

LOCATION DEPENDENT DATA CACHING IN MOBILE ENVIRONMENT

A THESIS

*Submitted in partial fulfilment of the
requirements for the award of the degree
of*
DOCTOR OF PHILOSOPHY
in
COMPUTER SCIENCE AND ENGINEERING

by

AJEY KUMAR



DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE-247 667 (INDIA)

AUGUST, 2007

**©INDIAN INSTITUTE OF TECHNOLOGY ROORKEE, ROORKEE, 2007
ALL RIGHTS RESERVED**



INDIAN INSTITUTE OF TECHNOLOGY ROORKEE ROORKEE


CANDIDATE'S DECLARATION


I hereby certify that the work which is being presented in the thesis entitled **LOCATION DEPENDENT DATA CACHING IN MOBILE ENVIRONMENT** in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy and submitted in the Department of Electronics and Computer Engineering of the Indian Institute of Technology Roorkee, Roorkee is an authentic record of my own work carried out by me during a period from July 2002 to August 2007 under the supervision of Dr. Anil K. Sarje, Professor and Dr. Manoj Misra, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.


(AJEY KUMAR)

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

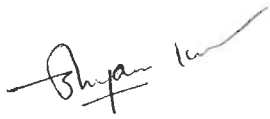

(Manoj Misra)
Professor
(Supervisor)


(Anil K. Sarje)
Professor
(Supervisor)

Date: 17/08/2007

The Ph.D. Viva-Voce Examination of **Mr. AJEY KUMAR**, Research Scholar, has been held on 23.02.2008


Signature of Supervisors


Signature of External Examiner

ACKNOWLEDGEMENTS

At the outset, I wish to express my sincere thanks to my august supervisors Dr. Anil K. Sarje, Professor, and Dr. Manoj Misra, Professor, Department of Electronics and Computer Engineering, IIT Roorkee for their meticulous guidance and pragmatic views of research. Their keen interest, sincere advice and kind help throughout this work had been a regular source of encouragement. Their affectionate treatment and magnanimity made it feasible to bring the present work to conclusion. It was a pleasant experience for me to be surrounded by these noble and affectionate people.

I am thankful to Prof. Kumkum Garg, Prof. Bharoti Sinha, Prof. Arun Kumar, Prof. R.C. Joshi, Prof. D. K. Mehra, Dr. S. Chakravorty and Dr. Durga Toshniwal of Electronics and Computer Engineering Department of the Institute for their invaluable encouragement and support, and above all, the noblest treatment extended by them. I also wish to thank Prof. A. K. Saraf, of Earth Science Department for his help in Global Positioning Satellite System.

There are not enough words to express my gratitude to Dr. Bahuia Zheng, Assistant Professor, Singapore Management Institute, Singapore, for her online help, advices and suggestions on the reported work. I am also thankful to Dr. Chittaranjan Hota, Assistant Professor, BITS Pilani, India, for his invaluable encouragement and support. They truly embody the best of what the technical education is all about.

I acknowledge with deep sense of gratitude to P.I.(s) of Intel's PlanetLab of the Institute, for providing support and facilities for this work. I thank Bhanu, Ranjana, Shukti and Ganesh for their friendly cooperation in PlanetLab. I also express my sense of gratitude to Mr Raj Khati, Mr Anoop Yadav and Mr. Dhanpat Singh for their co-operation, assistance and technical supports in the Software Lab. of the department. I am thankful to the official staff in the department for expediting all the paper work. It is hard to single out any one in this list.

Fellowship awarded by the Ministry of Human Resource and Development (MHRD), Government of India is thankfully acknowledged. I would like to thank the Institute, Intel, ACM SIGAPP, Department of Science and Technology (DST) Govt. of India and Centre for Cooperation in S & T among Developing Societies (CCSTDS), Unit of Indian National Science Academy, New Delhi, India, for providing the Financial Assistance for registration as well as participation in International Conferences in India and abroad.

As I look back, I find some people have left deep impact on my life. Sentiments bring the memories of my association with my teachers Miss Audary, Sister Tresa, Sister Flesia and Mrs Meera Chowdhary from my school days. I am grateful to Mr. V.P.Singh, Officer, Indian Railways, for his *Shramdaan* towards me. I am extremely thankful to Dr. Udai Shanker and his family for having spent their valuable time in encouragement and motivation and for their unconditional love and support. I am very much thankful to my friends, R.B. Patel, Amit Kumar Kohli, Narottam Chand, Ratnadeep Paul Choudhury, Anand M. Lahoti, Balaji Bhosle, Nikunja Bihari Singha, Naveen Shukla, Manish Rana, Varun Singh, Avinash Arora, Sumit, Santosh Viswakarma, Rajwinder Singh and Navdeep Kaur for their inspiration and encouraging attitude to get through the difficult periods of my stay at Roorkee. Grateful thanks are due to Dr. Anibha Sarje and staff of our Institute's Hospital for their care, love and affection shown towards me during my illness. Words fail to express my gratitude to the mess staff, Azad Bhawan for perking me up whenever I was feeling low.

Many thanks to Mr Debmalya Biswas, Ph.D Student, INRIA, France and Mr. Rishi Khare, Software Engineer, ST Microelectronics, India for always being very close and helpful to me throughout my professional studies. Sincere thanks are also due to Mr. Il-Gon Kim, Senior Researcher, Korea Information Security Agency , Korea, for helping me in planning my journey and stay at Seoul.

I owe an immense debt of gratitude to my parents who brought me up to be a confident and well adjusted individual and always believed in my ability to finish the things I set out to do. Their love, support, inspiration and passion for education are highly regarded. Their devotion always kept my motivation up. Words can hardly explain the co-operation extended by my sister and brother-in-law. I am thankful to all those people who I have known all of my life and who remind me where my journey began.

Last but not the least, I am thankful to the almighty who gave me the strength and health for completing the work. I dedicate this work to my grandmother whose soul will feel very much proud of me. She deserves real credit for getting me this far, and no words can ever repay for her.

IIT Roorkee,
August 2007


AJEY KUMAR

LIST OF ABBREVIATIONS

| | |
|---------|---|
| AC | Approximate Circle |
| AFW | Adaptive Invalidation Report with Fixed Window |
| AS | Asynchronous and Stateful |
| AVP | Adaptive Value-based Prefetch |
| BB | Bit-Sequences with Bit Count |
| BOP | Benefit-Oriented Prefetching |
| BS | Base Station |
| BVC | Bit-Vector with Compression |
| CAIDS | CRF Area and Inverse Distance Size |
| CDPD | Cellular Digital Packet Data |
| CEB | Caching-Efficiency-Based |
| CEB_G | Generalized Caching Efficiency Based |
| CEFA | Caching Efficiency with Future Access |
| CEFAB | Caching Efficiency with Future Access Based |
| CEFAB_G | Generalized Caching Efficiency with Future Access Based |
| CIR | Cached Item Register |
| CMIP | Cache-Miss-Initiated Prefetch |
| CQ | Continuous Queries |
| CRF | Combined Recency and Frequency value |
| CSI | Cache State Information |
| DRCI | Dual-Report Cache Invalidation |
| FAR | Furthest Away Replacement |
| FCFS | First Come First Service |
| FMP | Future Movement Path |
| GBVC | Grouped Bit-Vector with Compression |
| GCORE | Grouping with COld update-set REtention |
| GD-LU | Greedy Dual Least Utility |
| GPS | Global Positioning System |
| GSM | Global System for Mobile communications |

| | |
|----------|---|
| HLC | Home Location Cache |
| IR | Invalidation Report |
| ISI | Implicit Scope Information |
| IT | Information Technology |
| LAN | Local Area Network |
| LDD | Location-Dependent Data |
| LDISs | Location-Dependent Information Services |
| LFU | Least Frequently Used |
| LRU | Least Recently Used |
| MARS | Mobility-Aware Replacement Scheme |
| MCs | Mobile Clients |
| MI | Moving Interval |
| Min-SAUD | Minimum Stretch integrated with Access rates, Update frequencies and Cache validation Delay |
| MSC | Mobile Switching Center |
| MSS | Mobile Support Station |
| PA | Probability Area |
| PAID | Probability Area Inverse Distance |
| PAR | Prefetch Access Ratio |
| PE | Polygon Endpoints |
| PPRRP | Prioritized Predicted Region based Cache Replacement Policy |
| PRRP | Predicted Region based Cache Replacement Policy |
| QI | Query Interval |
| SACCS | Scalable Asynchronous Cache Consistency Scheme |
| SAIU | Stretch Access-rate Inverse Update-frequency |
| SDCI | Selection Dual Report Cache Invalidation |
| UIR | Updated Invalidation Report |
| VD | Voronoi Diagram |
| VS | Valid Scope |
| WPRRP | Weighted Predicted Region based Cache Replacement Policy |

LIST OF FIGURES

| Figure No. | Caption | Page No. |
|-------------|---|----------|
| Figure 2.1 | Mobile Computing System Model | 10 |
| Figure 2.2 | Examples of Valid Scopes (Haridwar District) | 18 |
| Figure 2.3 | Taxonomy for Cache Management in Mobile Computing Environment | 22 |
| Figure 2.4 | Data Items with Different Distributions | 29 |
| Figure 3.1 | Client's Movement Path (a) Abstract Model (b) Discrete Model | 39 |
| Figure 3.2 | Algorithm for CEB_G | 42 |
| Figure 3.3 | Case Study | 44 |
| Figure 3.4 | Future Movement Path | 46 |
| Figure 3.5 | e_{EM} with respect to Valid scope v | 46 |
| Figure 3.6 | Algorithm for CEFAB | 48 |
| Figure 3.7 | Algorithm for CEFAB_G | 49 |
| Figure 3.8 | Scope Distributions for Performance Evaluation | 50 |
| Figure 3.9 | Cache Hit Ratio of Invalidation Schemes vs. Query Interval (Scope Distribution 1) | 54 |
| Figure 3.10 | Cache Hit Ratio of Invalidation Schemes vs. Query Interval (Scope Distribution 2) | 55 |
| Figure 3.11 | Cache Hit Ratio of Invalidation Schemes vs. Moving Interval (Scope Distribution 1) | 55 |
| Figure 3.12 | Cache Hit Ratio of Invalidation Schemes vs. Moving Interval (Scope Distribution 2) | 56 |
| Figure 3.13 | Cache Hit Ratio of Invalidation Schemes vs Data Size (Scope Distribution 1) | 56 |
| Figure 3.14 | Cache Hit Ratio of Invalidation Schemes vs Data Size (Scope Distribution 2) | 57 |
| Figure 4.1 | Current Moving Interval | 62 |
| Figure 4.2 | Predicted Region | 63 |
| Figure 4.3 | Predicted Region with Extreme Cases | 64 |

| Figure No. | Caption | Page No. |
|-------------------|--|-----------------|
| Figure 4.4 | Sub Regions within Service Region | 67 |
| Figure 4.5 | Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Query Interval (Scope Distribution 1) | 73 |
| Figure 4.6 | Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Query Interval (Scope Distribution 1) | 74 |
| Figure 4.7 | Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Query Interval (Scope Distribution 2) | 75 |
| Figure 4.8 | Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Query Interval (Scope Distribution 2) | 76 |
| Figure 4.9 | Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Moving Interval (Scope Distribution 1) | 78 |
| Figure 4.10 | Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Moving Interval (Scope Distribution 1) | 79 |
| Figure 4.11 | Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Moving Interval (Scope Distribution 2) | 80 |
| Figure 4.12 | Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Moving Interval (Scope Distribution 2) | 81 |
| Figure 4.13 | Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Cache Size (Scope Distribution 1) | 83 |
| Figure 4.14 | Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Cache Size (Scope Distribution 1) | 84 |
| Figure 4.15 | Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Cache Size (Scope Distribution 2) | 85 |
| Figure 4.16 | Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Cache Size (Scope Distribution 2) | 86 |
| Figure 4.17 | Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Client Speed (Scope Distribution 1) | 90 |
| Figure 4.18 | Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Client Speed (Scope Distribution 1) | 91 |

| Figure No. | Caption | Page No. |
|-------------|--|----------|
| Figure 4.19 | Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Client Speed (Scope Distribution 2) | 92 |
| Figure 4.20 | Cache Hit Ratio of Replacement Schemes (WPPRRP) vs. Client Speed (Scope Distribution 2) | 93 |
| Figure 4.21 | Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Zipf Parameter (Scope Distribution 1) | 94 |
| Figure 4.22 | Cache Hit Ratio of Replacement Schemes (WPPRRP) vs. Zipf Parameter (Scope Distribution 1) | 95 |
| Figure 4.23 | Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Zipf Parameter (Scope Distribution 2) | 96 |
| Figure 4.24 | Cache Hit Ratio of Replacement Schemes (WPPRRP) vs. Zipf Parameter (Scope Distribution 2) | 97 |
| Figure 5.1 | Effect of Frequency of Data Item Access | 101 |
| Figure 5.2 | Effect of Recency of Data Item Access | 102 |
| Figure 5.3 | Spectrum of Recency/Frequency According to Function $F(x) = (\frac{1}{2})^{\lambda x}$ where x is (current time – reference time) | 105 |
| Figure 5.4 | Access History of Data Item i | 106 |
| Figure 5.5 | Effects of Probability of Access P and CRF on Cache Hit Ratio (Scope Distribution 1) | 109 |
| Figure 5.6 | Cache Hit Ratio vs Query Interval (Scope Distribution 1) | 111 |
| Figure 5.7 | Cache Hit Ratio vs Moving Interval (Scope Distribution 1) | 112 |
| Figure 5.8 | Cache Hit Ratio vs Cache Size (Scope Distribution 1) | 113 |
| Figure 5.9 | Cache Hit Ratio vs Client Speed (Scope Distribution 1) | 114 |
| Figure 5.10 | Cache Hit Ratio vs Zipf parameter (θ) (Scope Distribution 1) | 115 |
| Figure 5.11 | Cache Hit Ratio vs Query Interval (Scope Distribution 2) | 116 |
| Figure 5.12 | Cache Hit Ratio vs Moving Interval (Scope Distribution 2) | 117 |
| Figure 5.13 | Cache Hit Ratio vs Cache Size (Scope Distribution 2) | 118 |
| Figure 5.14 | Cache Hit Ratio vs Client Speed (Scope Distribution 2) | 119 |
| Figure 5.15 | Cache Hit Ratio vs Zipf parameter (θ) (Scope Distribution 2) | 120 |

LIST OF TABLES

| Table No. | Caption | Page No. |
|------------|---|----------|
| Table 1.1 | Characteristic of Cache in Various Computing Environments | 4 |
| Table 2.1 | A Taxonomy of Enabling LDIS Technologies | 16 |
| Table 2.2 | Location Determination Technologies and Use of Symbolic/Geometric Coordinates | 16 |
| Table 2.3 | A Classification Framework for LDIS | 17 |
| Table 3.1 | Stepwise Execution for CEB and CEB_G | 44 |
| Table 3.2 | Stepwise Execution with best two in CEB | 45 |
| Table 3.3 | Configuration Parameters and Default Parameter Settings for Simulation Model | 52 |
| Table 4.1 | Sub Regions | 66 |
| Table 4.2 | Configuration Parameters and Default Parameter Settings for Simulation Model | 71 |
| Table 4.3 | Average Improvement of WPRRP-3, PPRRP and PRRP over PAID on Different Mean Query Intervals (Scope Distribution 2) | 77 |
| Table 4.4 | Average Improvement of WPRRP-3, PPRRP and PRRP over PAID on Different Moving Intervals (Scope Distribution 2) | 82 |
| Table 4.5 | Average Improvement of WPRRP-3, PPRRP and PRRP over PAID on Different Cache Size (Scope Distribution 2) | 87 |
| Table 4.6 | Improvement of PRRP over PAID on Different Speed Ranges (Scope Distribution 1) | 88 |
| Table 4.7 | Improvement of PPRRP over PAID on Different Speed Ranges (Scope Distribution 1) | 88 |
| Table 4.8 | Improvement of WPRRP-3 over PAID on Different Speed Ranges (Scope Distribution 1) | 88 |
| Table 4.9 | Improvement of PRRP over PAID on Different Speed Ranges (Scope Distribution 2) | 88 |
| Table 4.10 | Improvement of PPRRP over PAID on Different Speed Ranges (Scope Distribution 2) | 89 |

| Table No. | Caption | Page No. |
|------------------|---|-----------------|
| Table 4.11 | Improvement of WPRRP-3 over PAID on Different Speed Ranges (Scope Distribution 2) | 89 |
| Table 5.1 | Average Improvement of CAIDS over PAID by Varying System Parameters (Scope Distribution 2) | 121 |

ABSTRACT

Mobile computing as compared to traditional computing paradigms enables clients to have unrestricted mobility while maintaining network connections. Due to mobility, *location identification* has naturally become a critical attribute, as it determines the location of mobile users. The ability to pinpoint a mobile user's location due to the advances in global positioning technologies, such as Global Positioning System (GPS), along with the advances in wireless technology has motivated the emergence of a new class of mobile services commonly referred to as Location-Dependent Information Services (LDISs). The promising applications are travel and tourist information system, assistant and emergency system, nearest object searching system and local information access system, to name a few. Users of LDISs face many new challenges inherent to mobile environment. These challenges include limited bandwidth, intermittent connectivity, limited storage, slow CPU speed, low battery power and small user interface. Data management in this paradigm also poses new challenging problems. Thus, sophisticated data management and resource management techniques are needed to enhance the performance of data access in LDISs. The work presented in this thesis is an effort to address these issues by proposing new and efficient cache management schemes for location-dependent data (LDD) in mobile environment. We used geometric model based location identification technique for mobile clients.

First part of the thesis addresses the cache invalidation issues. A cache invalidation scheme maintains data consistency between the client's cache and the server. Unlike the common data, every item of LDD usually has various values, which are termed as data instances of an LDD item. Each instance is only valid within some specific region, which is termed as the Valid Scope (VS) of that data instance. For maintaining consistency of the cached LDD, LDIS stores valid scope of the data item along with its value in the client's cache. The valid scope of a data item is represented and stored as a convex polygon on the server. Downloading valid scope along with data consumes substantial bandwidth. The overhead of storing all end points of the polygon in client's cache is large, so a subset of valid scope is stored that approximates the original valid scope. But storing the subset of valid scope at client's cache reduces the precision of its validity. The mobile client may be shown outside of the valid scope even when actually it is within the original valid scope. Therefore, the problem of selecting the best subset (candidate) of valid

scope that balances the precision and overhead costs becomes crucial. We present a Generalized Caching Efficiency Based (CEB_G) algorithm which selects the best suitable candidate for valid scope that increases caching efficiency and compare its performance with the existing Caching-Efficiency-Based (CEB) algorithm. We then introduce a new metric called Future Access (FA), which takes into account the future movement behavior of the client and based on it propose Caching Efficiency with Future Access Based (CEFAB) algorithm, which selects the best suitable candidate for valid scope using FA. We further propose a generalized CEFAB algorithm (CEFAB_G). A number of simulation experiments are conducted to evaluate the performance of the proposed cache invalidation schemes. The results show that algorithms CEB_G, CEFAB, and CEFAB_G give better performance than CEB for different system settings. Among the proposed algorithms, CEFAB_G gives the best performance. But, computational overhead at the server for CEFAB_G and CEB_G is higher than CEFAB. Moreover, in CEFAB and CEFAB_G, the client has to send additional information to the server, which requires extra computation at the client's end, as compared to CEB_G. Thus, for low resource client CEB_G is preferred. Depending on the resources at the server, choice can be made between CEFAB and CEFAB_G.

Second part of the thesis presents new cache replacement algorithms for location-dependent information services. Due to the limitation of the cache size, it is impossible to hold all accessed data items in the cache. As a result, cache replacement algorithms are used to find a suitable subset of data items for eviction when the cache is full and a new data item is to be inserted into cache. Several location-dependent cache replacement policies have been proposed for LDISs. None of these cache replacement policies are suitable if client changes its direction of movement quite often. The impact of client's anticipated location or region in deciding cache replacement still remains unexplored. Existing cache replacement policies only consider the actual data distance (directional/undirectional), and not the distance based on the predicted region/area where the client can be in near future. When client movement pattern is random, retaining the data items in the direction of user movement and discarding the data items that are in the opposite direction of user movement may not always improve the performance. Therefore our cache replacement policy considers the predicted region of user presence in near future (rather than considering the direction of user movement only) while selecting a data item for replacement. The predicted region is based on the client's current movement pattern. We propose cache replacement algorithms based on the predicted region of user's presence in near future.

These algorithms predict an area in the vicinity of client's current position, and give priority to the cached data items that belong to this area irrespective of the client's movement direction. Based on the predicted region we propose **Predicted Region based Cache Replacement Policy (PRRP)**, **Prioritized Predicted Region based Cache Replacement Policy (PPRRP)** and **Weighted Predicted Region based Cache Replacement Policy (WPRRP)**. In PRRP, data distance is calculated such that the data items within the predicted region are given higher priority than the data items outside the predicted region. In PPRRP, in addition to giving highest priority to the data items within the predicted region, data items nearer to the client's current position are also favored over other data items in the same predicted region. WPRRP divides the whole area into different sub regions: in-direction, out-direction, predicted and non-predicted and then associates different weights with each of these sub regions. By changing these weights this scheme can adapt itself to suit to any situation. We compare our cache replacement policies with other existing cache replacement policies such as Probability Area Inverse Distance (PAID), Furthest Away Replacement (FAR), and Manhattan for LDIS. A number of simulation experiments have been conducted to evaluate the performance of the proposed cache invalidation schemes. The results show that proposed algorithms with different system settings, give much better performance than other policies.

