . . . . 3
1.2 The client-server architecture of a distributed system . . . . . . . . . 6
1.3 The peer-to-peer architecture of a distributed system . . . . . . . . . 7
1.4 Thesis Structure. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

2.1 A $T_{4,4}$ network, every column of which is arranged in decreasing order according to the node Id. . . . . . . . . . . . . . . . . . . . . . . . . . 28

3.1 All the nodes are connected through a bidirectional ring topology. $h, i, j$ and $k$ are four subsequent nodes in this ring. . . . . . . . . . 40
3.2 Illustrative example of the proposed election algorithm when failed leader recovers during the election. . . . . . . . . . . . . . . . . . . . 48
3.3 Illustrative example of the proposed election algorithm when failed leader does not recover during the election. . . . . . . . . . . . . . . . 50
3.4 In the best and worst cases, the total number of exchanged messages of different leader election algorithms. . . . . . . . . . . . . . . . . . . 55
3.5 In the best and worst cases, the total election time of different leader election algorithms. . . . . . . . . . . . . . . . . . . . . . . . . . . . 55

4.1 Details of a $2D$ torus network. . . . . . . . . . . . . . . . . . . . . . . 60
4.2 Building a ternary tree from a $(4 \times 4)$ $2D$ torus network . . . . . . . 62
4.3 Message sending patterns . . . . . . . . . . . . . . . . . . . . . . . . . 68
4.4 Example of the remaining $T_{8,8}$ network, after deleting 48 links from the $T_{8,8}$ network in the manner described above. . . . . . . . . . . . . 78
4.5 Example of the remaining $T_{9,9}$ network, after deleting 60 links from the $T_{9,9}$ network in the manner described above. . . . . . . . . . . . . 79
4.6 Different steps of the election process of the Lea-TN algorithm explained on a $5 \times 5$ $2D$ torus network with some link and node failures. 88
4.7 A $T_{4,4}$ network, every column of which is arranged in decreasing order according to the node ID. . . . . . . . . . . . . . . . . . . . . . . . . 92
4.8 These figures represent the simulation results of the “LEA with One Link Failure [94]”, the “Dynamic LEA with Multi Links Failure [8]” and the “Lea-TN algorithm” without considering the link failures. Figures (a) and (b) represent the graph of the number of messages exchanged in the best case and worst case, respectively, whereas (c) depicts the time taken to conduct the election. . . . . . . . . . . . . 92

4.9 These figures represent the simulation results of the Lea-TN algorithm considering $N-2\sqrt{N}-3$ (if $N$ is odd) or $N-2\sqrt{N}$ (if $N$ is even) link failures. Here, case 1 represents the scenario where one node with the highest leader factor initiates the election, and case 2 represents the scenario where all nodes initiate the election simultaneously. Graphs in (a) represent the number of exchanged messages by the Lea-TN algorithm to elect a leader in case 1 and case 2. The graph in (b) depicts the time taken to conduct the election both in case 1 and case 2. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93

4.10 These figures represent the simulation results of the Lea-TN algorithm considering the different number of link failures. Here, the x-axis represents the number of link failures, and the y-axis represents the number of exchanged messages to elect the leader. In these figures, $N$ refers to the total number of nodes in a system. Graph (a) represents the simulation result where the node with the highest leader factor initiates the election. Graph (b) represents the simulation result where all the nodes initiate the election simultaneously. . . 93

5.1 An arbitrary network consisting of 12 nodes and 15 links. . . . . . . . 100

5.2 $(a)$ is an arbitrary network consisting of 12 nodes and 15 links. $(b)$, $(c)$, $(d)$, $(e)$ and $(f)$ are the several steps of the proposed algorithm to identify the higher potential nodes of the inner layer. . . . . . . . . 107

5.3 The total number of exchanged messages required by different algorithms to complete the election process. (a) and (b) represent the number of exchanged messages in the best case and worst case, respectively. (c) and (d) represent the required time in the best case and worst case, respectively . . . . . . . . . . . . . . . . . . . . . . . . . 112

5.4 The total time required by different algorithms to complete the election process. (a) and (b) represent the required time in the best case and worst case, respectively . . . . . . . . . . . . . . . . . . . . . . . . . 112

6.1 $(a)$ A distributed system consisting of 7 nodes and 9 edges. $(b)$, $(c)$, $(d)$, $(e)$, $(f)$, $(g)$, $(h)$, $(i)$, $(j)$ and $(k)$ are the different steps of the proposed leader election algorithm to elect the leader for the system. 135

6.2 Node and link failures in the network. . . . . . . . . . . . . . . . . . . . 136

6.3 $(a)$ eight replicated servers connected through an arbitrary network topology, $(b)$ ten replicated servers connected through an arbitrary network topology and $(c)$ twelve replicated servers connected through an arbitrary network topology . . . . . . . . . . . . . . . . . . . . . . . 139

6.4 Performance comparison of the PALE algorithm and the proposed algorithm. $(a)$ and $(b)$ show the required time to elect the leader in the best case and worst case, respectively. $(c)$ and $(d)$ show the number of exchanged messages to elect the leader in the best case and worst case, respectively. . . . . . . . . . . . . . . . . . . . . . . . . . . 143

# List of Tables

2.1 The corresponding detour routing for each link failure of a node . . . 28

3.1 Message complexity in terms of O-notation of $N$ and number of exchanged messages in terms of $N$, where $N$ is the total number of nodes. . . . 56

3.2 Time complexity in terms of O-notation of $N$ and number of required time steps in terms of $N$, where $N$ is the total number of nodes. . . . 57

4.1 List for store the information of adjacent node and link failures of a node $i$ . . . 66

4.2 This table represents a comparative study between the Lea-TN algorithm and the other existing algorithms that work on different networks. Comparison parameters are message complexity, time complexity, network topology, total number of links, and maximum link failure tolerability. Here, $N$ is the total number of nodes in the system, and complexities are represented using Big O-notation. . . . 95

5.1 Details of the arbitrary network shown in Figure 5.2 $(a)$ . . . 106

5.2 Details of the different arbitrary networks for the simulation. . . . 111

6.1 Time complexity and message complexity in the best case and worst-case scenarios of the proposed algorithm in different regular network topologies. . . . 132

6.2 The weight given by the experts, the minimum value and the maximum value of the attributes. . . . 139

6.3 Information of all the nodes of the system consisting of 8 nodes and 9 links. . . . 139

6.4 Information of all the nodes of the system consisting of 10 nodes and 13 links. . . . 140

6.5 Information of all the nodes of the system consisting of 12 nodes and 22 links. . . . 140