Contents

Acknowledgments **iii**

Abstract **v**

Resum **vii**

Resumen **ix**

List of Figures **xvi**

List of Tables **xvii**

1 Objectives of the Thesis **1**
1.1 General Objective 1
1.2 Motivation 1
1.3 Partial Objectives 2
1.3.1 To build a prototype that considers the client, the server and the protocol 2
1.3.2 To conduct a performance evaluation study of web prefetching techniques considering real conditions 2
1.3.3 To study real factors that affect the web prefetching limits, and quantify their impact 2
1.3.4 To explore how web prefetching techniques can be improved . . 3
1.3.5 To propose a Low Cost Prediction Algorithm 3

2 Web Prediction and Prefetching **5**
2.1 Introduction 5
2.2 Web Prefetching 6
2.3 Web Prediction Algorithms 7
2.4 Performance Evaluation 9
2.5 Software Implementations 10

3 Delfos. Architecture Prototype with Prefetching Support 13
3.1 Introduction . . . 13
3.2 Framework Architecture . . . 14
3.3 Prediction Engine: *Eprefes* . . . 14
3.3.1 Features . . . 15
3.3.2 Connectivity . . . 16
3.3.3 Serving requests . . . 17
3.3.4 Prediction algorithms . . . 17
3.4 Web Server: *Mod-prefetch* for Apache 2 . . . 21
3.5 Web Client: Web Prefetching in Mozilla . . . 21
3.6 Interrelation Between the Components . . . 23
3.7 Experiments in real environment . . . 23
3.8 Conclusions . . . 26

4 Delfos: Evaluation Environment 27
4.1 Introduction . . . 27
4.2 *CARENA* . . . 27
4.3 Modules for Evaluation . . . 28
4.3.1 Client Pool with *mod-trainer* . . . 28
4.3.2 Statistics gathering with *mod-stats* and *mod-report* . . . 29
4.4 Evaluation . . . 30
4.4.1 Performance indexes . . . 30
4.4.2 Workload Description . . . 31
4.5 Performance Evaluation Using Delfos . . . 33
4.5.1 Experiments . . . 33
4.5.2 System statistics . . . 37
4.6 Conclusions . . . 40

5 Predict at Prefetch: a Technique to Improve Prefetching 43
5.1 Introduction . . . 43
5.2 Predict at Prefetch (P@P) . . . 44
5.3 Experimental results . . . 45
5.3.1 Cost-Benefit . . . 46
5.3.2 Prediction related performance indexes . . . 48
5.3.3 Algorithm Storage Usage . . . 49
5.4 Conclusions . . . 50

6 Theoretical limits of Web Prefetching in a real environment 53
6.1 Introduction . . . 53
6.2 The Perfect Prediction Algorithm . . . 54
6.3 Conditions to Prefetch . . . 55
6.3.1 Maximum Number of Hints per Prediction . . . 55
6.3.2 Browser Idle Time . . . 57
6.3.3 Type of Hints . . . 59

6.4 Enhancing Web Prefetching . . . 61
6.4.1 Predict on Secondary . . . 61
6.4.2 POS and P@P Experimental Results . . . 61
6.5 Performance Comparison With Real Prediction Algorithms . . . 63
6.6 Conclusions . . . 65

7 Referrer Graph: a low-cost prediction algorithm 67
7.1 Introduction . . . 67
7.2 Referrer Graph . . . 68
7.2.1 General Description . . . 68
7.2.2 Theoretical Example . . . 68
7.2.3 Data Structures . . . 69
7.2.4 Learning Process . . . 71
7.2.5 Prediction Process . . . 72
7.2.6 Working Example . . . 72
7.3 Experimental Results . . . 75
7.3.1 Page Latency Saving and Byte Traffic Increase . . . 75
7.3.2 Resource Consumption . . . 76
7.4 Graph Pruning . . . 81
7.4.1 General Issues . . . 81
7.4.2 Proposed Pruning Algorithm . . . 82
7.4.3 Example of RG Pruning . . . 83
7.4.4 Experimental Results of Pruning . . . 85
7.5 Conclusions . . . 91

8 Conclusions and Future Work 93
8.1 Conclusions . . . 93
8.2 Summary of Contributions . . . 94
8.3 Future Work and Open Research . . . 94

A *CARENA* 97
A.1 Introduction . . . 97
A.2 Logging web user requests . . . 98
A.3 Related work . . . 98
A.4 The CARENA Solution . . . 100
A.4.1 Programming Environment . . . 101
A.4.2 Capturing . . . 104
A.4.3 Saving and Importing . . . 105
A.4.4 Replying . . . 105
A.5 Working Example . . . 107
A.6 Conclusions . . . 109

Bibliography 111