The third part of the thesis considers the effects of recency and frequency of data items accessed, on location-dependent cache replacement. We propose a cache replacement policy namely *CRF Area and Inverse Distance Size (CAIDS)* which uses the Combined Recency and Frequency (CRF) value, valid scope area, data distance and data size of a data item, to select a data item for replacement. Simulation results show that CAIDS has an edge over existing policies. Earlier cache replacement policies for LDIS consider only the recency of data items. But CAIDS considers both recency and frequency factors while deciding the item to be replaced from the cache when the cache is full and new data item is to be inserted in the cache.

Lastly, the contributions made in the thesis in the area of cache management for LDIS have been summarized and scope for future work is outlined.

CONTENTS

| | Page No. |
|---|-----------------|
| Candidate's Declaration | i |
| Acknowledgement | iii |
| List of Abbreviations | v |
| List of Figures | vii |
| List of Tables | xi |
| Abstract | xiii |
| Contents | xvii |
| CHAPTER 1: INTRODUCTION | |
| 1.1 Overview of LDIS | 1 |
| 1.2 Statement of the Problem | 5 |
| 1.3 Contributions of the Thesis | 6 |
| 1.4 Organization of the Thesis | 7 |
| CHAPTER 2: LITERATURE REVIEW | |
| 2.1 Introduction | 9 |
| 2.2 Background | 10 |
| 2.2.1 Mobile Computing System Model | 10 |
| 2.2.2 Location-Dependent Information Services (LDISs) | 13 |
| 2.2.2.1 LDIS Terminology | 14 |
| 2.2.2.2 LDIS Queries | 19 |
| 2.2.3 Data Caching | 20 |
| 2.3 Related Work on Cache Management | 21 |
| 2.3.1 Cache Invalidation and Replacement | 21 |
| 2.3.1.1 Time-Dependent Data | 21 |
| 2.3.1.2 Location-Dependent Data | 28 |
| 2.3.2 Prefetching | 32 |
| 2.4 Summary | 34 |

CHAPTER 3: LOCATION-DEPENDENT CACHE INVALIDATION

| | | |
|-------|---|----|
| 3.1 | Introduction | 37 |
| 3.2 | System Model | 38 |
| 3.3 | Generalized Caching Efficiency Based (CEB_G) Algorithm | 40 |
| 3.3.1 | Case Study | 43 |
| 3.3.2 | Precision versus Computational Complexity | 43 |
| 3.4 | Caching Efficiency with Future Access Based (CEFAB) Algorithm | 45 |
| 3.5 | Simulation Model | 47 |
| 3.5.1 | System | 47 |
| 3.5.2 | Client | 51 |
| 3.5.3 | Server | 51 |
| 3.6 | Performance Evaluation | 52 |
| 3.6.1 | Performance Parameters | 52 |
| 3.6.2 | Performance Metric | 53 |
| 3.6.2 | Comparison of Location-Dependent Invalidation Schemes | 53 |
| 3.7 | Conclusions | 58 |

CHAPTER 4: PREDICTED REGION BASED CACHE REPLACEMENT

| | | |
|-------|---|----|
| 4.1 | Introduction | 59 |
| 4.2 | Cache Replacement Policies based on Predicted Region | 61 |
| 4.2.1 | Predicted Region Based Cache Replacement Policy (PRRP) | 63 |
| 4.2.2 | Prioritized Predicted Region Based Cache Replacement Policy (PPRRP) | 65 |
| 4.2.3 | Weighted Predicted Region Based Cache Replacement Policy (WPRRP) | 66 |
| 4.3 | Simulation Model | 69 |
| 4.4 | Performance Evaluation | 70 |
| 4.4.1 | Performance Parameters | 70 |
| 4.4.2 | Performance Metric | 71 |
| 4.4.3 | Comparison of Location-Dependent Cache Replacement Schemes | 71 |
| 4.5 | Conclusions | 98 |

| | |
|--|---------|
| CHAPTER 5: RECENCY/FREQUENCY BASED CACHE REPLACEMENT | |
| 5.1 Introduction | 101 |
| 5.2 Recency/Frequency based Cache Replacement Policy | 103 |
| 5.3 Performance Evaluation | 107 |
| 5.3.1 Performance Parameters | 107 |
| 5.3.2 Performance Metric | 108 |
| 5.3.3 Comparison of Location-Dependent Cache Replacement Schemes | 108 |
| 5.4 Conclusions | 121 |
| CHAPTER 6: CONCLUSIONS AND SCOPE FOR FUTURE WORK | |
| 6.1 Conclusions | 123 |
| 6.2 Scope for Future Work | 125 |
| REFERENCES | 129 |
| Author's Research Publication | 143 |

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF LDIS

Computing has evolved very rapidly over the past couple of decades. With the advances in wireless technology, wireless devices, such as laptops, mobile phones, personal digital assistants, smartcards, watches and the like, are gaining wide popularity. Their computing capabilities are growing and size is shrinking day by day. The proliferation of wireless networks and smart portable wireless devices has led to the emergence of mobile computing. Today, more and more users can access networked services by using portable devices and easily accessible wireless connections, even when on the move. Initially, the main objective of the wireless networks was to enable mobile units to communicate. Nowadays, these networks support various services and applications ranging from simple network enabled printers to more sophisticated application-level services, e.g., traffic reports, multimedia tourist guides, etc., with a goal to provide information anytime, anywhere, and on any device. However, all the advantages of wireless access by mobile users come with their own costs. Users in mobile wireless environment suffer from intermittent connectivity, narrow bandwidth and limited local resources. Their mobility introduces new challenges. The ability to pinpoint a mobile user's location due to the advances in global positioning technologies, such as Global Positioning System (GPS) [43], along with the advances in wireless technology has motivated the emergence of a new class of mobile services commonly referred as Location-Dependent Information Services (LDISs). *LDISs are IT services for providing information that has been created, compiled and selected or filtered based on the current location of the user or the target object.* The attractiveness of LDIS results from the fact that their participants do not have to enter their location manually, but it is automatically pinpointed and tracked. A traveler may use his laptop to query for weather, traffic, hotels, petrol pumps, adjacent ATM locations, tour guide, shopping guide etc., while traveling. LDIS provides answers depending on the location of the traveler. However, as the queries generated by LDIS users are spatial in nature, sophisticated data and resource management techniques are needed to efficiently provide data to LDIS users. Techniques for improving the capability and quality of wireless environment to provide data to mobile users have become a hot research area in the

academics and industry [23, 72]. Problems in LDIS such as the high network traffic, heavy server workload and long user query delay arise due to mobility of user in the wireless environments and caching is one of the basic methods to overcome these drawbacks.

Cache is a small but fast memory meant to hold frequently accessed data for reuse in near future. Caching has been successfully used for improving performance in traditional operating systems, distributed systems, and Web environments [1,19,22,26,47,61,82,100,122]. In a mobile wireless environment, caching expensive data on client side is desirable to handle data reusability. Since each time the data is requested, one can turn to the cache rather than going to the server and fetching the result. This not only improves query latency but also decreases network access and load at the server, thereby increasing system performance [1,3, 4,5,12, 14, 21, 66, 82, 92,108].

However, in applications like LDIS the same service request may need to be answered with completely different results as the user changes his location or the target moves. Because of this dependency on location, traditional cache management techniques are not well suited for LDIS and the design of an efficient data / cache management strategy for LDIS becomes a major challenge. The main challenges in the management of data / cache for LDIS are summarized below [13,37,50,78,81,82,109]:

- **New types of queries:** Queries raised by LDIS users can be categorized into two types [11,12]. The first one includes queries issued from mobile terminals querying data related to fixed objects (e.g. hotels, gas station, etc). The second category includes the queries issued from mobile or fixed terminals and querying data related to moving objects (e.g. vehicles, planes, peoples). The movement of LDIS users and target objects introduces new issues in maintaining cache consistency.
- **Constraints of wireless environments:** Wireless communication is inherently more interference-prone than wired communication because the surrounding environment interacts with signal, blocks signal paths, and introduces noise and echoes. As a result, wireless connections are of poorer quality and lower bandwidth, are error-prone, and suffer from more frequent disconnections than wired connections. In contrast to wired networks, wireless networks typically have lesser resources, low bandwidth and high latency. Caching should act as a shield to these constraints for users. A good cache

management mechanism should require as little co-operation as possible between the server and the client.

- **Portability of Mobile devices:** Due to portability, mobile devices have a lot of inherent limitations, such as low battery power, slow CPU speed, high risk of data loss, smaller user interfaces, and limited storage. Therefore, all applications in a mobile computing environment including LDIS should take these limitations into consideration and should emphasize on data reusability.
- **Spatial property of Location Dependent Data:** Location-dependent data (LDD) [29,50,82] may show different results for different locations even with the same query, which brings new challenges for cache management. For example, suppose a traveler wants to find out information about nearest hotel in the middle of his journey from Delhi to Haridwar. He issues a query to obtain this information. The answer to this query depends upon the geographical location of the traveler. At one place, for example Roorkee, the answer might be a Roorkee Inn and at another place, near to Haridwar, the response might be Hotel Ganges. In this example data item is Hotel and data values are Roorkee Inn and Hotel Ganges. Due to the spatial property of data item, the data value changes with the movement of client. Hence, the cache management techniques should consider the issues related to LDD.
- **User Mobility:** The ability to change location while connected to a network decreases the validity of information. Certain data, considered static in fixed computing environments, becomes dynamic in LDIS. The information stored in user's cache may become useless if the user changes his location. This introduces a new challenge, requiring new cache management techniques.

Above factors, together with scalability, variable data sizes, heterogeneous access patterns, and frequent data updates, make the design of client cache management a challenge in LDIS. Because of these characteristics of the computing environment, the existing cache management techniques proposed for the distributed-computing arena that operates in a wired environment may not be applicable in LDIS. Ideally, the user should be able to access services in one location and continue to use the equivalent services without interruption when moving to another location or to different wireless network, provided the service is accessible in this new location.

A summary of characteristics of cache in various computing environments is given in Table

1.1

Table 1.1 Characteristics of Cache in Various Computing Environments

| | | Traditional Computing | Web Computing | Mobile Computing | |
|------------------------------------|----------------------------|---------------------------------|---|---|---|
| | | | | Time-Dependent Data | Location-Dependent Data |
| Environment Characteristics | Connectivity | Strong | Strong | Weak/Intermittent | Weak/Intermittent |
| | Data Size | Fixed | Distributed between small(i.e. text)to very large(i.e. video) | Distributed between small(i.e. text)to very large(i.e. video) | Distributed between small(i.e. text)to very large(i.e. video) |
| | Data Access Pattern | Sequential | Random/Zipf's distribution | Random/Zipf's distribution | Random/Zipf's distribution/ Location dependent |
| | Communication Speed | Very high | Fast/Medium | Very slow | Very slow |
| Cache Characteristics | Notion | CPU Cache | Web Cache | Mobile Cache | Mobile Cache |
| | Cache Type | Fast, expensive volatile memory | Nonvolatile memory, i.e disk/ flash memory | Nonvolatile memory, i.e disk/ flash memory | Nonvolatile memory, i.e disk/flash memory |
| | Cache Size | Very small | large | Very small | Very small |
| | Cache Data | Read/Write | Read only | Read only | Read only |
| | Cache Inconsistency | Due to update in cache | Due to update at server | Due to update at server | Due to mobility of client |

In general, cache in mobile client is a small space allocated in the nonvolatile memory such as hard disk or flash memory of the hand held devices. The advantages of using client-side data caching are as follows:

- It can improve data access performance since a portion of queries, if not all, can be satisfied locally.
- It can help to save energy since wireless communication is required only for cache-miss queries.
- It can reduce contention on the narrow-bandwidth wireless channels and reduce workload of the server. This improves the system throughput.
- It can improve data availability when clients are disconnected or weakly connected as queries can be answered from the cached data.

- In LDIS, certain information is same for a given area, for example weather in a city. Caching reduces the need to repeatedly query the server for the same result, thereby saving battery power and wireless bandwidth.

There are three main issues involved in client cache management [3,4,12,22,50,55,56,59,66,82,87,92]:

- **Cache consistency:** A cache consistency scheme, or, as it is called, a cache invalidation scheme maintains consistency between cached data at the client and those at the server.
- **Cache replacement:** A cache replacement policy determines which data item(s) should be dropped from the cache when the free space is insufficient to accommodate an item to be cached.
- **Cache prefetching:** A cache prefetching policy, or called a cache hoarding mechanism, automatically preloads data into the cache for possible future access requests. The main purpose of prefetching is to reduce cache miss costs.

In summary, characteristics of a mobile computing environment in LDISs pose many challenges that do not exist in traditional computing environments. Client-side data caching in LDIS is attractive, as it overcomes, to some degree, constraints such as scarce bandwidth, limited power source and mobility. As very little work has been done in this area, there is a need to devise effective caching mechanisms to handle location-dependent data.

1.2 STATEMENT OF THE PROBLEM

Caching is an effective technique to reduce query latency, bandwidth and power consumption in mobile environment. The spatial property of LDD opens up new challenges and opportunities for data caching research. First, the cached result for a query becomes invalid when the client moves from one location to another. Second, a cache replacement policy that has to identify the data unlikely to be used again needs to consider spatial factors in addition to temporal factors. In LDISs, the chance for a data instance to be used again depends on the size of its valid scope (the area in which the data instance is valid) and the mobility of the user. Therefore, traditional cache consistency and replacement policies can not be used for location-dependent data.

The main objective of the present research work is “*to investigate and propose client-side cache management techniques for location-dependent data in mobile environment*”. The above problem can be further subdivided into smaller objectives as follows:

1. To investigate types of queries issued by the mobile client’s.
2. To explore client’s movement pattern.
3. To investigate the various location models used in determining user’s position in LDIS.
4. To examine the existing cache management strategies for LDIS.
5. To propose new cache invalidation strategies for LDIS and evaluate their performance.
6. To propose and evaluate new cache replacement policies for LDIS.

1.3 CONTRIBUTIONS OF THE THESIS

This thesis attempts to address the *issues concerning the client-side caching for location-dependent data in mobile environment* and proposes new techniques for maintaining cache consistency and selecting data item for cache replacement. The main contributions of the thesis are as follows:

- *Location-dependent cache invalidation.* For maintaining consistency of the cached Location Dependent Data (LDD), LDIS stores valid scope of the data item along with its value in the client’s cache. Valid scope of the data item is represented and stored as a polygon (convex and irregular) on the server. The overhead of storing all end points of the polygon in client’s cache is large, so a subset of valid scope is stored that approximate the original valid scope. But storing the subset of valid scope at client’s cache reduces the precision of its validity. The client may be shown outside of the valid scope even when actually he is within the original valid scope. Therefore, the problem of selecting the best subset of valid scope becomes crucial. In this thesis, we propose a Generalized Caching Efficiency Based algorithm (CEB_G) which selects the best suitable candidate for valid scope. We also propose a new Caching Efficiency with Future Access Based (CEFAB) cache invalidation policy that tries to improve the performance by speculating the user’s future access based on his movement pattern.

- *Location-dependent cache replacement.* Due to cache size limitation, the choice of cache replacement technique to find a suitable subset of items for eviction from cache becomes important. We propose a new Predicted Region based Cache Replacement Policy (PRRP) for the replacement of location-dependent data in mobile environment. The proposed policy uses a predicted region based cost function to select an item for eviction from cache. The policy selects the predicted region, based on client's movement and uses it to calculate the data distance of an item. This makes the policy adaptive to client's movement pattern unlike earlier policies that consider the directional / non-directional data distance only. We also propose Prioritized Predicted Region based Cache Replacement Policy (PPRRP) and Weighted Predicted Region based Cache Replacement Policy (WPRRP) based on it.

It has been shown that Combined Recency and Frequency (CRF) value gives better results than the traditional Least Recently Used (LRU) and Least Frequently Used (LFU) cache replacement policies [25]. We propose a cache replacement policy known as CRF Area and Inverse Distance Size (CAIDS) which uses CRF, valid scope area, data distance and data size of a data item, while selecting an item for replacement. Simulation results show that the proposed cache management policies perform better than the existing policies in terms of cache hit ratio.

1.4 ORGANIZATION OF THE THESIS

This thesis is organized as follows. Chapter 2, provides an introduction to the subject of LDIS and a review of related work. The background includes the mobile computing system model and a brief description of its characteristics. Formal definitions of *location-dependent information services (LDISs)* and the concept of valid scope is also given. Finally, the work done on cache invalidation, cache replacement and prefetching for time-dependent data as well as location-dependent data is reviewed.

In Chapter 3, a generalized cache invalidation algorithm is presented. A new cache invalidation scheme, CEFAB is also described in this chapter, which tries to improve the performance by speculating the user's future accesses based on his movement patterns.

Chapter 4, deals with the design of cache replacement policies that use a predicted region based cost function to select data items for eviction from cache. These policies select the

predicted region based on client's movement and use it to calculate the data distance of a data item. This makes the policy adaptive to client's movement pattern unlike earlier policies that consider the directional/non-directional data distance only. PRRP, PRRP and WPRRP have been explained in detail.

Chapter 5, describes recency/frequency based location-dependent cache replacement policy, CAIDS, which considers both recency and frequency of data item for eviction from cache.

Finally, Chapter 6, concludes the thesis by summarizing our work and suggests areas/issues which can be explored in future. A list of the author's research publication is also given at the end of the thesis.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

The emergence of powerful portable computers, along with advances in wireless communication technologies, has made mobile computing a reality. Mobility arises naturally in wireless mobile computing since as mobile users move, their point of attachment to the fixed network changes. Mobile (or wireless) applications, despite being potentially very different in nature from each other, all share a common characteristic that distinguishes them from their wireline counterparts: they allow their users to move around while remaining capable of accessing the network and its services. Mobility and portability has created an entire new class of applications and possibly new massive markets combining personal computing and consumer electronics. Among the applications that are finding their way to the market of mobile computing - those that involve data management - hold a prominent position [23,31,37,78]. Due to mobility, *location identification* has naturally become a critical attribute, as it opens the door to a world of applications and services that were unthinkable only a few years ago. The term *Location-Dependent Information Services (LDISs)* has been coined to group together all those applications and services that utilize information related to the geographical position of their users in order to provide value-added services to them. In the mobile wireless computing environment, massive number of low powered mobile/wireless devices query databases over the wireless communication channels. These devices act as clients (data consumers), while servers having databases typically reside on the wired network (mobile data management, typically based on the client/server model). Mobile clients may often be disconnected for prolonged periods of time to save battery power. They can also frequently relocate between different cells and connect to different data servers at different times [72,78,80,81,109,113].

In the past few years, there has been a tremendous surge of research in the area of data management in mobile computing. This research has produced interesting results in areas such as data dissemination over limited bandwidth channels, location-dependent querying of data, and advanced interfaces for mobile computers to provide location-dependent information services to mobile users [23,29,62,78,80,89,91,105,109]. This chapter is an effort to survey these techniques and to classify this research in a few broad areas.

2.2 BACKGROUND

2.2.1 MOBILE COMPUTING SYSTEM MODEL

A mobile computing system [13,23,31,50,74,75,81,109] is usually made up of a server, moving clients, and a wireless connection between them (see Figure 2.1). The geographical area is divided into small regions, called cells. Each cell has a *Base Station* (BS) or *Mobile Support Station (MSS)* augmented with wireless interfaces and a number of *Mobile Clients* (MCs). Inter-cell and intra-cell communications are managed by the MSSs. The MCs communicate with the MSS by wireless links within its radio coverage area. An MC can move freely from one location to another within a cell or between cells while retaining its network connection. An MC can either connect to a MSS through a wireless communication channel or disconnect from the MSS by operating in the *doze* (power save) mode. The MC queries the database servers that are connected to a wired network. The wireless channel is logically separated into two sub channels: *uplink channel* and *downlink channel*. The uplink channel is used by MCs to submit queries to the server via an MSS, while the downlink channel is used by MSSs to disseminate information or to forward the answers from the server to the target client.

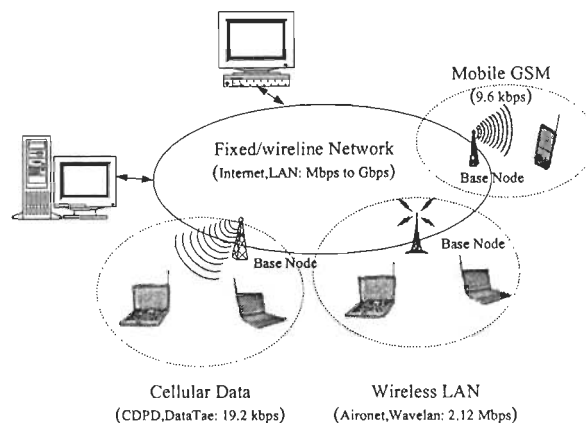


Figure 2.1 Mobile Computing System Model

The mobile computing platform can be effectively described under the *client/server* paradigm [46]. A data item is the basic unit for update and query. MCs only issue simple requests to read the most recent copy of a data item. There may be one or more processes running on an MC. These processes are referred to as clients (we use the terms MC and client/users interchangeably). In order to serve a request from a client, the MSS needs to communicate with the database server to retrieve

the data items. Since the communication between the MSS and the database server is through wired links and is transparent to the clients (i.e., from the client's point of view, the MSS is the same as the database server), we also use the terms MSS and server interchangeably. The system provides location dependent services to mobile clients. The geographical area covered by the system is referred to as the service area. Moreover, the data item value is different from data item. Data item value for a data item is an instance of the item valid for a certain geographical region. So, a data item can show different values when clients at different locations query it. For example, "restaurant" is a data item, and the data values for this data item vary depending on the location of query i.e. point at which the query "*Tell me the nearest restaurant*" was issued by a mobile client.

Characteristics of Mobile Computing System

Although a wireless network with mobile clients is essentially a distributed system, there are some characteristic features that make mobile computing system unique and a fertile area of research.

These features are summarized below [8,13,23,28,31,37,50,60,72,78,80,81,108,109]:

- **Constrained and unreliable wireless communications.** The radio spectrum used for wireless communications is inherently scarce. For example, GSM operates only between 880 MHz and 960 MHz. The data rates for a single wireless channel is limited, varying from 1.2 Kbps for a slow paging channel, 19.2 Kbps for CDPD, to about 11 Mbps for a wireless LAN. Furthermore, wireless transmission is error-prone because the surrounding environment interacts with the signal, blocks signal paths, and introduces noise and echoes. Data might be corrupted or lost due to many factors such as signal interference and obstruction by tall trees and buildings. As a result, wireless connections are of poor quality.
- **Limited power source.** The battery power of wireless portable devices is limited, ranging from only a few hours to about half a day with continuous use. Moreover, only a modest improvement in battery capacity of 20-30% can be expected over the next few years [15,31]. It is also worth noting that sending data consumes much more power than receiving data. For example, a WaveLan card consumes 1.7 W when the receiver is "on" but 3.4 W when the transmitter is "on"
- **Frequent disconnections.** To save energy or connection costs, mobile clients frequently disconnect themselves from the network and are kept in a weak connection status.

Furthermore, due to unreliable wireless communication links, mobile clients are also often disconnected by failure.

- **Asymmetric communication.** The bandwidth in the downstream direction (servers-to-clients) is much greater than that in the upstream direction (clients-to-servers). Stationary servers have powerful broadcast transmitters while mobile clients have little or no transmission capability [23]. Even in the case of an equal communication capacity, the data volume in the downstream direction is estimated to be much greater than that in the upstream direction [97]. Example include Information dispersal systems for time-sensitive information such as stock prices, weather information, traffic updates, factory floor information, etc.
- **Unrestricted mobility.** Mobile users can move from one location to another freely while maintaining network connectivity, which enables their almost unrestricted mobility. Locations and movements of mobile users are therefore hard to predict. The ability to change location while connected to a network increases the volatility of some information. Certain data, considered static in stationary computing environments, become dynamic in mobile computing scenarios. Therefore, the management of location-dependent information is a new challenge. Mobility also makes the network address of a mobile computer change dynamically, which is not supported by traditional wired networking. Thus, new protocols need to be devised.
- **Heterogeneous network.** In contrast to most stationary computers, which stay connected to a single network, mobile computers encounter more heterogeneous network connections in several ways. First, as they leave the range of one network transceiver and switch to another, they may also need to change transmission speeds and protocols. Second, in some situations a mobile computer may have access to several network connections at once, for example, where adjacent cells overlap or where it can be plugged in for concurrent wired access. Also, mobile computers may need to switch interfaces, for example, when going between indoors and outdoors. Infrared interfaces cannot be used outside because sunlight drowns out the signal. Even with radio frequency transmission, the interface may still need to change access protocols for different networks, for example, when switching from cellular coverage in a city to satellite coverage in the country. This heterogeneity makes mobile networking more complex than traditional networking.

- **Limited client capacities.** Portable wireless devices are restricted by weight, size, and ergonomic considerations, which limits their capacities for CPU cycle, storage, and display. Because of their portability, mobile devices have a lot of inherited limitations, such as low power, high risk of data loss, small user-interfaces, and limited storage. Therefore, all the applications in a mobile computing environment should take these limitations into account.
- **Wireless Data deliverables.** There are two fundamental information delivery methods for wireless data applications: *point-to-point access* and *data broadcast* [49,52]. Compared with point to point access, data broadcast is a more attractive method for several reasons [13,17,20,50,51,52,57,84]:
 - i. A single broadcast of a data item can satisfy all the outstanding requests for that item simultaneously. As such, broadcast can scale up to an arbitrary number of users.
 - ii. Data broadcast can take advantage of the large downlink capacity when delivering data to clients, and
 - iii. A wireless communication system essentially employs a broadcast component to deliver information. Thus, data broadcast can be implemented without introducing any additional cost.

In brief, the specific features of wireless scenarios introduce a lot of new research challenges that do not exist in the traditional wired environments. As computing is becoming increasingly pervasive, information services are getting more and more complex and challenging

2.2.2 LOCATION-DEPENDENT INFORMATION SERVICES (LDISs)

Location-dependent information services (LDISs) are services that answer queries based on the locations with which the queries are associated, normally the locations where the queries are issued. The emergence of LDISs is the result of advances and convergence in high-speed wireless networks, personal portable devices, and location identification techniques. There are a variety of promising applications with LDISs, including:

- Tourism information, e.g., finding nearby hotels/motels, nearby restaurants, local attractions, and local maps.
- Driving information, e.g., downloading local traffic report and finding the nearest gas station.

- Health and entertainment information, e.g., finding local cinemas, clinics, and health centers.
- Emergency information, e.g., finding the nearest hospital, police, etc.
- Targeted advertisement: advertisements are delivered to the client based on its current location, e.g., when the client is passing by a supermarket, on-sale information is provided.

Although, LDISs exist in the traditional computing environments (e.g. Yahoo Local, MSN's yellow page, etc.), but their greatest potential would be utilized in a mobile/pervasive computing environment [50], where users enjoy unrestricted mobility and ubiquitous information access. Take a traveler on the road as an example. Travel plan can be done in the traditional way, e.g., access to web, online databases, travel agent, secretary, etc. Once on the road, many problems are situational. Fastest transportation may depend on the next bus departure time, traffic situation, and time of day. There are many uncertainties which may arise, e.g., relocation of bus terminals, change of schedules, and so on. LDIS may be helpful in many of these situations.

2.2.2.1 LDIS Terminology

In this subsection, we defined the terminologies that are frequently used with LDISs. They are as follows:

Location: A location is a geographical area. Location may be expressed with various granularities. For instance, it may be specified as a longitude-latitude pair, a city, a country or a region covered by a cell or a group of cells when referring to the cellular architecture in wireless communications.

Location-Dependent Data (LDD): Location-dependent data (LDD) refers to data whose values depend on location i.e. a data item is location-dependent if it takes on different values based on its location [10,27, 82,104]. For example, a location-dependent data item may have some value **a** in region **A** and some other value **b** in another region **B** at the same time. Both values are correct in their respective regions and represent the same data object. The value **b** may be related to value **a** by some functional mapping which may depend on factors such as the distance between the two regions or the two values may be independent. We assume that values in a location remain the same unless explicitly updated. We have not considered items whose value changes continuously with time.

Location Identifying Models: Location plays a central role in LDISs. A location needs to be specified explicitly or implicitly for any information access. A location model depends heavily on the underlying location identification technique employed in the system. The available mechanisms for identifying locations can be categorized into two models [14,24,50,55,56]:

- **Geometric Model:** A location is specified as an n -dimensional coordinate (typically $n = 2$ or 3), e.g., the latitude/longitude pair in the *GPS* [43]. GPS data resolves the latitude and longitude of a mobile user on the Earth's surface using a satellite-based triangulation system. The main advantage of the geometric model is its compatibility across heterogeneous systems. Geometric coordinates also provide a natural primitive to support a range of spatial queries such as containment. For example, if we define a polygonal region by specifying the vertices of the polygon in geometric coordinates, it is very easy to compute if a particular (x,y,z) location coordinate lies inside or outside the polygon. Similarly, geometric coordinates are especially useful in determining proximity measures in the physical world, e.g., determining the geographically closest cafeteria or restaurant. However, providing such fine-grained location information may involve considerable cost and complexity. The basic problem with geometric data is that it cannot reflect the notion of containment without numerical computation.
- **Symbolic Model:** The location space is divided into disjointed zones, each of which is identified by a unique name. Examples are Cricket [84] and the cellular infrastructure. For example, the PCS/cellular systems identify the mobile phone using the identity of its current serving mobile switching center (MSC); in the Internet, the IP address associated with a mobile device (implicitly) identifies the subnet/domain/service provider to which it is currently attached. Compared to the geometric model, the symbolic model normally has a coarser location granularity and is cheaper to deploy. Also, being discrete and well-structured, location information based on the symbolic model is easier to manage.

It is important to realize that any geometric coordinate can be easily converted into a symbolic namespace simply by treating the physical coordinate space as the symbolic namespace. For example, we could treat each PIN code in a country as a separate symbol and

map each geometric coordinate to its PIN code. The geometric representation may be more natural for applications where the physical coordinates of the mobile node is of principle interest (such as the E-911 [24] initiative for emergency response). Also, applications that require extremely fine-grained location may find the symbolic namespace to be unacceptably coarse. For example, in follow-me application wishing to activate wall-mounted displays based on user head motion, merely obtaining the ID of the nearest Active Badge sensor may not be good enough. The geometric and symbolic location models have different overheads and levels of precision in representing location information. The appropriate location model to be adopted really depends on its applications. Table 2.1, Table 2.2 and Table 2.3 gives the various technologies and applications of LDIS [24,50].

Table 2.1 A Taxonomy of Enabling LDIS Technologies

| Technology Category | Technology | Coverage Range | Accuracy Support | Application Environment |
|---------------------------------------|-------------|----------------|------------------|-------------------------|
| Mobile Network Dependent Technology | Cell-ID | Long | Low | Indoor/Outdoor |
| | TOA | Long | Medium | Indoor/Outdoor |
| | OTD | Long | Medium | Indoor/Outdoor |
| Mobile Network Independent Technology | GPS / A-GPS | Long | High | Outdoor |
| | BLUETOOTH | Short | High | Indoor |
| | WLANs | Short | Low to Medium | Indoor |
| | RFID | Short | High | Indoor |

Table 2.2 Location Determination Technologies and Use of Symbolic/Geometric Coordinates

| Product/Research Prototype | Primary Goal | Underlying Physical Technology | Techniques Employed | Location Representation |
|----------------------------|----------------------------|--------------------------------|--|-------------------------|
| GPS | Outdoor tracking | RF | Triangulation | Geometric |
| Active Badge | Indoor tracking | Infrared | Vicinity-based reporting | Symbolic |
| Active Bats | Follow-me indoor computing | Ultrasonic | Paging | Geometric |
| Cricket | Indoor location tracking | RF and ultrasonic | Location updates | Geometric/Symbolic |
| RADAR | Indoor location tracking | 802.11 WLAN | Triangulation, location updates | Geometric |
| 3D-iD | Indoor location tracking | Active RFID | Triangulation, paging | Geometric |
| LANDMARC | Indoor location tracking | Active RFID | Multilateration, minimum distance estimation | Geometric |
| Smart floor | Indoor user tracking | Foot pressure | Location updates | Geometric |

Table 2.3 A Classification Framework for LDIS

| Services | Examples | Accuracy Needs | Application Environments | Corresponding Enabling Technologies | Corresponding Facilitating Technologies | Service Charging Scheme |
|-------------|---------------------------------|----------------|--------------------------|-------------------------------------|---|-------------------------|
| Emergency | 911 calls | Medium to High | Indoor / Outdoor | TOA / OTD / A-GPS | | Free-of-charge |
| | Automotive Assistance | Medium | Outdoor | TOA / OTD / A-GPS | | User-charged |
| Navigation | Directions | High | Outdoor | | GIS | User-charged |
| | Traffic Management | Medium | Outdoor | TOA / OTD / A-GPS | WAP / GIS | User-charged |
| | Indoor Routing | High | Indoor | BLUETOOTH / WLANs / RFID | WAP / GIS | Free-of-charge |
| | Group Management | Low to Medium | Outdoor | CELL-ID / TOA / OTD / A-GPS | | User-charged |
| Information | Travel services | Medium to High | Outdoor | TOA / OTD / A-GPS | WAP / GPRS / UMTS / GIS | User-charged |
| | Mobile yellow Pages | Medium | Outdoor | TOA / OTD / A-GPS | WAP / GIS | User-charged |
| | Infotainment Services | Medium to High | Outdoor | TOA / OTD / A-GPS | WAP / GPRS / UMTS / GIS | User-charged |
| Advertising | Banners, Alerts, Advertisements | Medium to High | Outdoor | TOA / OTD / A-GPS | WAP / GPRS / UMTS / GIS | Free-of-charge |
| Tracking | People Tracking | High | Indoor / Outdoor | OTD / A-GPS | | User-charged |
| | Vehicle Tracking | Low | Outdoor | CELL-ID | GIS | Corporate User-charged |
| | Personnel tracking | Medium | Outdoor | TOA / OTD / A-GPS | GIS | Corporate User-charged |
| | Product Tracking | High | Indoor | BLUETOOTH / RFID | | Corporate User-charged |
| Billing | Location-sensitive billing | Low to Medium | Indoor / Outdoor | CELL-ID / TOA / OTD | | Free-of-charge |

Valid Scope: Unlike the common data, every data item of LDD usually has various values, which are defined as *data instances* of an LDD item. Each instance is only valid within some specific region, which is termed as the *Valid Scope (VS)* of that data instance. Take PIN code as an example (see Figure 2.2 (a)). For the query "Give me the PIN Code of my current location", clients in Roorkee will receive 247667, while those in Haridwar will get 249403. PIN code can be regarded as an LDD item, and its VSs are defined by the PIN code boundary map.

The main benefit of knowing the VS for a query is that once a query is answered, the same query does not have to be asked again as long as the user stays within the VS. This can be used as a spatial caching scheme to significantly reduce the number of queries submitted and is mandatory for the support of continuous queries [12,13,14,50,58]. VSs can be defined differently for various types of applications. The VS of PIN code is defined by the postal office, whereas the VS of nearest-neighbor queries is defined by the *Voronoi Diagram (VD)*. Formally, given a set of points $O = \{o_1, o_2, \dots, o_n\}$, $V(o_i)$, the *Voronoi Cell (VC)* for o_i , is defined as the set of points q in the space such that $dist(q, o_i) < dist(q, o_j), \forall j \neq i$. That is, $V(o_i)$ consists of the set of points for which o_i is the nearest neighbor.

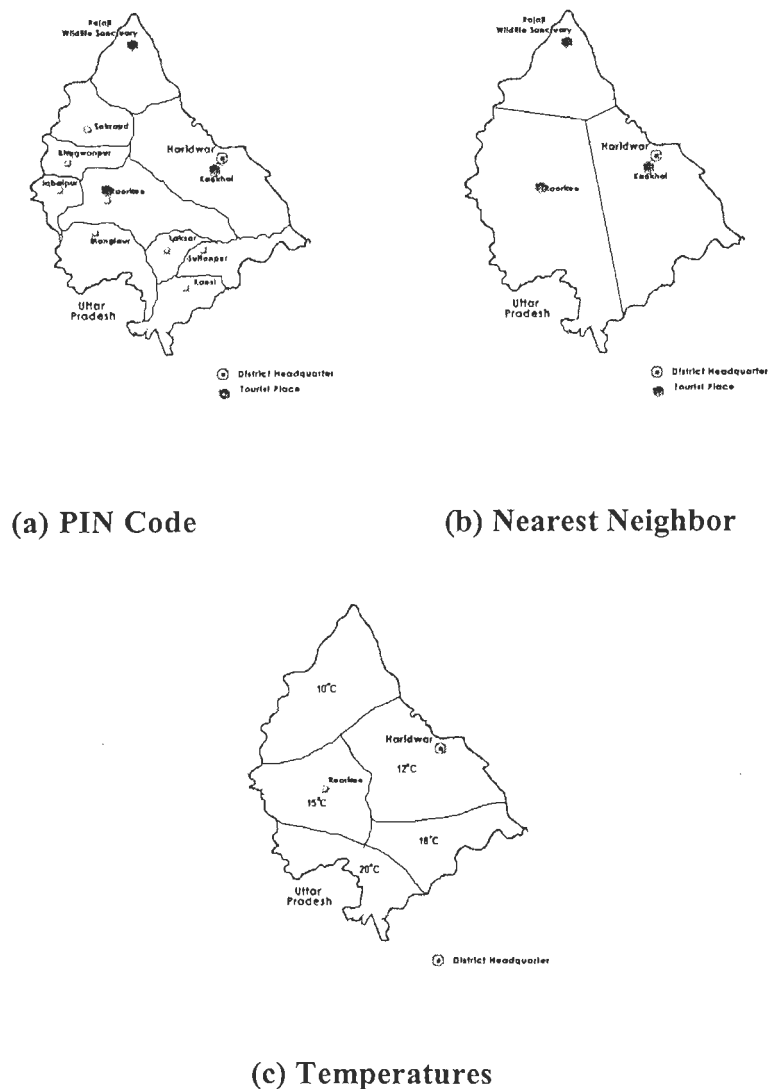


Figure 2.2 Examples of Valid Scopes (Haridwar District)

As shown in Figure 2.2(b), there are three Tourist Places in Haridwar District. The district is partitioned into three parts by the VD. For this simple example, it is a simple perpendicular bisector between the two tourist places. For clients in the Roorkee region, tourist place in Roorkee is the nearest. VS can also be based on temperature in the geographical region. For example, when a user asks for the temperature at his current location (i.e. Roorkee), the answer will be 15°C (see Figure 2.2 (c)).

VSs can also be defined symbolically. For example, when a user moves across cells in a cellular system, he (or his mobile phone) may ask for the *cell id* of his current cell. The valid scope for these types of queries is the radio coverage of the cells.

2.2.2.2 LDIS Queries

According to the mobility of the clients and the data objects queried by the clients, location-dependent queries can be classified into three types [2,12,13,14,50,110]

- *Mobile clients querying static objects:* Queries like “Tell me where the nearest gas station is” and “where the nearest restaurant is?” are popular queries in real-world applications. In general, the clients submitting this kind of queries are mobile and data objects are fixed. The main challenge of this type of queries is how to get the locations of clients and also guarantee the validation of the results when the client keeps moving during the query evaluation process. Queries such as “Report all the available hospitals within a 500 meter radius” are an extensions of this type of query.
- *Stationary clients querying moving objects:* An example of this type of query is “Report all cars that pass gas station A in next 10 minutes”. Here, gas station A is static and moving cars are objects being queried.
- *Mobile clients querying mobile objects:* In this case, both, the clients submitting the queries and the data objects are continuously moving. For example query of type “Tell me all the cars that will pass me after 20 minutes”.

All the queries listed above can also contain query about location-independent attributes, such as: “Tell me the nearest restaurant providing Chinese food”. Since these queries can be broken down into two parts: one for location-dependent information and the other for location-independent attributes, we only consider queries about location-dependent information, as the others can be handled by traditional query-processing methods.

A location-dependent query becomes difficult to answer when it is submitted as a continuous query. For example, a client in a moving car may submit the query: “Tell me the room rate of all the hotels within a 500 meter radius of me”, continuously in order to find cheap hotel. Since the client keeps moving, the query result becomes time-sensitive in that each result corresponds to one particular position and has a valid duration because of location dependency. The representation of this duration and how to transmit it to client are the major focuses of Continuous Queries (CQ). Sistla et. al. employed a tuple $(S, begin, end)$ to bound the valid time duration of the query result [7,9]. Based on this method, they also developed two approaches to transmit the results to the client: an *immediate approach* and a *delayed approach*. The former transmits the results immediately after they are computed. Thus, some later updates may cause changes to the results. The latter, transmits S only at time *begin*, so the results will be returned to the client periodically, thus increasing the wireless network burden. To alleviate the limitations of both approaches, new approaches, such as the Periodic Transmission (PT) Approach, the Adaptive Periodic Transmission (APT) approach and the Mixed Transmission (MT) Approach, were proposed [12].

Most, if not all, of the location-dependent queries can be categorized as one of the three types described above. We can analyze each type separately in order to define the scenario clearly and simplify the problem. The rest of this thesis will focus on the first type only

2.2.3 DATA CACHING

Most mobile applications deal with two type of data: *Time-dependent data* and *Location-dependent data* [13,14,50,55,117]. Time-dependent data is a data whose value depends on time. For example, stock prices, reservations, match scores, etc. Whereas, location-dependent data is the data whose value is determined by the location to which it is related, for example hospitals, restaurant, gas station, etc. *Caching* is considered as one of the important techniques to relieve bandwidth constraint imposed on wireless mobile systems [5,21,22,40,41,53,112,119,121]. Copies of remote data can be kept in the local memory of mobile devices to substantially reduce data retrievals from the original server. This not only reduces the uplink and downlink bandwidth consumption but also the average data access latency. In a majority of mobile devices like laptops, palmtops and cellular phones, wireless communication is one of the major sources of energy consumption that reduces battery life [71]. Caching frequently accessed data in mobile devices can potentially minimize communication and hence conserve battery power. In client-server paradigm, when the client

receives a query, it first searches its cache. If there is a valid copy in the cache, it returns an answer immediately. If not, the client attempts to obtain the data item from the server. Advantages of data caching on mobile clients are:

- improved access latency,
- less wireless bandwidth requirements,
- low energy/power consumption due to lower data transmission, and
- Improved data availability in case of disconnection.

Client-side data caching has been considered a good solution for coping with the constraints of wireless/mobile environments. There exists considerable number of papers discussing general issues and research challenges related to caching strategies in wireless environment.

2.3 RELATED WORK ON CACHE MANAGEMENT

Client-side data caching is attractive in a mobile computing environment as it can overcome to some degree the constraints of wireless environment /devices such as scarce wireless bandwidth and limited power source. However, factors such as frequent client disconnections and movements across cells make the design of cache management a challenge. In the past few years, a lot of research effort has been done to develop efficient cache invalidation, replacement, and prefetching strategies. In the following, we briefly review related studies based on the taxonomy shown in Figure 2.3. In subsection 2.3.1, we will review the cache invalidation and replacement studies for temporal and location-dependent data. Then in subsection 2.3.2, review of work related to prefetching is presented.

2.3.1 CACHE INVALIDATION AND REPLACEMENT

Cache invalidation helps to ensure consistency between the cached data items at client end and the original data items stored at the server. It maintains the correctness of data in the client's cache. On the other hand, cache replacement evicts data items from cache in order to accommodate new data item in cache when the cache is full. In the following subsection, we review cache invalidation and replacement policies for time-dependent data as well as location-dependent data.

2.3.1.1 Time-Dependant Data

Since data may be updated at the server from time to time (referred to as time-dependent invalidation), the cached copy of the data at client may differ from the current copy of that data at

the server and therefore, cached data becomes invalid. Hence, the data cached at a client should be kept consistent with those at the server. Various cache invalidation schemes have been developed to ensure the data consistency between a client and the server [21,50,66,85,88,115]. Two basic categories of cache invalidation schemes, namely, *stateful server* approach and *stateless server* approach, have been discussed in the literature.

In stateful server approach, the server keeps the information about which data item is cached by which mobile client(s). Whenever a data item is updated, the server sends an invalidation message or a refresh message (with the new value) to those clients that cached this item. This approach requires the server to locate mobile clients. Therefore, the challenge for such an approach is how to handle disconnections and mobility. Moreover, the server stores the cache states for all the mobile clients, and hence, it is not scalable to a large client population.

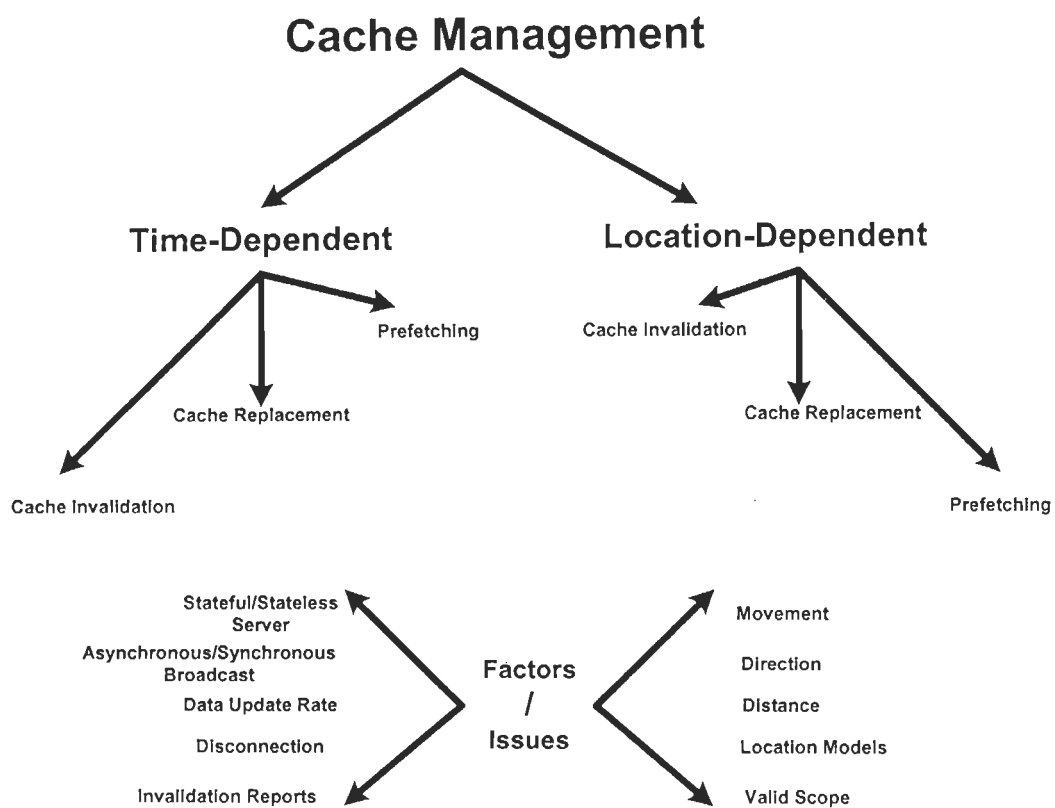


Figure 2.3 Taxonomy for Cache Management in Mobile Computing Environment

The stateless server approach does not require the server to be aware of the state of a client's cache. Instead, the server keeps track of the update history (of a reasonable length) and provides this information in the form of an *Invalidation Report* (IR) to the clients by periodic broadcasting or as requested by the clients. For instance, one could send a list of IDs for the items that have changed since the last IR was sent. The mobile clients, if active, listen to the broadcast IRs and update their caches accordingly. Most of the existing studies are based on the IR-based stateless approach. Stateless server approaches can be further categorized into *synchronous* and *asynchronous* approaches. In asynchronous approaches, invalidation reports are sent out on data modification. In synchronous approaches, the server sends out invalidation reports periodically.

Barbara and Imielinski [21,22], were first to introduce issues of cache invalidation using a broadcast medium in wireless mobile environment. They proposed three new cache invalidation methods suitable for wireless environment with high rate of client's disconnection. They are Timestamp (TS), Amnesic Terminals (AT) and Signatures (SIG). In all three strategies, the server (stateless) periodically broadcasts the report which reflects the changing database state. They categorized the mobile units into *sleepers* and *workaholics* on the basis of the amount of time they spend in their sleep mode. In the TS or AS algorithms, the entire cache will be invalidated if the disconnection time exceeds an algorithm-specified value (w seconds in TS and L seconds in AT), regardless of how many data items have actually been updated during the disconnected period. Their results showed that SIG which are based on the data compression technique for file compression are the best for long sleepers, when the period of disconnection is long and difficult to predict. As the rate of updates increases, TS becomes less and less efficient. AT method was the best for workaholics, that is, units which rarely go to sleep and are awake most of the time. They also extended the TS strategy to dynamically adjust window size to the changing query, update and wake-up ratios of the environment. The server has to use some feedback from the clients to modify the window size accordingly. The broadcast based solution is attractive because it can scale to any number of clients who listen to the broadcast report.

To salvage as many cache contents as possible, Wu et al. [67] presented an energy efficient cache invalidation method called Grouping with COld update-set REtention (GCORE). It modifies the TS or AT algorithms to include cache validity checks after reconnection. This scheme explores the fact that some of cached objects are still valid after reconnection. The server partitions the database into a number of groups. It maintains for each group the object update history of past W

broadcast intervals ($W \geq w$), where w is invalidation broadcast window. It also maintains the number of distinct objects that were most recently updated between $(T-W*L)$ and $(T-w*L)$ to speed up the group validity checking, where T is the current timestamp and L is the interval after which IR is broadcasted. The side effect of this method is that it requires uplink bandwidth and an update history window of past W broadcast interval to be specified, which introduces the same basic problem as in the TS method (e.g. when the disconnection time is greater than W , nothing can be salvaged). The effectiveness of the algorithm also depends heavily on the update pattern of the data items.

To make cache invalidation energy efficient, Tan et al. [45,65,66] proposed four schemes for cache invalidation. Two of them are - *Dual-Report Cache Invalidation* (DRCI) and *Bit-Sequences* (are variations of existing schemes) and other two schemes- *Selection Dual Report Cache Invalidation* (SDCI) and *Bit-Sequences with Bit Count* (BB), that support selective tuning to minimize energy consumption. Under DRCI, the server broadcasts every L time units a pair of invalidation reports, an Object Invalidation Report (OIR) and a Group Invalidation Report (GIR). The GIR report consists of the server's update history at a group level up to W ($W > w$) intervals, which aims at cutting down the probability of discarding the entire cache. Unlike DRCI, SDCI broadcasts GIR before OIR and the entries in OIR are ordered and broadcasts based on Groups. An additional pointer is added to each element of the GIR, which reflects the starting position of the objects within this group in the OIR. In BB, each bit sequence is associated with a bit count array. SDCI and BB consume significantly less energy than their counterparts, but clients has to wait for the invalidation report before their cache contents can be validated.

A major challenge for broadcast-based solution is to optimize the organization of broadcast reports. Jing et al. [8,47] addressed the report size optimization problem. They introduced a new cache invalidation algorithm called *Bit-Sequences* (BS), in which a periodically broadcast invalidation report is organized as a set of binary bit sequences with a set of associated timestamps. The BS algorithm with static (implicit) bit mapping was found to support clients regardless of the length of their disconnection times and offers the effectiveness of the report for data items covered in the report at the cost of about 2 bits/item. The effectiveness of a report is measured by the number of cached data items that can be accurately verified for a client by the use of the report. These bits can be used to cover the data items that are cacheable and most frequently referenced. The BS algorithm with dynamic (explicit) bit mapping was found to offer the same level of

effectiveness at the expense of only about half of the report size. The coarse granularity bit technique enables the static BS algorithm to cover more data items without increasing the size of the report. The hybrid BS scheme with the coarse granularity bit technique was also found to improve the effectiveness of the report by including recently updated data items in a dynamic BS scheme in the report. In general, changing workload parameters such as disconnection time, update rates, query rates, etc., has little impact on the performance of the BS algorithm. However, BS algorithm requires much greater invalidation report size than those in TS or AT methods, especially when the database size is larger.

In [87,88] , Hu and Lee, introduced two adaptive cache invalidation schemes namely *Adaptive Invalidation Report with Fixed Window* (AFW) and *Adaptive Invalidation Report with Adjusting Window* (AAW_TS and AAW_AT). Their second scheme is hybrid of the schemes proposed in [21,22,47]. In AFW, Invalidation Report (IR) consists of two types of information update history for window w , IR (w) and bit-sequences structure, IR (BS). Depending upon the feedback of timestamp of cache update of clients, the broadcasting is toggled between IR (w) and IR (BS). In AAW_TS, a dummy record ($dummyid, T_{lb}$) is used by the server to enlarge the window size in the next IR, where $dummyid$ is a special id not to be used by any data as id and T_{lb} is the latest timestamp. Client k checks to see whether its T_{lb}^k is within w or not. Otherwise, it checks to see whether the report include the dummy record and whether $T_{lb} < T_{lb}^k$. Otherwise, it sends back its T_{lb}^k to the server to request more update information. AAW_AT is variation of AAW_TS, instead of using w as the default window, the default window only cover one interval of update information. These adaptive methods make use of workload information from both client and server, so that the system workload has less impact on its performance while maintaining low uplink and downlink bandwidth requirement.

In IR based approaches, before answering a query from its local cache a client must listen to next IR to conclude if its cached copy is valid or not. Consequently, the average latency for answering a query is the sum of the actual query processing time and half of the IR Interval. Based on this observation, G. Cao in [33,34,35,36] proposed UIR-based approach. In this approach, a small fraction of the essential information (called Updated Invalidation Report (UIR)) related to cache invalidation is replicated several times within an IR interval and hence the client can answer a query without waiting until the next IR. He further proposed Counter-based cache invalidation algorithm. The counter-based scheme helps the server to find out hot data items and broadcast their updates to

the clients. Also, when the counter associated with a data item becomes 0, the server does not add it to the IR even though the data is updated during the last IR interval, hence, saving the broadcast bandwidth. To deal with counter accuracy problem, stateful server approach is used, in which the server maintains a Cached Item Register (CIR) for each client.

Chand et al. in [85] presented a synchronous stateful caching strategy called Update Report (UR), where all the recently updated/requested items are broadcasted immediately after the IR. The track of cached items for each client is maintained at home mobile support station in the form of Cache State Information (CSI). The use of CSI reduces the size of IR by filtering out non-cached items, handles long disconnection and supports client mobility. To further reduce query latency, the strategy uses Request Reports (RRs), where all the recently requested items are broadcasted after the UIR.

Kahol and Khurana [3,4] proposed cache invalidation scheme called AS (Asynchronous and Stateful). AS scheme, allows invalidation reports to be broadcasted only if any client has valid data in cache and as and when data changes (asynchronous). Queries are answered as they are generated and arbitrary sleep patterns are supported. This method requires Home Location Cache (HLC) for each mobile host at the server. This minimizes the overhead of validating MH's cache after each disconnection. Though maintaining state information (HLC) at the MSS can be considered as an overhead, but it can be benefited in profiling techniques and prefetching or hoarding data at the clients.

Wang et al. [118,120,121] proposed Scalable Asynchronous Cache Consistency Scheme (SACCS), which is hybrid of both stateful and stateless algorithms. Unlike stateful algorithms, SACCS maintains only one flag bit for each data item in MSS and unlike the existing synchronous stateless approaches, it does not require periodic broadcast of IRs. It has been shown through simulation that the proposed algorithm offers significantly better performance than TS and AS in both single cell and multi-cell environments.

Recently, Yeung and Kwok [79,83] have proposed cache invalidation strategies for error prone channel. Simulation results show that they perform better than the original IR strategy by alleviating the effect of transmission errors on the broadcast traffic and the impact of broadcast traffic on the downlink traffic in the system. The major drawback is large size of the invalidation report due to broadcast of same IR's in different cells, which in turn leads to higher query latency and client energy consumption.

The cache replacement issues for wireless data dissemination were first studied by Acharya et al. [96,97]. They described a new architecture called “*Broadcast Disks*”. In this approach, server continuously and repeatedly broadcasts data items to the clients. In effect, the broadcast channel becomes a “disk” from which clients can retrieve data as it goes by. The broadcast is created by assigning data items to different “disks” of varying sizes and speeds, and then multiplexing the disks on the broadcast channel. Items stored on faster disks are broadcast more often than items on slower disks. This approach creates a memory hierarchy in which data on the fast disks are closer” to the clients than data on slower disks. The number of disks, their sizes, and relative speeds can be adjusted, in order to more closely match the broadcast with the desired access probabilities at the clients. If the server has an indication of the client access patterns (e.g., by watching their previous activity or from a description of intended future use from each client), then hot pages (i.e., those that are more likely to be of interest to a larger part of the client community) can be brought closer while cold pages can be pushed further away. Acharya et al. proposed an optimal cache replacement policy known as *PLX* (P inverse X) for this new architecture, that replaces the cache-resident page having lowest ratio between its probability of access (P) and its frequency of broadcast (X). Simulation based study showed that this strategy could significantly improve the access latency over the traditional LRU and LFU policies.

Xu et al. [50,53,54] proposed a gain based cache replacement policy, namely *Stretch Access-rate Inverse Update-frequency* (SAIU), for on-demand broadcasts. Previous studies assumed that data items had the same size and ignored data updates and client disconnection. Different from previous work SAIU considered a real life application environment and was developed by taking into consideration various factors affecting cache management, such as varied data item sizes, retrieval delays, access probabilities and update frequencies. But, the optimal formula for determining the best cached items to be replaced based on above factors and also the influence of the cache consistency requirement was not considered in SAIU. The author in [50,54], further proposed an optimal cache replacement policy called Min-SAUD (*Minimum Stretch integrated with Access rates, Update frequencies and Cache validation Delay*), which accounts for the cost of ensuring cache-consistency before each cached items is used.

Recently, Yin et al. [74,75] proposed a generalized value function for cache replacement algorithm for wireless networks under strong consistency model. The distinctive feature of value function is that it is generalized and can be used for various performance metrics by making the

necessary changes. The authors proved that the proposed value function could optimize the access cost. However this strategy suffers from high computational complexity and does not consider association among data items.

2.3.1.2 Location-Dependant Data

Most of the previous work studied the cache consistency problem incurred by data updating (time-dependent update). In mobile computing environment, besides the temporal-dependent updates, cache inconsistency can also be caused by location changing (location-dependent updates). We broadly classify the location-dependent cache invalidation policies into two class- Symbolic Model based and Geometric Model based.

In [55,56], Xu et al. proposed three symbolic model based location-dependent cache invalidation schemes *Bit-Vector with Compression (BVC)*, *Grouped Bit-Vector with Compression (GBVC)* and *Implicit Scope Information (ISI)*. These schemes differ from each other in scope information organization.

In the *BVC* scheme, the *complete* validity information is attached to a data item value, i.e., the complete set of cells in which the data value is valid, is kept in the cache. It uses a bit vector (*BV*), corresponding to all the cells, to record valid scope. The length of a *BV* is equal to the number of cells in the system. A "1" in the n^{th} bit indicates that the data item value is valid in the n^{th} cell while "0" means it is invalid in the n^{th} cell. It is obvious that in *BVC* the overhead would be significant when the system is large.

In *GBVC* scheme, the whole geographical area is divided into disjoint districts and all the cells within a district form a group. A CID, denoted by (*group-ID*, *intra-group-ID*), consists of a group ID and a cell ID within the group. Validity information attached to a cached data value is represented as a vector of the form (*group-ID*, *BV*) and includes the current *group-ID* and a *BV* which corresponds to all the cells within the current group. Note that while delivering a data value to a client only the *BV* is attached since the *group-ID* can be inferred from the current CID.

For example, if there are 12 cells in the system, then a *BV* with 12 bits is constructed for each cached data item value. If the *BV* for a data item value is 000000111000, it means this value is valid in the 7th, 8th, and 9th cells only. Suppose that the whole geographical area is

further divided into two groups, such that cells 1-6 form group 0 and the rest form group 1. With the *GBVC* method, one bit is used to construct *group-ID* and a six-bit *BV* is used to record the cells in each group. For the data item value mentioned earlier, in group 0, the attached bit vector is (0,000000); in group 1, the attached bit vector is (1,111000). As can be seen, compared with the *BVC* method, the overhead for scope information is reduced in the *GBVC* method.

In *BVC*, a client stores in the cache the complete validity information of each cached data in the form of a bit vector. The disadvantage of this method is that the size of the validity information could be very large especially when the system consists of a large number of cells. Consequently, a large bandwidth and cache memory are needed. The advantage is that the validation process is very simple; only the current cell ID is needed. *GBVC* attempts to reduce the size of the validity information by only keeping partial information in the cache.

The *ISI* scheme attempts the other direction by trying to minimize the size of validity information at the expense of the validation procedure. Under this scheme, the server enumerates the scope distributions of all items and numbers them sequentially. The valid scopes within a scope distribution are also numbered sequentially. For any value of data item i , its valid scope is specified by a 2-tuple $(SDNi, SNi)$, where $SDNi$ is the scope distribution number and SNi denotes the scope number within this distribution. The 2-tuple is attached to a data item value as its valid scope. For example, suppose there are three different scope distributions (see Figure 2.4) and data item 4 has distribution 3. If item 4 is cached from cell 6 (i.e., CID=6), then $SDN4 = 3$ and $SN4 = 3$. That implies that the cached item 4's value is valid in cells 6 and 7 only.

| | | | | | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|---|----|----|----|
| CID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| (SDN)Scope | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| (SDN)Scope | 1 | | | 2 | | | 3 | | | 4 | | |
| (SDN)Scope | 1 | | 2 | | | 3 | | 4 | | | 5 | |

Figure 2.4 Data Items with Different Distributions

Though, these schemes used location dependent queries based on current location of a client, but they are also applicable to queries that are bound to other locations. Drawback of these schemes is that the client has to validate its cache every time it goes to new service area.

Zheng and Lee [12] presented a method for answering location-dependent queries in a mobile computing environment. They investigated a common scenario, where data objects (e.g. restaurant and gas station) are stationary while clients that issue queries are mobile (i.e. Mobile clients querying static objects). They presented an indexing and semantic cache method that records a cached items as well as its valid range for location-dependent queries based on the Voronoi Diagram (VD). They also proposed three cache replacement scheme based on *area*, *dist* (distance) and *ComA* (Common Area). The drawback is that a semantic cache and a VD index are very efficient when the number of service objects are limited. When the number becomes very large, the semantic cache can only contain a very limited area and the cache-hit rate is considerably reduced. Moreover, only a single cell environment was considered.

Zheng et al. in [14] studied the issues of cache invalidation and cache replacement for location-dependent data under a geometric location model. They introduced a new performance criterion, called *caching efficiency* and proposed basic schemes for location-dependent invalidation namely *Polygon Endpoints* (PE), *Approximate Circle* (AC) and a generic method *Caching-Efficiency-Based* (CEB). The *PE* scheme is a straightforward way to record the valid scope of a data value. It records all the endpoints of the polygon representing the valid scope. It contains complete knowledge of the valid scope of a data value. Its performance suffers when a polygon has a large number of endpoints. In AC scheme, a valid scope is approximated by the center of the inscribed circle and the radius value. Hence, the overhead is minimized. However, the inscribed circle is only a conservative approximation of a valid scope. When the shape of the polygon is thin and long, the imprecision introduced by the *AC* method is significant. This will lead to a lower cache hit ratio, since the cache will incorrectly treat valid data as invalid if the query location is outside the inscribed circle but within the polygon. The CEB scheme attempts to balance the storage overhead and the precision of invalidation information when selecting an approximation of a valid scope has to be decided.

Among the location-dependent cache replacement proposed in literature, Manhattan Distance-based cache replacement policy [98] was proposed to support location-dependent queries in urban environments. Cache replacement decisions are made on the basis of distance between a client's

current location and the location of each cached data object. Objects with the highest Manhattan distance from the client's current location are evicted at cache replacement. While the Manhattan based policy accounts for the distance between clients and data objects, the major limitation of this approach is that it ignores the temporal access locality of mobile clients and the direction of client movement while making cache replacement decisions.

Furthest Away Replacement (FAR) [90] policy uses the current location and movement direction of mobile clients to make cache replacement decisions. Cached objects are grouped into two sets, viz., in-direction set and the out-direction set. Data objects in the out-direction set are always evicted first before those in the in-direction set. Objects in each set are evicted in the order based on their distance from the client. Similar to the Manhattan approach, FAR also neglects the temporal properties of clients' access pattern. It also becomes ineffective when mobile clients change direction frequently due to frequent change in the membership of objects between the in-direction and out-direction sets.

In addition, two cache replacement policies *Probability Area (PA)* and *Probability Area Inverse Distance (PAID)* were proposed by Zheng et al. [14], that consider the valid scope area (for both methods) and data distance (for PAID only) and combine these with access probability. The cost function of PAID is given by $P_i A(vs_i) / D(vs_i)$, where, P_i is the probability of access of data item i , vs_i is the valid scope of the instance of data item i , $A(vs_i)$ is the area of the valid scope vs_i and $D(vs_i)$ is the distance of the vs_i from the current position of client. It neither takes into account the size of the data object nor does it give priority to the data objects in cache that are near to the mobile client. *Mobility-Aware Replacement Scheme (MARS)* [68] policy is also a cost based policy, which comprises of temporal score, spatial score and cost of retrieving an object. Unlike PAID, it takes into account the updates of data objects. But as far as location-dependent data (LDD) is concerned, their update rate (if exist) is negligible as compared to temporal data. Thus, for LDD, only spatial score dominates which consists of area of valid scope, data distance from current client location and data distance from future client location. The impact of client's anticipated location or region in deciding cache replacement still remains unexplored.

None of the existing these cache replacement policies are suitable if client changes its direction of movement quite often. They only consider the data distance (directional/undirectional) but not the distance based on the predicted region or area where the client can be in near future. Very few of these policies [14,68] account for the location and movement of mobile clients.

2.3.2 PREFETCHING

Prefetching is a technique that can reduce access latency and improve cache hit ratio. In prefetching, access to remote data is anticipated and the data is fetched *a priori*. Most of the existing prefetching schemes [36,40,41,63,94] use uplink bandwidth and battery energy to improve cache hit ratio and reduce access latency. Broadcast and multicast [49,95] are proven as effective data dissemination techniques in mobile computing environments. A prefetching scheme that considers both power and bandwidth efficiency needs to be investigated for data dissemination environments.

Prefetching and caching are also effective techniques for improving the performance of file systems, but they have not been studied in an integrated fashion. The goal of a prefetching and caching policy is to make the decisions like when to fetch a block from disk, which block to fetch and which block to replace when the fetch is initiated so that the total elapsed time is minimized. Cao et al. in [86], proposed four properties that optimal integrated prefetching and caching strategy must satisfy, and then presented and studied two such integrated strategies, called *aggressive* and *conservative*. They proved that the performance of the conservative approach is within a factor of two of optimal prefetching schedule and that the performance of the aggressive strategy is a factor significantly less than twice that of the optimal prefetching case.

Prefetching has also been investigated to reduce web access latency in wireline networks [1,6,24,32,38,93,103]. Existing works [1,24,32] investigate prefetch schemes involving point-to-point session transmission model, which is different from the broadcast communication model in wireless mobile networks. A prefetching technique for *Broadcast Disks* [97] known as *PT* was proposed by Acharya et al. [96]. This technique uses a heuristic that computes a value for each data page by multiplying the probability of access for that page by the time that will elapse before that page appears next on the broadcast disk, this value is called as data page's *pt* value. *PT* finds the page in the cache with the lowest *pt* value, and replaces it with the currently broadcast page if the latter has a higher *pt* value. The *pt* value of a data page is dynamic because the time parameter of the metric is constantly changing.

In recent years, hybrid of prefetching and data invalidation has been studied for performance tradeoff [36,73] in wireless environment. G. Cao in [36] calculated the prefetch access ratio (*PAR*) for mobile devices, which is the number of prefetches divided by the number of accesses for each data item. Mobile devices use a threshold *PAR* to determine whether to prefetch a data item or not.

Yin et al. in [73], used an adaptive value function instead of *PAR* to evaluate each data item. All of these schemes use prefetching to retrieve the desired data in a proactive fashion, in which mobile devices need to send an uplink message to the base station and then wait for a period of time to acquire the data from downlink channel. Since mobile devices consume much more energy for sending an uplink message than receiving a downlink message of the same size, due to channel contention and data retransmission [114], proactive prefetching schemes are not expected to be energy efficient in general.

Shen et al. [41] proposed a novel energy and bandwidth efficient data caching mechanism, called *Greedy Dual Least Utility* (GD-LU) that enhances dynamic data availability while maintaining consistency. The proposed utility-based caching mechanism considers several characteristics of mobile distributed systems, such as connection-disconnection, mobility handoff, data update and user request patterns to achieve significant energy savings in mobile devices. They developed an analytical model for energy consumption of mobile devices in a dynamic data environment. Based on the *utility function* derived from the analytical model, cache replacement and passive prefetching of data objects was done.

Dissemination of data by broadcasting may induce high access latency if number of broadcasted data items are large. Saygien et al. [116] proposed two methods aiming to reduce client access latency of broadcast data. These methods are based on analyzing the broadcast history (i.e., the chronological sequence of items that have been requested by clients) using data mining techniques. Data mining research [64,69,102,111] deals with finding relationships among data items and grouping the related items together. The two basic relationships that are of particular concern were:

- Association, where the only knowledge we have is that the data items are frequently occurring together, and when one occurs, it is highly probable that the other will also occur.
- Sequence, where the data items are associated, and in addition to that, we know the order of occurrence as well.

Their main interests were in finding the sequences among the data items that occur frequently. With the first method, the data items in the broadcast disk are organized in such a way that the items requested subsequently are placed close to each other. The second method, focuses on improving the cache hit ratio by enabling clients to prefetch the data from the broadcast disk based on the rules extracted from previous data request patterns. Authors used Web logs to estimate the effectiveness

of both strategies and it was shown through performance experiments that the proposed rule-based methods are effective in improving the system performance in terms of the average latency as well as the cache hit ratio.

Yin and Cao [73] proposed a power-aware prefetch scheme, called *Adaptive Value-based Prefetch (AVP)* scheme. The *AVP* scheme defines a value function which can optimize the prefetch cost to achieve better performance. Also, *AVP* dynamically adjusts the number of prefetches to get better tradeoff between performance and power. To address the issues of power constraints of the mobile clients and other factors such as the data size, the data access rate, and the data update rate, they first propose a value-based (*VP*) scheme, which makes prefetch decisions based on the value of each data item considering various factors such as access rate, update rate, and data size. Then, they extended the *VP* scheme and presented adaptive value-based prefetch (*AVP*) scheme, which achieved a balance between performance and power based on different user requirements. Extensive simulations were used to justify the analysis. Their proposed schemes reduced the energy consumption and improved the system performance in terms of *stretch* [54] under various scenarios.

Song and Cao [42] realized that cache misses are not isolated events, and a cache miss is often followed by a series of cache misses. They addressed the prefetching issues among related data items by using a *cache-miss-initiated prefetch (CMIP)* scheme, which is based on association rule mining technique.

Drakatos et al. in [99] proposed a prefetching strategy that prefetches data items with maximum benefits and evicts cache data with minimum benefit. The data item benefit is evaluated based on the user's query context defined as a set of constraints of both movement pattern and information context requested by the mobile user. Similarly, Chen et. al [18] also proposed a *Benefit-Oriented Prefetching (BOP)* that efficiently selects the LDD of interest to a client and prefetches them for the client.

2.4 SUMMARY

Mobile computing has proven a fertile area of work for researchers in the areas of database and data management. The inherent limitations of mobile computing systems present a challenge to the traditional techniques used in database management. As we can see, the amount of research in this area in the last few years has been staggering and there are many problems that remain open for research. There is a need for better protocols in the area of data sharing and transaction

management, better interfaces, clever algorithms that exploit locality to shape the answers to queries. The above literature review shows that most of the work available today has addressed cache invalidation and replacement strategy for temporal data, simple queries, fixed sized data, assumed error free wireless environment, etc. Some amount of work has been done for LDIS, but it is an area that still has many unexplored issues - caching being one of them that plays an important role in improving the performance of any location-dependent service.

CHAPTER 3

LOCATION-DEPENDENT CACHE INVALIDATION

3.1 INTRODUCTION

Mobile computing, as compared to traditional computing paradigms, allows mobile users to access information anywhere, any time, and in any form. Data caching at mobile clients reduces data access time and increases its availability, thus, improving system performance. In location-dependent information services, data values for a data item depend on geographical locations. Traditional caching strategies did not consider this and therefore, are inefficient for location-dependent data. As mentioned earlier in this thesis, caching issues for location-dependent data become challenging because of spatial property of LDD and mobility of users. Spatial data cached in the mobile user's device may become invalid when the user moves to a different location. Therefore, issues in cache invalidation and replacement need to be re-examined for location-dependent data. In this chapter, we focus on location-dependent cache invalidation for geometric location model.

Since the server generally does not know which items are cached by the clients, a common method to perform location-dependent cache invalidation is to attach a valid scope with each data value returned to the client. The client caches the data as well as its valid scope that can be used for checking its validity without connecting to the data server on fixed network. There are two situations where checking the validity of data is necessary at the client end: 1) the same query may be issued later when the client has moved to a new location; 2) a mobile client may keep on moving after it submits a query, and it may have moved to a new location when the response comes back (if there is a long data access delay) [14,50]. In both cases, if client's location is not within the valid scope attached with data, the data is marked as invalid and a new query is submitted. In this thesis, we assume that when a data value is delivered from the server to a client, its valid scope is also attached with it so that the client can check the data validity against its location. However, different methods might be employed at the server to represent a valid scope to be sent and stored in the client cache along with its data value.

An important aspect of cache invalidation in LDISs is how to represent the valid scopes. Downloading and storing valid scope along with data consumes more bandwidth and needs more

storage space. The actual valid scope is represented by convex polygon under geometric location model. Cost of storing all endpoints of a polygon is high. Therefore, an approximation of the actual polygon is needed to reduce the cost. Thus, the issue is how to represent the valid scope in order to balance the precision and overhead costs. The concept of valid scope information was first proposed by Zheng et al. [14], in which it was used to construct a semantic cache to reuse the cached data. Authors in [14] try to find a representation of the valid scope which does not introduce too much overhead. In this chapter, we show that the algorithm given in [14] does not always give best possible valid scope. We present a modified procedure for finding best suitable candidate for valid scope that increases caching efficiency. We compare our Generalized Caching Efficiency Based (CEB_G) algorithm which selects the best suitable candidate for valid scope with the Caching-Efficiency-Based (CEB) algorithm proposed in [14]. We further introduce a new metric Future Access (FA) and based on it propose Caching Efficiency with Future Access Based (CEFAB) algorithm which selects the best suitable candidate for valid scope using FA. We further generalize CEFAB algorithm into CEFAB_G.

3.2 SYSTEM MODEL

This section describes the system model adopted in this thesis. As described in the previous chapter (see section 2.2.1), the information system provides location-dependent information services to mobile clients. We refer to the geographical area covered by the system as the service area. Unlike the common data, every item of LDD usually has various values, which are termed as *data instances* of an LDD item. Hence, data item may show different values when it is queried by clients at different locations.

We assume a geometric location model (see section 2.2.2.1), i.e., a location is specified as two-dimensional coordinate. However, it can be easily extended to 3-dimension space by including the third dimension. Mobile clients can determine their locations using system such as the Global Positioning System (GPS) [43]. In two-dimensional space, a valid scope v can be represented by a geometric polygon $p(e_1, e_2, \dots, e_n)$, where e_i 's are endpoints of the polygon.

A mobile client can cache data on its local disk or in any storage system that survives power-off. In this chapter, data values are assumed to be of fixed size and read-only so that we can omit the influence of data sizes and updates on cache performance and concentrate on the impact

caused by the unique properties of location-dependent data. In the abstract model, the path of a moving client is represented by a curve in 2-dimension (x-y plane), as shown in Figure 3.1(a).

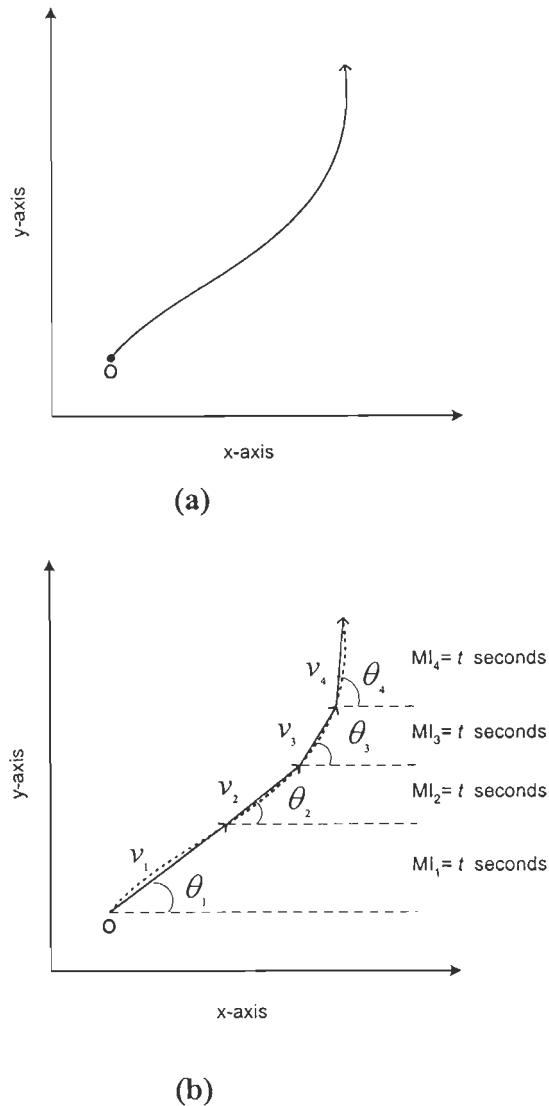


Figure 3.1 Client's Movement Path (a) Abstract Model (b) Discrete Model

Though abstract model is simple, from computer implementation point of view, discrete model is preferred [77]. In the discrete model the path traveled is modeled as a sequence of line segments, each associated with fixed velocity and direction, as shown in Figure 3.1(b). Length of the line segment depends on the rate of change of direction and velocity. For random movement this duration between change in direction and velocity is small and for regular movement and highway users this duration is large. This duration is known as *Moving Interval* (MI) [14,50,68, 107]. Figure 3.1(b) shows the discrete movement of a mobile user with MI of t seconds. The

distance between any two locations or points is the length of a straight line connecting the two points (i.e. Euclidean distance).

3.3 GENERALIZED CACHING EFFICIENCY BASED (CEB_G) ALGORITHM

Zheng et al. proposed three cache invalidation schemes based on geometric model [14]: Polygonal Endpoints (PE) Scheme, Approximate Circle (AC) Scheme and Caching-Efficiency-Based (CEB) Method. The PE scheme, records all endpoints of the polygon representing the valid scope. However, when the number of endpoints is large the overall performance deteriorates, because PE scheme will consume a large portion of the wireless bandwidth and the client's limited cache space for storage, effectively reducing the amount of space available for caching the data itself. The advantage is that it contains complete knowledge of the valid scope of data value.

An alternative to PE scheme, is the AC scheme, where an inscribed circle is use to approximate the polygon instead of recording the whole polygon. In other words, a valid scope can be approximated by the center of the inscribed circle and its radius value. The medial axis approach is used for generating inscribed circle [48,50,76] in a polygon. When the shape of the polygon is thin and long, the imprecision introduced by the AC method is significant. This leads to a lower cache hit ratio, since the cache incorrectly treats valid data as invalid if the query location is outside the inscribed circle but within polygon.

CEB is a generic method for balancing the overhead and the precision of valid scopes. It is based on caching efficiency. Suppose that, the valid scope of a data value is v , and v'_i is a sub region contained in v . Let D be the data size, $A(v'_i)$ the area of v'_i , and $O(v'_i)$ the overhead needed to record the scope v'_i . Then, *caching efficiency* of the data value with respect to a scope v'_i is defined as follows [14,50]:

$$E(v'_i) = \frac{A(v'_i)/A(v)}{(D + O(v'_i))/D} = \frac{A(v'_i)D}{A(v)(D + O(v'_i))} \quad (3.1)$$

CEB scheme can be stated as follows:

For a data item value with valid scope of v , given a candidate valid scope set $V' = \{v'_1, v'_2, \dots, v'_k\}, v'_i \subseteq v, 1 \leq i \leq k$, choose the scope v'_i that maximizes caching efficiency $E(v'_i)$ as the valid scope to be attached to the data.

Thus, CEB scheme generates candidate valid scopes and then selects the best one. Greedy approach is used to generate a series of candidate polygons. Suppose the current candidate polygon is v'_i . CEB scheme considers all polygons resulting from deletion of one endpoint from v'_i and chooses the next candidate, v'_{i+1} , the polygon which has the maximal area. The algorithm can be seen in [14,50] which describe the generation of candidate valid scopes and the selection of the best valid scopes.

To look in detail how the greedy approach works in CEB, consider an example of polygon consisting of 7 sides. CEB sets the original polygon as candidate polygon (size=7). The first iteration finds all polygons with 6 vertices from the original polygon having 7 vertices and finds the best among the six sided polygons. It then sets it as the candidate polygon (size=6) for the next iteration. The second iteration finds all polygons with 5 vertices from 6 vertices of the candidate polygon (size=6), finds the best among them and sets it as the candidate polygon (size=5) for next iteration. This process is repeated with successive lower order polygons until the number of sides of the sub polygon becomes 3.

The complexity of CEB algorithm is $O(n^2)$. However, if the polygons are not regular CEB does not ensure that the final polygon selected is always optimal. The optimal polygon may be among those polygons which CEB never considers. In each iteration, CEB always selects the best out of the polygons constructed during that iteration. It then explores only the sub cases of this best. It is shown in section 3.3.1 that in many cases CEB may not give the optimum solution. This affects the overall performance of the system resulting in less cache hit.

We propose a generalized method CEB_G for the generation of candidate valid scope set. This method explores all possible combinations of sub polygons in the original polygon. The Algorithm A1, in Figure 3.2 describes proposed CEB_G method for the generation of candidate valid scopes and for the selection of the best valid scope.

Now consider the same example as above, i.e., a polygon consisting of 7 sides. In Algorithm A1 (see Figure 3. 2), the first iteration, finds all 7C_6 combinations of 6 sided sub polygon from the original polygon (size=7) and finds the best among them. The second iteration finds all 7C_5 combinations of 5 sided sub polygons from the original polygon (size=7) and not the polygon

selected as best in first iteration. It then finds the best among them. This goes on until the side of the sub polygon becomes 3. This method has all possible valid scopes available for selecting the best in each iteration as compared to CEB.

Algorithm A1: Selection of the Best Valid Scope for the CEB_G Method

Input: valid scope $v = p(e_1, \dots, e_n)$ of a data value;

Output: the attached valid scope v' ;

Procedure:

```

1:  $v'_1 :=$  the inscribed circle of  $p(e_1, \dots, e_n)$ 
2:  $v' := v'_1; E_{max} := E(v'_1);$ 
3:  $v'_2 := p(e_1, \dots, e_n);$ 
4:  $i := 2;$ 
5: while  $n - i \geq 1$  do
6: //containing at least three end-points for a polygon
7: if  $E(v'_i) > E_{max}$  then
8:  $v' := v'_i; E_{max} := E(v'_i);$ 
9: end if
10: if  $n - i > 1$  then
11:  $v'_{i+1} :=$  the polygon having maximum area, consisting of  $((n - 1) - i + 2)$  endpoints of
     $v$  and being bounded by  $v$ ;
12: end if
13:  $i := i + 1;$ 
14: end while
15: output  $v'$ .

```

Figure 3.2 Algorithm for CEB_G

Although, the complexity of CEB_G is exponential (${}^n C_3 + {}^n C_4 + \dots + {}^n C_{n-1} \approx 2^n$) but exponential factor matters when the number of sides of polygon is high. Following observations make CEB_G attractive in comparison to CEB.

- The algorithm is to be used at the server end which is (assumed to be) a powerful machine with high resources.

- In actual scenario, maximum number of polygon sides may vary from 6 to 10. Beyond 10 sides, the polygon resembles more towards circle as circle is a polygon having infinite sides. So for polygons with number of sides ≤ 10 even exponential complexity may be acceptable.
- Calculation of the best valid scope needs to be done only once and can be stored on the server along with the actual valid scope.
- CEB_G selects more precise representation of valid scope as compared to CEB which improves the over all performance, resulting in higher cache hit than that of CEB.

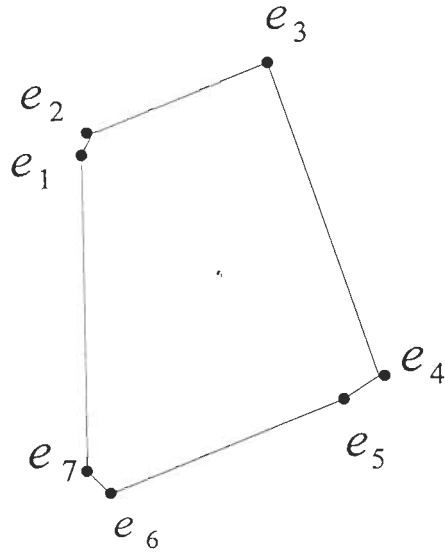
3.3.1 Case Study

We consider the polygon given in Figure 3.3(a) with endpoints $e_1(1351.5,3513.22)$, $e_2(1352.89,3516.69)$, $e_3(1480.88,3580.69)$, $e_4(1535.16,3307.59)$, $e_5(1522.8,3279.94)$, $e_6(1354.61,3183.61)$ and $e_7(1351.5,3187.75)$. CEB selects the polygon $p_{CEB}(e_1, e_3, e_4, e_7)$ given in Figure 3.3(b) as the best candidate for the valid scope to be sent to client along with data, whereas CEB_G selects the polygon $p_{CEB_G}(e_1, e_3, e_4, e_6)$ given in Figure 3.3(c) as the best candidate.

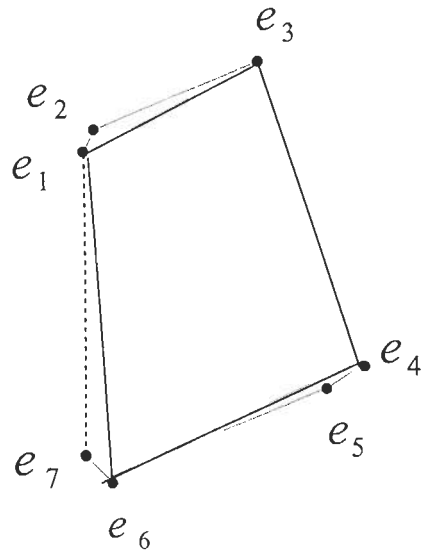
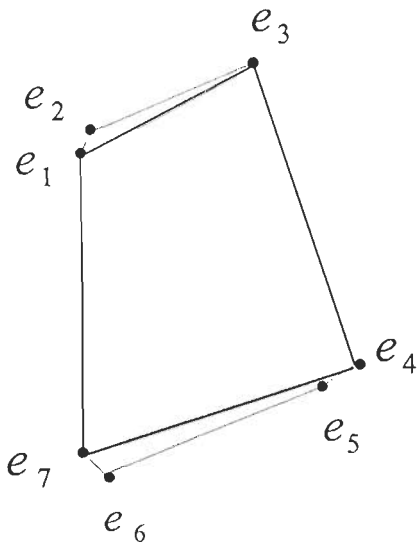
Stepwise execution is shown in Table 3.1. The entries in each row shows the best polygon selected by CEB and CEB_G in each iteration. The final best polygons selected by CEB and CEB_G to be sent to client along with data are $p(e_1, e_3, e_4, e_7)$ and $p(e_1, e_3, e_4, e_6)$ respectively. Both polygons have size 4, which means that the over head of sending and storing the polygon with data is same. But the area of p_{CEB_G} is greater than the area of p_{CEB} , which means p_{CEB_G} has more precise representation than p_{CEB} . This results in more cache hit at the client side.

3.3.2 Precision versus Computational Complexity

The greedy approach of CEB, selects only one best candidate valid scope in each iteration. It has low computational complexity but at the same time low precision also. CEB_G considers all possible sub polygons of the original polygon. This increases precision but also computational complexity. However, if the server has limited computing power, lesser number of sub polygons may be considered in each iteration. Considering best two candidate valid scopes instead of one in each iteration, we can get more precise representation of valid scope in many cases, thus improving the caching efficiency than CEB.



(a) Original Polygon



(b) Best candidate for CEB, p_{CEB}

(c) Best candidate for CEB_G, p_{CEB_G}

Figure 3.3 Case Study

Table 3.1 Stepwise Execution for CEB and CEB_G

| Iteration | CEB | CEB_G |
|-----------|--|--|
| 0 | $p(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ | $p(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ |
| 1 | $p(e_1, e_3, e_4, e_5, e_6, e_7)$ | $p(e_1, e_3, e_4, e_5, e_6, e_7)$ |
| 2 | $p(e_1, e_3, e_4, e_5, e_7)$ | $p(e_1, e_3, e_4, e_5, e_7)$ |
| 3 | $p(e_1, e_3, e_4, e_7)$ | $p(e_1, e_3, e_4, e_6)$ |
| 4 | $p(e_1, e_4, e_7)$ | $p(e_1, e_4, e_6)$ |

Table 3.2 Stepwise Execution with best two in CEB

| Iteration | Best Two Candidate Valid Scores |
|-----------|--|
| 0 | $p(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ |
| 1 | $p(e_1, e_3, e_4, e_5, e_6, e_7), p(e_2, e_3, e_4, e_5, e_6, e_7)$ |
| 2 | $p(e_1, e_3, e_4, e_5, e_7), p(e_1, e_3, e_4, e_5, e_6)$ |
| 3 | $p(e_1, e_3, e_4, e_7), p(e_1, e_3, e_4, e_6)$ |
| 4 | $p(e_1, e_4, e_7), p(e_1, e_4, e_6)$ |

Computing time only increases by a constant and the complexity remains same, i.e., $O(n^2)$. Consider again the original polygon in Figure 3.3(a). If we select best two candidate valid scopes in each iteration, polygon $p(e_1, e_3, e_4, e_6)$ is selected as the best valid scope, same as the polygon selected by CEB_G method. Stepwise execution is shown in Table 3.2.

3.4 CACHING EFFICIENCY WITH FUTURE ACCESS BASED (CEFAB) ALGORITHM

Predicting accurately the movement behavior of the client is a challenging task. Lots of research is going on in this area [107]. Using the discrete model (described in section 3.2), we can track the future movement of the client for a certain amount of time. Since the client changes speed and direction randomly, so it is very difficult to track its entire path. However we can make use of the Moving Interval (MI) to track client's future path up to the end of MI from the current query location. Moving Interval is duration within which client's velocity and direction remains constant.

Basic Idea:

Given two sub regions v_i and v_j of the valid scope v of a data item. Choose the sub region v_i (v_j) if the mobile client will remain in v_i (v_j) for a longer duration than v_j (v_i) even if $A(v_j) > A(v_i)$ ($A(v_i) > A(v_j)$).

Approach:

Suppose S_{MI} be the start and E_{MI} be the end of Moving Interval as shown in Figure 3.4. Let T_Q be the time at which a query is executed by the client, where $S_{MI} \leq T_Q \leq E_{MI}$. Also, let e_{T_Q} and $e_{E_{MI}}$ be the points in x-y plane at time T_Q and E_{MI} respectively. We define Future Movement Path (FMP) for interval $[T_Q, E_{MI}]$ as:

$$FMP_{T_Q, E_{MI}} = Line_Segment(e_{T_Q}, e_{E_{MI}}) \quad (3.2)$$

Server selects the sub-valid scope that contains maximum part of FMP and sends it to the client along with data value. Consider the scenario as shown in Figure 3.5.

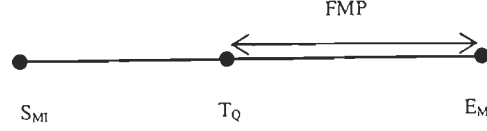


Figure 3.4 Future Movement Path

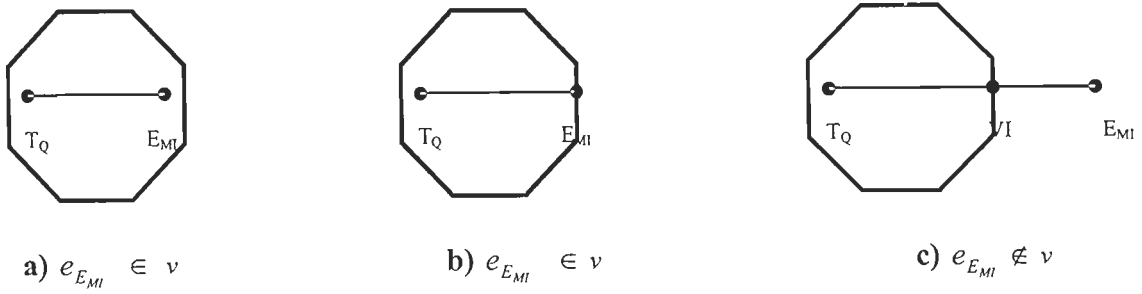


Figure 3.5 $e_{E_{MI}}$ with respect to Valid scope v

In the first and the second case the FMP is within the valid scope, but in the third case, we have to consider the intersection point of $Line_Segment(e_{T_Q}, e_{E_{MI}})$ with the valid scope v , let it be e_{VI} , because we select the best candidate valid scope v , so we consider line segment that is within the valid scope. Redefining FMP with respect to the valid scope v for interval $[T_Q, E_{MI}]$, we have

$$FMP_{T_Q, E_{MI}}(v) = \begin{cases} Line_Segment(e_{T_Q}, e_{E_{MI}}) & \text{if } e_{E_{MI}} \in v \\ Line_Segment(e_{T_Q}, e_{VI}) & \text{if } e_{E_{MI}} \notin v \end{cases} \quad (3.3)$$

Our ultimate goal is to select a valid scope that increases the cache hit of the client, which means the sub polygon which retains the total FMP. Keeping this fact, we define a new metric called *Future Access* (FA) for valid scope v_i' for interval $[T_Q, E_{MI}]$, given by

$$FA_{T_Q, E_{MI}}(v_i') = \frac{Length(FMP_{T_Q, E_{MI}}(v_i'))}{Length(FMP_{T_Q, E_{MI}}(v))} \quad (3.4)$$

where, v_i' = sub region contained in v

v =valid scope of a data value

Length= computes length of line segment between two given end points.

FA helps to find out the best candidate polygon/sub polygon with respect to its future validity in client's cache, because it takes into account the future path to be traversed by the client from the current position.

After integrating FA with caching efficiency we get an integrated metric, called *Caching Efficiency with Future Access* (CEFA) for valid scope v_i' in interval $[T_Q, E_{MI}]$, given by:

$$CEFA_{T_Q, E_{MI}}(v_i') = E(v_i') * FA_{T_Q, E_{MI}}(v_i') = \frac{A(v_i')D}{A(v)(D + O(v_i'))} * FA_{T_Q, E_{MI}}(v_i') \quad (3.5)$$

The new metric takes into account the future movement behavior of client. We propose a new cache invalidation algorithm called *Caching Efficiency with Future Access Based* (CEFAB) that uses CEFA metric. CEFAB scheme can be stated as follows:

For a data item value with valid scope of v , given a candidate valid scope set $V' = \{v'_1, v'_2, \dots, v'_k\}, v'_i \subseteq v, 1 \leq i \leq k$, choose the scope v'_i , that maximizes $CEFA_{T_Q, E_{MI}}(v'_i)$, as the valid scope to be attached to the data.

Thus, CEFAB also generates candidate valid scopes and then selects the best one. If $T_Q = E_{MI}$, then it returns the original polygon as the best valid scope because client's movement behavior cannot be predicted for next Moving Interval. For other cases, suppose the current candidate polygon is v'_i , CEFAB considers all polygons resulting from the deletion of one endpoint of v'_i and chooses the next candidate, v'_{i+1} , the polygon which has maximum $Length(FMP_{T_Q, E_{MI}}(v'_i))$. In case of tie for length, the polygon with maximal area is selected. The algorithm of CEFAB is described in Figure 3.6. Similar to CEB_G, CEFAB can also be generalized into CEFAB_G, described by Figure 3.7.

3.5 SIMULATION MODEL

This section describes the simulation model used to evaluate the performance of the proposed location-dependent cache invalidation methods. Our Simulator is implemented in C++ and setup is similar and in accordance with those used in earlier studies [14,44,50].

3.5.1 System

Since seamless hand-off from one cell to another is assumed, the network can be considered a single, large service area within which the clients can move freely and obtain location-dependent

Algorithm A2: Selection of the Best Valid Scope for the CEFAB Method

Input: valid scope $v = p(e_1, \dots, e_n)$ of a data value, T_Q and E_{MI} ;

Output: the attached valid scope v' ;

Procedure:

```
1:  if  $T_Q == E_{MI}$  then
2:     $v' := v$ ;
3:    go to 19;
4:  end if
5:     $v'_1 :=$  the inscribed circle of  $p(e_1, \dots, e_n)$ 
6:     $v' := v'_1$ ;  $CEFA_{max} := E(v'_1)$ ;
7:     $v'_2 := p(e_1, \dots, e_n)$ ;
8:     $i := 2$ ;
9:    while  $n - i \geq 1$  do
10:   //containing at least three end-points for a polygon
11:   if  $CEFA_{T_Q, E_{MI}}(v'_i) > CEFA_{max}$  then
12:     $v' := v'_i$ ;  $CEFA_{max} := CEFA_{T_Q, E_{MI}}(v'_i)$ ;
13:   end if
14:   if  $n - i > 1$  then
15:     $v'_{i+1} :=$  the polygon that is deleted one endpoint from  $v'_i$  while being bounded by  $v$  and
           having maximum  $Length(FMP_{T_Q, E_{MI}}(v'_i))$ ;
16:   end if
17:    $i := i + 1$ ;
18:   end while
19:  output  $v'$ .
```

Figure 3.6 Algorithm for CEFAB

Algorithm A3: Selection of the Best Valid Scope for the CEFAB_G Method

Input: valid scope $v = p(e_1, \dots, e_n)$ of a data value, T_Q and E_{MI} ;

Output: the attached valid scope v' ;

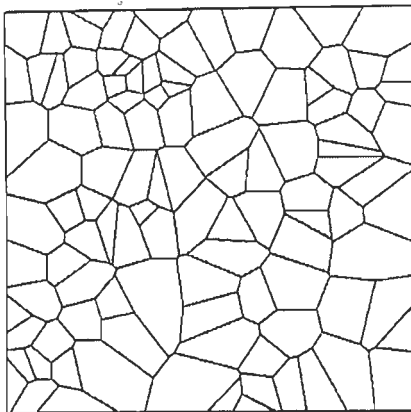
Procedure:

```
1:  if  $T_Q == E_{MI}$  then
2:     $v' = v$ ;
3:  go to 19;
4:  end if
5:   $v'_1 :=$  the inscribed circle of  $p(e_1, \dots, e_n)$ 
6:   $v' := v'_1$ ;  $CEFA_{max} := E(v'_1)$ ;
7:   $v'_2 := p(e_1, \dots, e_n)$ ;
8:   $i := 2$ ;
9:  while  $n - i \geq 1$  do
10: //containing at least three end-points for a polygon
11:  if  $CEFA_{T_Q, E_{MI}}(v'_i) > CEFA_{max}$  then
12:     $v' := v'_i$ ;  $CEFA_{max} := CEFA_{T_Q, E_{MI}}(v'_i)$ ;
13:  end if
14:  if  $n - i > 1$  then
15:     $v'_{i+1} :=$  the polygon having maximum  $Length(FMP_{T_Q, E_{MI}}(v'_i))$ , consisting of
         $((n - 1) - i + 2)$  endpoints of  $v$  and being bounded by  $v$ ;
16:  end if
17:   $i := i + 1$ ;
18:  end while
19:  output  $v'$ .
```

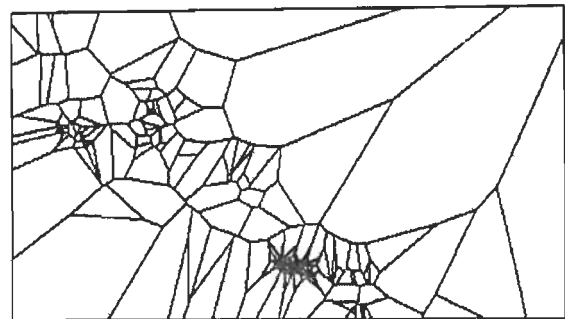
Figure 3.7 Algorithm for CEFAB_G

information services. In our simulation, the service area is represented by a rectangle with a fixed size of $Size$. We assume a "wrapped-around" [14,44,107] model for the service area. In other words, when a client leaves one border of the service area, it enters the service area from the opposite border at the same velocity.

The database contains $ItemNum$ items. Every item may display $ScopeNum$ different values for different client locations within the service area. Each data value has a size of $DataSize$. In the simulation, the scope distributions of the data items are generated based on voronoi diagrams (VDs) [48,76] because valid scopes of nearest neighbor queries is defined by VD. Formally, given sets of point $O=\{o_1, o_2, \dots, o_n\}$, $V(o_i)$, the Voronoi Cell (VC) for o_i , is defined as the set of points q in the space such that $dist(q, o_i) < dist(q, o_j), \forall j \neq i$. That is, $V(o_i)$ consists of set of points for which o_i is nearest neighbor. In our simulation, first data set Scope Distribution 1 (Figure 3.8 (a)) contains 110 points randomly distributed in a square Euclidean space. The second data set, Scope Distribution 2 (Figure 3.8 (b)), contains the locations of 215 hospitals in California area, which is extracted from the point data set available at [106].



(a) Scope Distribution 1 ($ScopeNum=110$)



(b) Scope Distribution 2 ($ScopeNum=215$)

Figure 3.8 Scope Distributions for Performance Evaluation

This model assumes that two floating-point numbers are used to represent a two-dimensional coordinate and one floating-point number to represent the radius of circle. The size of a floating-point number is $FloatSize$. The wireless network is modeled by an uplink channel and a downlink channel. The uplink channel is used by clients to submit queries, and the downlink channel is used by the server to return query responses to target clients. The communication between the server and a client makes use of a point-to-point connection. It is assumed that the

available bandwidth is *UplinkBand* for the uplink channel and *DownlinkBand* for the downlink channel.

3.5.2 Client

The mobile client is modeled with two independent processes: *query process* and *move process*. The *query process* continuously generates location-dependent read-only queries for different data items. After the current query is completed, the client waits for an exponentially distributed time period with a mean of *QueryInterval* before the next query is issued. The client access pattern over different items follows a *Zipf* distribution with skewness parameter θ , which is shown to be a realistic approximation of skewed data access and are frequently used to model non-uniform distribution [14,50,70,74,75]. In the Zipf distribution, the access probability of the i^{th} ($1 \leq i \leq N$) data item is represented as follows

$$p_i = \frac{i^{-\theta}}{\sum_{j=1}^N j^{-\theta}} \quad (3.6)$$

where N is the number of items in the database and $0 \leq \theta \leq 1$.

When $\theta = 0$, the access pattern is uniform. As θ value is increased the skewness increases. When $\theta = 1$, it is the strict Zipf distribution. Large θ results in more “skewed” access distribution. To answer a query, the client first checks its local cache. If the data value for the requested item with respect to the current location is available, the query is satisfied locally. Otherwise, the client submits the query and its current location to the server and retrieves the data through the downlink channel. The *move process* controls the movement pattern of the client using the parameter *MovingInterval*. After the client keeps moving at a constant velocity for a time period of *MovingInterval*, it changes the velocity randomly: the next moving direction (represented by the angle relative to the x-axis, counter clock wise taken as positive) is also selected randomly between 0° to 360° , and the next speed is selected randomly between *MinSpeed* and *MaxSpeed*. If the difference between *MinSpeed* and *MaxSpeed* is low the mobile users move with almost same velocity. The client is assumed to have a cache of fixed size, which is a *CacheSizeRatio* ratio of the database size.

3.5.3 Server

The server is modeled by a single process that services the requests from clients. The requests are buffered at the server if necessary, and an infinite queue buffer is assumed. The FCFS service principle is assumed in the model. To answer a location-dependent query, the server locates the

correct data value with respect to the specified location. Since the main concern of this thesis is the cost of the wireless link(i.e. transmission time, receiving time and disconnections), which is more expensive than the wired-link and disk I/O costs(i.e. disk access time), the overheads of request processing and service scheduling at the server are assumed to be negligible in the model.

3.6 PERFORMANCE EVALUATION

This section describes the performance parameters and metric used for simulation and analyze the results of the simulation

3.6.1 Performance Parameters

This subsection describes the parameters used for simulation. The default values of different parameters used in the simulation experiments are given in Table 3.3. They are chosen to be in accordance with those used in earlier studies [14,50,44,68].

Table 3.3 Configuration Parameters and Default Parameter Settings for Simulation Model

| Parameter | Description | Setting |
|-----------------------|---|---|
| <i>Size</i> | size of the rectangle service area | 4000m*4000m, 44000m*27000m |
| <i>ItemNum</i> | number of data items in the database | 500 |
| <i>ScopeNum</i> | number of different values at various locations for each item | 110, 215 |
| <i>DataSize</i> | size of a data value | 128 bytes |
| <i>UplinkBand</i> | bandwidth of the uplink channel | 19.2 kbps |
| <i>DownlinkBand</i> | bandwidth of the downlink channel | 144 kbps |
| <i>FloatSize</i> | size of a floating-point number | 4 bytes |
| <i>QueryInterval</i> | average time interval between two consecutive queries | 50.0 s |
| <i>MovingInterval</i> | time duration that the client keeps moving at a constant velocity | 100.0s |
| <i>MinSpeed</i> | minimum moving speed of the client | 1m s ⁻¹ , 5 m s ⁻¹ |
| <i>MaxSpeed</i> | maximum moving speed of the client | 2m s ⁻¹ , 10 m s ⁻¹ |
| <i>CacheSizeRatio</i> | ratio of the cache size to the database size | 10 % |
| θ | skewness parameter for the Zipf access distribution | 0.5 |

Experiments are performed using different workloads and system settings. The performance analysis presented here compares the effects of different parameters such as *QueryInterval*,

MovingInterval and *DataSize* on the performance of our and other algorithms. In order to get the true performance for each algorithm, we collect the results only after the system becomes stable, which is defined as the time when the client caches are full [14,50,74,75]. Each simulation is run for 20,000 client issued queries and each result obtained in the experiment is taken as the average of 10 simulation runs with Confidence Interval of 96 percent.

For simulation purpose, we assume that all data items follow the same scope distribution in a single set of experiments. Two scope distributions with 110 and 215 valid scopes are used (see Figure 3.8). Since the average valid scope areas differ for these two scope distributions, different moving speeds are assumed, i.e., the pair of (*MinSpeed*,*MaxSpeed*) is set to (1,2) and (5,10) for Scope Distribution 1 and Scope Distribution 2, respectively. The LRU cache replacement policy is employed for cache management. For calculating data distance between valid scope (either a polygon or a circle) and current location we select a reference point for each valid scope and take the distance (Euclidean distance) between the current location and the reference point. For polygonal valid scope, the reference point is defined as the endpoint that is closest to the current location and for circular valid scope, it is defined as the point where the circumference and the line connecting the current location and the center of the circle meet.

3.6.2 Performance Metric

Our primary performance metric is *cache hit ratio*. This is because other performance can be derived from the cache hit ratio. Cache hit ratio can be defined as the ratio of the number of queries answered by the client's cache to the total number of queries generated by the client. Specifically, higher the cache hit ratio, higher is the local data availability, less is the uplink and downlink costs and the battery consumption [14,50].

3.6.3 Comparison of Location-Dependent Invalidation Schemes

This subsection examines the performance of different location-dependent invalidation schemes, namely, CEB, CEB_G, CEFAB and CEFAB_G. Figures 3.9 to 3.14 show the cache hit ratio for both scope distributions (Figure 3.8) under various query intervals, moving intervals and data sizes.

Figure 3.9 shows that for Scope Distribution 1, CEFAB, CEB_G and CEFAB_G, perform better than over CEB scheme for all query intervals (from 20s to 200s). We observe that the generalized procedure of selecting the best candidate valid scope in CEB_G helps to achieve maximum improvement of 6 % better than CEB. It can also be seen that CEFAB performance is

up to 4 % better than CEB and CEFAB_G performs up to 9 % better than CEB. Similar performance is observed in Figure 3.10 for Scope Distribution 2. Here, the average improvement of CEFAB_G, CEB_G, CEFAB is 3%, 2 % and 1.5% over CEB respectively.

CEFAB_G performs better than all. It is the increase in precision provided by the algorithms while selecting candidate valid scopes that improves performance. As the query interval increases, cache hit ratio of all algorithms decreases, because the probability that the user remains in the same valid scope decreases and hence the gains achieved by using a candidate valid scope of high precision decreases.

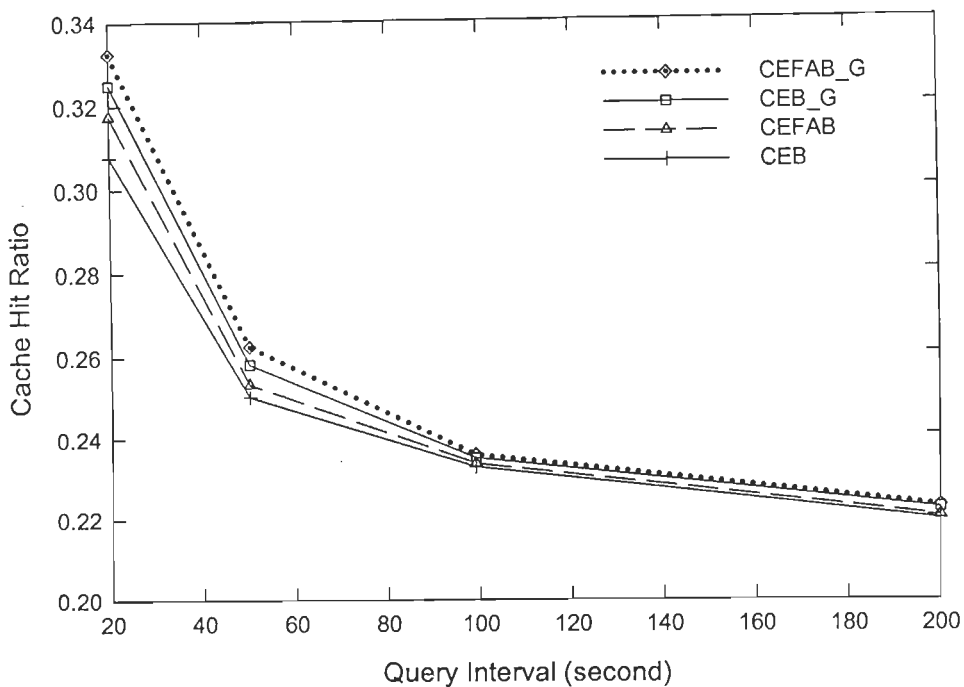


Figure 3.9 Cache Hit Ratio of Invalidation Schemes vs. Query Interval (Scope Distribution 1)

The effect of Moving Interval (varied from 20s to 400s) on all invalidation algorithms is shown in Figure 3.11 and Figure 3.12. The longer the moving interval, the less frequently the client changes velocity and direction and, hence, less random is the client's movement. In Figure 3.11 (for Scope Distribution 1) CEB_G gives an average improvement of 6 percent over CEB.

The performance of CEFAB depends on the Moving Interval, so smaller moving interval reflects highly random user and small future prediction of path, which results in less precise selection of valid scope. On the other hand, larger moving interval means future prediction path is long and thus highly precise valid scope selection is done. Hence, CEFAB shows improvement

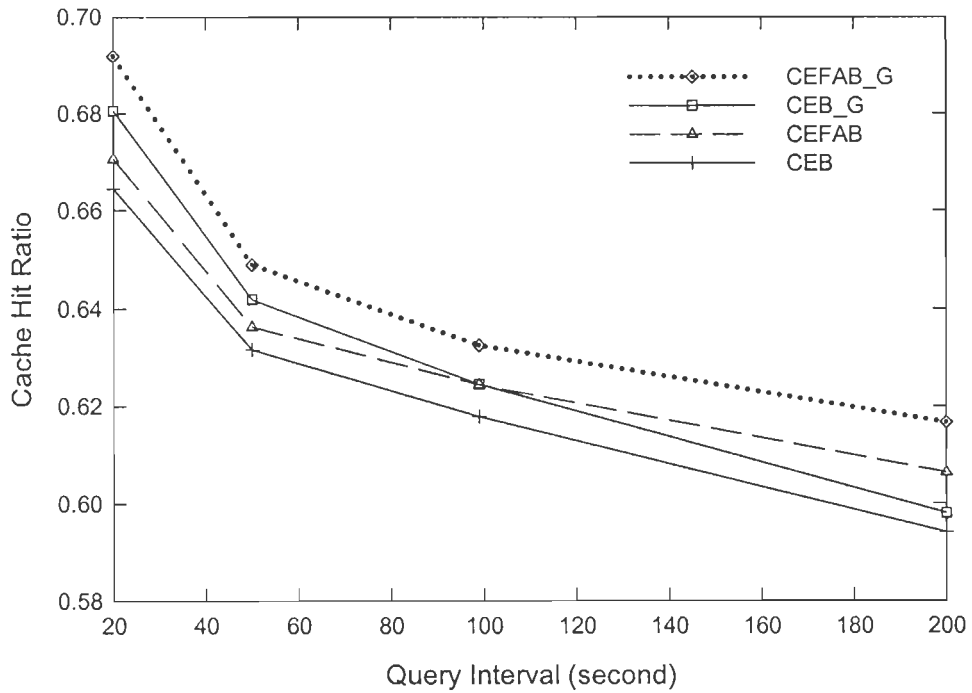


Figure 3.10 Cache Hit Ratio of Invalidation Schemes vs. Query Interval (Scope Distribution 2)

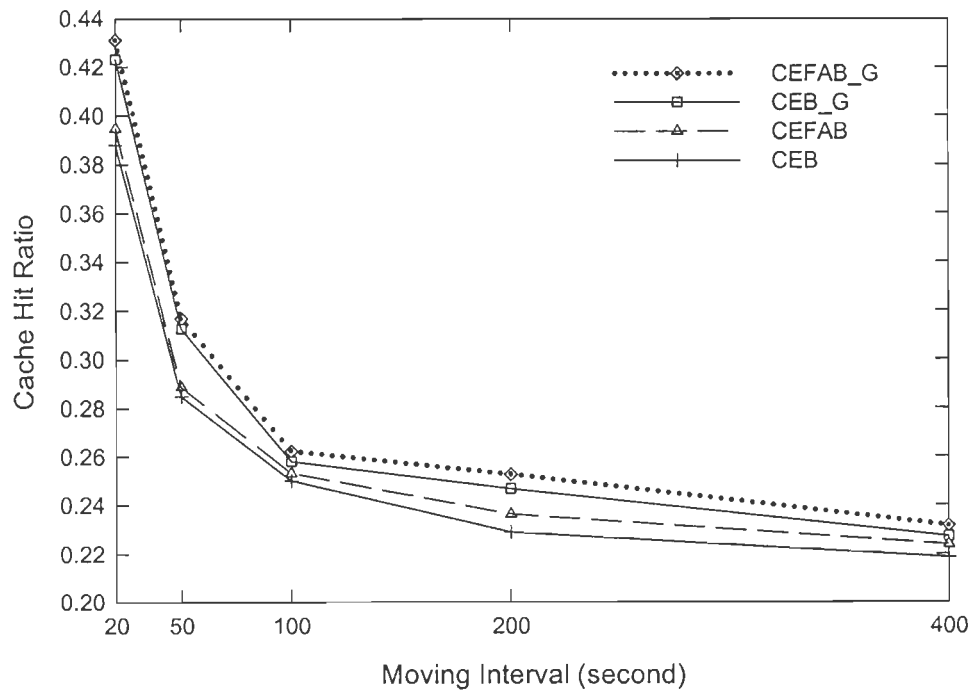


Figure 3.11 Cache Hit Ratio of Invalidation Schemes vs. Moving Interval (Scope Distribution 1)

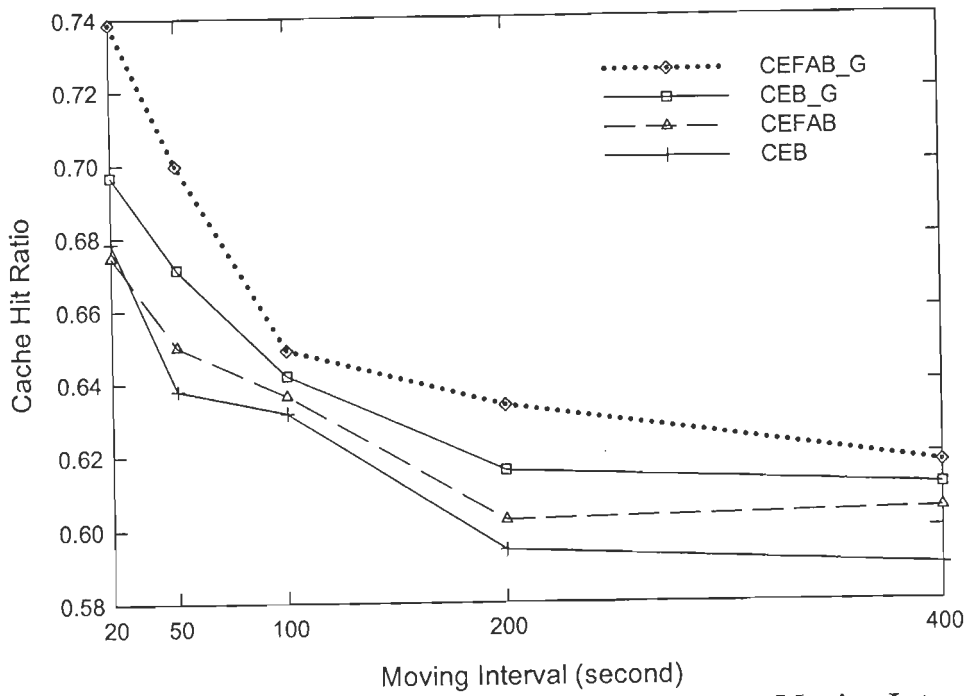


Figure 3.12 Cache Hit Ratio of Invalidation Schemes vs. Moving Interval (Scope Distribution 2)

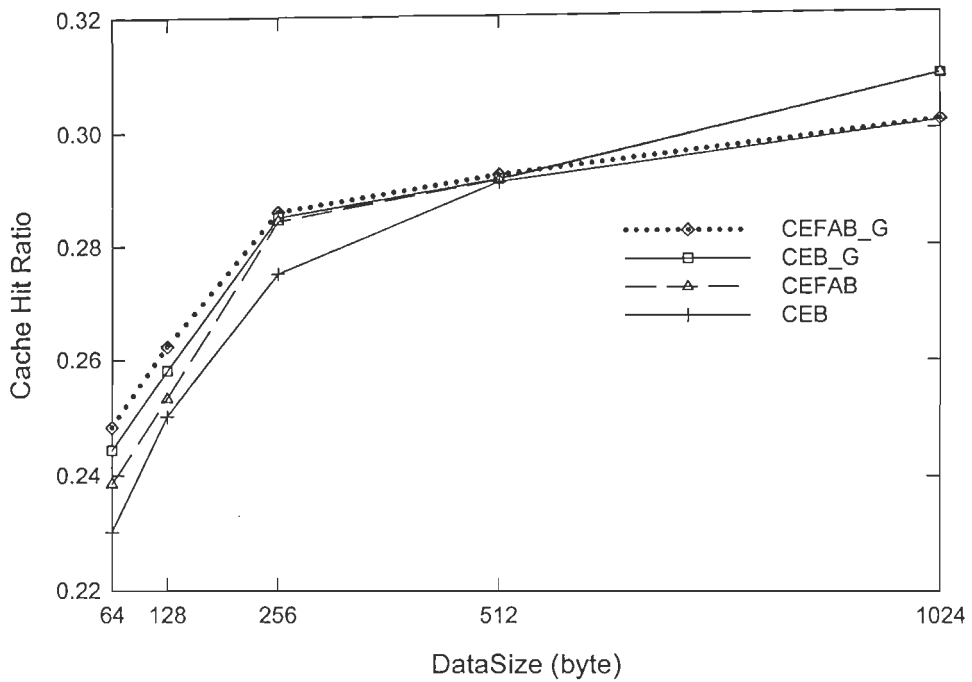


Figure 3.13 Cache Hit Ratio of Invalidation Schemes vs Data Size (Scope Distribution 1)

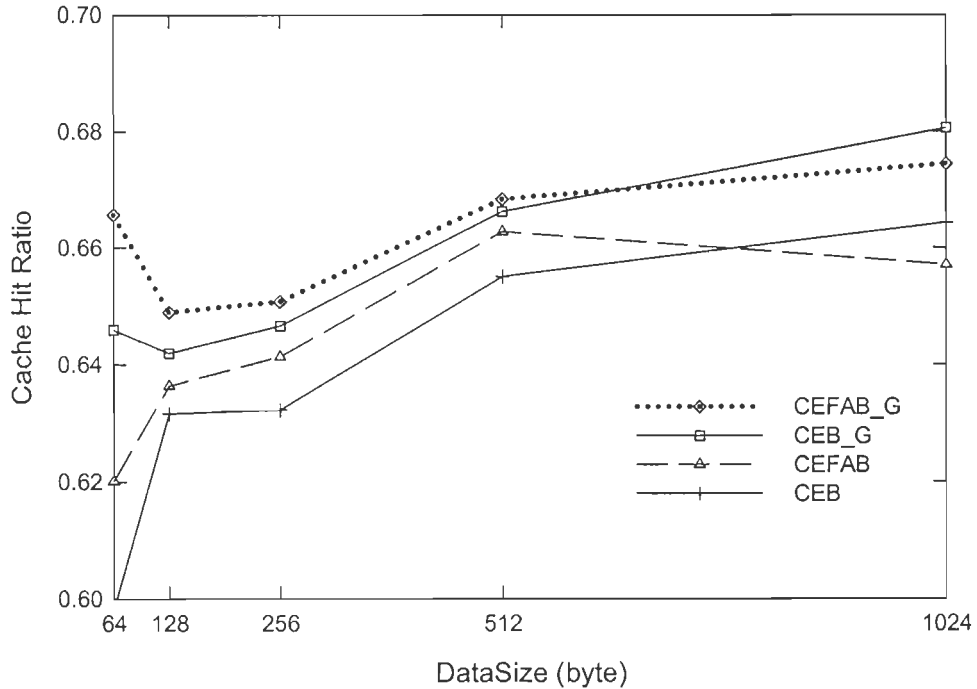


Figure 3.14 Cache Hit Ratio of Invalidation Schemes vs Data Size (Scope Distribution 2)

over CEB, with increase in moving interval. CEFAB shows an improvement of about 3 percent over CEB for different moving intervals.

CEFAB_G which combines both generalized selection procedure of valid scope and the future access shows the best performance with an average improvement of 8 % over CEB. Similar improvements in performance can be seen for Scope Distribution 2 in Figure 3.12 for algorithms CEFAB_G, CEB_G, CEFAB with the average improvements of 10 %, 4 % and 1.5 % over CEB respectively.

Figures 3.13 and 3.14, show the results when the data size is varied from 64 bytes to 1024 bytes. When the data size increases, the overhead of the invalidation information becomes relatively smaller and all of the schemes perform better. It is observed for both scope distributions, that for data size of 64 bytes, the overhead of invalidation information is large and algorithms CEFAB_G, CEB_G and CEFAB give better performance over CEB. As the data size increases, their performance reduces as compared to CEB.

3.7 CONCLUSIONS

In this chapter, we explored cache invalidation issues for location-dependent data under a geometric location model. We observed that modifying CEB to consider more choices in each iteration gives better results. Based on this observation, we proposed a generalized algorithm CEB_G that selects the best suitable candidate for valid scope to maximize the caching efficiency. Though, CEB_G takes more computational time, but as these candidate valid scopes can be calculated only once and then stored at the server along with its actual valid scope, it is acceptable. However, CEB_G improves precision by selecting more precise representation of valid scope as compared to CEB. We compared its performance with the existing CEB algorithm. We also introduced a new metric, *Future Access*, which takes into account the future movement behavior of client and proposed CEFAB and CEFAB_G algorithms based on it. A number of simulation experiments have been conducted to evaluate the performance of the proposed cache invalidation schemes. The results show that algorithms CEB_G, CEFAB, and CEFAB_G with different system settings, give better performance than CEB. Among the proposed algorithms, CEFAB_G gave the best performance. But, computational overhead at server for CEFAB_G and CEB_G is higher than CEFAB. Moreover, in CEFAB and CEFAB_G, client has to send additional information to the server, which requires extra computation at client's end, as compared to CEB_G. Thus, for low resource client CEB_G is preferred. Depending on the resources at the server, choice can be made between CEFAB and CEFAB_G.

CHAPTER 4

PREDICTED REGION BASED CACHE REPLACEMENT

4.1 INTRODUCTION

Since a mobile client has only limited cache space, cache replacement is another important issue for client cache management. In Chapter 3, we focused on location-dependent cache invalidation. This chapter focuses on location-dependent cache replacement.

Caching frequently accessed data items on the client side improves the system performance in wireless networks. Due to the limitation of the cache size, it is impossible to hold all the accessed data items in the cache. As a result, cache replacement algorithms are used to find a suitable subset of data items for eviction when the cache is full. Cache replacement algorithms have been extensively studied in the context of operating system, virtual memory management and database buffer management. These cache replacement policies rely on the temporal locality of user's access pattern to improve cache performance. In this context, cache replacement algorithms usually maximize the cache hit-ratio by attempting to cache the items that are most likely to be accessed in the future. These algorithms, however, are not ideal in supporting mobile clients using location-dependent applications. As mobile clients can move freely from one location to another, their access pattern not only exhibits temporal locality, but also exhibits spatial locality. In order to ensure efficient cache utilization, it is important to take into consideration the location and movement direction of mobile clients, when performing cache replacement.

A cache replacement policy determines, which data item(s) should be deleted from the cache when the cache does not have enough free space to accommodate a new item [14,68,74]. The problem can be formally defined as follows. Let V denotes the set of all the cached data items with total size S_V in mobile client's cache of size $cacheSize$. The ultimate goal of a cache replacement policy is to determine the set of objects V^* to evict from the client's cache when a new data item of size S_{new} has to be inserted in cache and $(S_V + S_{new}) \geq cacheSize$, such that the cost of the objects evicted are minimized. Many existing cache replacement algorithms employ a cost function $cost(i)$ to calculate the cost of data item i in cache which depends on different factors such as time since last access, entry time of the item in the cache, transfer time, item expiration time, valid scope of item, data distance of item and so on. A cache replacement policy

can be viewed as a specialized instance of the well-known knapsack problem [68,74,75]. The objective is to minimize the cost of items that are purged from cache while satisfying a size constraint. The formulated problem is shown below:

The objective is to *minimize*

$$\sum_{i \in V^*} cost(i) \quad \forall V^* (V^* \subseteq V) \quad (4.1)$$

subject to the *constraint*

$$\sum_{i \in V^*} s_i \geq s_{new} \quad (4.2)$$

where, s_i is the size of i^{th} data item. It is well known that the knapsack problem is NP complete. Although, there is no optimal solution to the problem, but when the data size is relatively small compared to cache size, heuristics can be used to find a sub-optimal solution in polynomial time. Generally, the heuristics chosen to solve this knapsack problem is that when a client needs to insert a new item into the cache and the cache is full, the cached item with the lowest cost is removed until there is enough available space in the cache to store the new item [14,68,74,75].

In mobile networks, where clients utilize location dependent information services, clients access pattern do not only exhibit temporal locality, but also exhibit dependence on the location of data, location of the client and direction of the client's movement [14,27]. Hence, relying solely on temporal locality while making cache replacement decisions will result in poor cache hit ratio in LDIS. To overcome this problem, several location-aware cache replacement policies [14,50,68,90,98] have been proposed for location dependent information services. None of these cache replacement policies are suitable if client changes its direction of movement quite often. Existing cache replacement policies only consider the data distance (directional/undirectional) but do not try to predict the region or area where the client can be in near future. Very few of these policies [14,50,68] account for the location and movement pattern of mobile clients.

In this chapter, we propose cache replacement algorithms based on the predicted region of user's presence in near future. These algorithms predict an area in the vicinity of client's current position, and give priority to the cached data items that belong to this area irrespective of the client's movement direction. Based on the predicted region we propose Predicted Region based Cache Replacement Policy (PRRP), Prioritized Predicted Region based Cache Replacement Policy (PPRRP) and Weighted Predicted Region based Cache Replacement Policy (WPRRP). We also compared our cache replacement policies with other existing cache replacement policies

such as PAID, FAR, and Manhattan for LDIS.

The next section describes our proposed cache replacement policies. The system model used is same as described in section 3.2.

4.2 CACHE REPLACEMENT POLICIES BASED ON PREDICTED REGION

Traditional cache replacement policies, due to their temporal nature, consider access probability as the most important factor that affects cache performance. A data item with least access probability is evicted from cache. Since LDIS is spatial in nature, distance of data item from client's current position and its valid scope area should also be taken into account for cache replacement. In LDIS, the server responds to the user query with suitable data value that depends on client's current location. Greater the distance of valid scope of data from the user's current position, lower is the chance that client will enter in the valid scope area of that data in near future. Thus, it is better to eject the farthest data item value when replacement takes place. Moreover, because the client is mobile, its position at the time of next query will be different from its current position. Therefore, the client's movement should also be taken into account. *Locations in the direction opposite to client's movement have very low chance of being revisited soon, though they may be very close to it.* Based on this reasoning, existing cache replacement schemes like FAR and PAID(Directional) [14,50] assign higher priorities to data items in the client's direction of movement giving very low priority to the items in the direction opposite to user's movement. However, with client's random movement patterns, it is not always necessary that client will continue moving in the same direction. Therefore, *evicting data values which are in the opposite direction to client's movement but are very close to client current position may degrade the overall performance* [14,50].

Basic Idea:

When client's movement pattern is random, retaining the data items in the direction of user movement and discarding the data items that are in the opposite direction of user movement may not always improve the performance, because in random movement client's direction may change frequently. Therefore, our cache replacement policy considers the predicted region of user presence in near future (rather than considering the direction of user movement only) while selecting a data item for replacement. The predicted region is based on the client's current movement pattern. We show that it is useful to calculate the data distance with respect to this

region, so that the data items in the vicinity of client's current position are not purged from the cache. Valid scope area of the data item and space required to store the data item in cache are also used in selecting an item for replacement. This is because the client has a higher chance of being in larger region rather than in smaller regions and keeping smaller size data items in cache helps to accommodate a large number of data items in the cache. Hence, our cache replacement policy selects a victim with low access probability, small valid scope area falling outside the predicted region and large data size.

Approach:

We make use of discrete model for client's movement path as described in Section 3.2 and used in [14,44,68]. Assuming a predefined path of mobile user or a predefined destination is generally not possible unless we are dealing with a case where the user is moving in a train or a ship and the entire path of the user is known well in advance. For discrete model, the direction and velocity of the user are known for current MI. At the end of each MI, direction is selected randomly between 0° to 360° degrees and the velocity between minimum (v_{\min}) and maximum speed (v_{\max}) of the client. This assumption allows us to predict a region of user's presence in the near future.

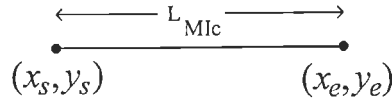


Figure 4.1 Current Moving Interval

Let v_c be the velocity in the current moving interval MI_c , L_{MIc} be the length of MI_c along direction θ_c and (x_s, y_s) and (x_e, y_e) be the starting and end points of MI_c respectively (see Figure 4.1). Assuming that the next L_{MI} is almost same, after reaching (x_e, y_e) mobile user selects the direction randomly and covers a distance of L_{MI} in next MI. We can predict the region of user presence in near future by the circle with radius L_{MIc} and centre (x_e, y_e) as shown in Figure 4.2.

One of the advantages of using predicted region is that it dynamically changes with speed of client and Moving Interval and so takes into account the random movement of client. Our cache replacement policy uses this region to calculate the data distance in various ways. For each case we define a new cache replacement policy. They are as follows:

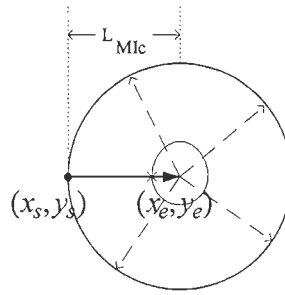


Figure 4.2 Predicted Region

4.2.1 PREDICTED REGION BASED CACHE REPLACEMENT POLICY (PRRP)

In this cache replacement policy, we differentiate between the data items inside and outside the predicted region and calculate their distances as follows:

- (i) **The distance of data items outside the predicted region is calculated from the center of the circle**

Instead of assigning equal low priority to the data items that lie outside the predicted region, we assign priorities to different items based on their distance from the center of circle (given in Figure 4.2), as it is the farthest location that the user will reach in its current MI. A fair selection is thus made for all data item that lies outside the predicted region.

- (ii) **The distance of data items inside the predicted region is the minimum distance of the data item from any point on the circle and from its center.**

Logically, we can think of predicted region as a reference of assigning priority to data items that the user will visit in very near future. Generally, if we consider circle's center as the current position of user and if we calculate the distance of data items from it, then the maximum data distance is same as the radius of the circle and the minimum is zero if it lies in the center. But, then it is same as (i) above. For assigning a much higher priority to data items in the predicted region, we calculate its distance from all points on the circle as well as from the center of circle and assign its minimum distance as its data distance (thus, the distance is never more than half of the radius). Using this method, a data item near to center and near to circle edge gets nearly equal priority. However, this does not affects the performance much, as the priority of the data item inside the predicted region is greater

than those outside the region, since outside data item has minimum distance greater than the radius of circle.

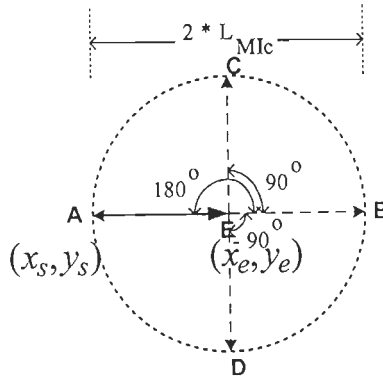


Figure 4.3 Predicted Region with Extreme Cases

Calculating the distance of data items from the points on the circle, means, calculating the distance from all the points that lies on the circumference of it. Since it is impossible to calculate the distance of data items valid scope from all the points on the circle, we choose only four points (simplest possible points) as explained later. One can choose more points depending on the application and its permissibility of complexity. We consider following four extreme cases:

- user travels in same direction as θ_c in new MI.
- user travels in direction opposite to the θ_c in new MI, i.e. $\theta_c + 180^\circ$, and
- user travels in direction perpendicular to θ_c in new MI, i.e. $\theta_c \pm 90^\circ$.

Figure 4.3 shows the four point on the circle (A, B, C, D) with $\theta_c = 0^\circ$. So a user can reach points B, A, C and D in the above worst case at the end of next MI with velocity v_c , and direction θ_c , $\theta_c + 180^\circ$, $\theta_c + 90^\circ$ and $\theta_c - 90^\circ$ respectively.

Now, we define our cost based cache replacement policy PRRP which takes into consideration access probability, data distance from the predicted region, valid scope area and data size in cache. Each cached data object is associated with its replacement cost. When a new data object needs to be cached and there is insufficient cache space, the object(s) with lowest replacement cost is removed until there is enough space to cache new object. The cost of replacing a data value j of data item i in client's cache is calculated as:

$$PRRP_Cost_{i,j} = \begin{cases} \frac{P_i \cdot A(vs_{i,j})}{S_{i,j}} \cdot \underset{\forall p \in Pred_Point}{\text{maximum}} \left\{ \frac{1}{D(vs_{i,j}, p)} \right\} & \text{if } vs_{i,j} \in Pred_Reg \\ \frac{P_i \cdot A(vs_{i,j})}{S_{i,j}} \cdot \frac{1}{D(vs_{i,j})} & \text{if } vs_{i,j} \notin Pred_Reg \end{cases} \quad (4.3)$$

where,

P_i : access probability of data item i

$A(vs_{i,j})$: area of valid scope $vs_{i,j}$ for data value j of data item i

$S_{i,j}$: storage space (size) needed to store data value j and its valid scope $vs_{i,j}$

$D(vs_{i,j}, p)$: distance between $vs_{i,j}$ and point (p_x, p_y)

$D(vs_{i,j})$: distance between the center of the predicted region and the valid scope $vs_{i,j}$

$Pred_Reg$: predicted region, and

$Pred_Point$: set of points consisting of extreme case points and center of $Pred_Reg$

4.2.2 PRIORITIZED PREDICTED REGION BASED CACHE REPLACEMENT POLICY (PPRRP)

In this cache replacement policy, we differentiate between the data items inside and outside the predicted region as well as data items nearer to user's current position within the predicted region and calculate their distances as follows:

- (i) The distance of data items outside the predicted region is calculated from the centre of the circle.
- (ii) The distance of data items inside the predicted region is calculated as the minimum of (L_{MIC} , distance of the valid scope from the current position of the user).

Calculating the distance of data items in this way ensures that

- (i) Items outside the predicted region always have lower priority than the items inside the predicted region.
- (ii) Items inside the predicted region, close to the user have higher priority.

Now, we define our cost based cache replacement policy PPRRP that considers access probability, predicted region based data distance, valid scope area and size of the data.

Associated with each cached data object is the replacement cost. When a new data object needs to be cached and there is insufficient cache space, the objects with lowest replacement cost are removed until there is enough space to cache new object. The cost of replacing a data value j of data item i in client's cache is calculated as:

$$PPRRP_Cost_{i,j} = \begin{cases} \frac{P_i \cdot A(vs_{i,j})}{S_{i,j}} \cdot \frac{1}{\text{minimum}\{L_{MC}, D(vs_{i,j})\}} & \text{if } vs_{i,j} \in \text{Pred_Reg} \\ \frac{P_i \cdot A(vs_{i,j})}{S_{i,j}} \cdot \frac{1}{D'(vs_{i,j})} & \text{if } vs_{i,j} \notin \text{Pred_Reg} \end{cases} \quad (4.4)$$

where,

P_i : access probability of data item i

$A(vs_{i,j})$: area of the valid scope $vs_{i,j}$ for data value j of data item i

$S_{i,j}$: storage space (size) needed to store data value j and its valid scope $vs_{i,j}$

$D(vs_{i,j})$: distance of the valid scope $vs_{i,j}$ from the current user position

$D'(vs_{i,j})$: distance of the valid scope $vs_{i,j}$ from the centre of the predicted region, and

Pred_Reg : predicted region

4.2.3 WEIGHTED PREDICTED REGION BASED CACHE REPLACEMENT POLICY (WPRRP)

To make our earlier schemes adaptable to different user movement patterns, we divide the service area into four sub regions (R₁-R₄) as given in Table 4.1.

| | Inside Predicted Region | Outside Predicted Region |
|----------------------|----------------------------|-----------------------------|
| In-Direction | R ₂ | R ₃ |
| Out-Direction | R ₁ | R ₄ |

This not only helps us in differentiating between the data items inside and outside the region but also between the data items along and opposite to the direction of user movement. We study the effect of assigning different weights to the data items in different regions while calculating

their distance from the current position of the user.

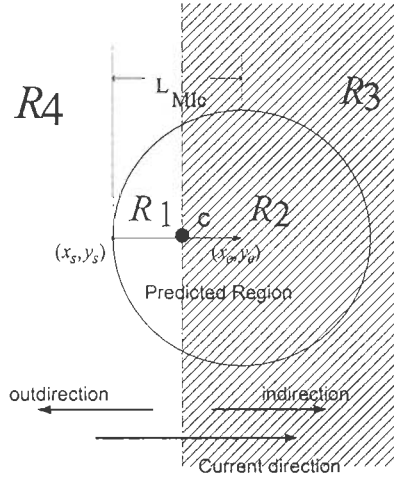


Figure 4.4 Sub Regions within Service Region

The sub regions are shown in Figure 4.4, where c is the current position of client. Weights W_1 , W_2 , W_3 and W_4 are assigned to R_1 , R_2 , R_3 and R_4 respectively. Data items falling in the sub region R_i is assigned a weight of W_i . By selecting appropriate value of W_i , we can give different priorities to the items in those regions. This is done by multiplying respective weights to the data distance of data items (specific cases are considered below).

We call our replacement policy *weighted predicted region based cache replacement policy* (WPRRP). The cost of replacing a data value j of data item i in client's cache in WPRRP is calculated as:

$$\begin{aligned}
 WPRRP_Cost_{i,j} &= \frac{P_i \cdot A(vs_{ij})}{S_{ij}} * \text{weighted_distance} \\
 &= \frac{P_i \cdot A(vs_{ij})}{S_{ij}} \cdot \frac{W_K}{D(vs_{ij})} \\
 &= W_K \cdot \frac{P_i \cdot A(vs_{ij})}{S_{ij} \cdot D(vs_{ij})} \quad \text{where } vs_{ij} \in R_K \wedge R_K \in \{R_1, R_2, R_3, R_4\} \quad (4.5) \\
 & \quad \quad \quad 0 < W_K \leq 1
 \end{aligned}$$

where,

P_i : access probability of data item i

$A(vs_{i,j})$: area of the valid scope $vs_{i,j}$ for data value j of data item i

$S_{i,j}$: storage space (size) needed to store data value j and its valid scope $vs_{i,j}$

W_K : weight of the R_K sub region, and

$D(vs_{i,j})$: distance of the valid scope $vs_{i,j}$ from the current user position

To demonstrate the adaptiveness of this scheme, let us consider following three cases with different weights in range 0.1 to 1, where 1 is the highest priority and 0.1 is the lowest priority assigned as weights.

Case 1: Assigning Priority to in-direction data items only ($W_1=0.1$; $W_2=1$; $W_3=1$; $W_4=0.1$)

The in-direction items falls in regions R_2 and R_3 so they are given the highest priority (i.e. 1) and the out-direction data items falls in R_1 and R_4 region and they were given lowest priority (i.e. 0.1), because priority is given to the in-direction items only. This case is same as those taken in FAR and PAID(Directional) as far as data item with respect to direction of client is concerned.

Case 2: Assigning highest priority to data items in the predicted region and in the direction of user movement over data items outside the predicted region and out-direction. ($W_1=1/3$; $W_2=1$; $W_3=1/2$; $W_4=1/4$)

In this case, weight assignment depends on two factors: direction and predicted region. Setting W_2 and W_3 larger than W_1 and W_4 gives higher priority to data items in the direction of user movement. Based on predicted region, R_2 should be given higher priority than R_3 and R_1 over R_4 . We therefore assign highest value to $W_2 (=1)$ as it is in the predicted region as well as in the direction of user's movement. W_3 is assigned the weight of $1/2$ as it is in the direction of user movement. Starting from highest priority R_2 is assigned the weight of 1 and R_3 is assigned the weight of $1/2$. Since W_1 and W_4 should have lower weights than W_2 and W_3 , so W_1 is assigned weight of $1/3$ and W_4 as it falls outside the predicted region, and in opposite to the direction of user's movement, it is assigned the lowest weight of $1/4$. The combination of weights comes out to be $W_2=1$, $W_3=1/2$, $W_1=1/3$ and $W_4=1/4$.

Case 3: Assigning highest priority to Predicted Regions over Non-predicted ones ($W_1=1$; $W_2=1$; $W_3=1/2$; $W_4=1/2$)

In this case, priority is given to data items only based on predicted regions. Hence, R_1 and R_2 are assigned highest priority by setting $W_1 = W_2 = 1$ as compared to R_3 and R_4 , as W_3 and W_4 are both set to $1/2$.

Let these replacement policies, be called as WPRRP-1, WPRRP-2, and WPRRP-3 for case1,

case2 and case3 respectively.

4.3 SIMULATION MODEL

To study the performance of the above mentioned protocols we used the same simulation model and scope distributions as described in section 3.5 with following additional features:

The size of data value varies from S_{\min} to S_{\max} and has following three types of distributions [50,74]:

- **IncreasingSize:** The size S_i of data item i grows linearly as i increases, and is given by:

$$S_i = S_{\min} + \frac{(i-1)*(S_{\max} - S_{\min})}{ItemNum-1}, i = 1, \dots, ItemNum; \quad (4.6)$$

- **DecreasingSize:** The size S_i of data item i decreases linearly as i increases, and is given by:

$$S_i = S_{\max} - \frac{(i-1)*(S_{\max} - S_{\min})}{ItemNum-1}, i = 1, \dots, ItemNum; \quad (4.7)$$

- **RandomSize:** The size S_i of data item i falls randomly between S_{\min} and S_{\max} , given by:

$$S_i = S_{\min} + \lfloor prob()*(S_{\max} - S_{\min}) \rfloor, i = 1, \dots, ItemNum; \quad (4.8)$$

where, $prob()$ is a random function with uniformly distributed value between 0 and 1. The choice of the size distributions are based on previously published trace analysis [74]. Though, some researchers have shown that small data items are accessed more frequently than large data items, but recent web trace analysis shows that the correlation between data item size and access frequency is weak and can be ignored [70]. Combined with the skewed access pattern, IncreasingSize and DecreasingSize represent client's preference for frequently querying smaller items and larger items respectively. In other words, with IncreasingSize setting, the clients access the smallest item most frequently and with DecreasingSize setting, the clients access the largest item most frequently. RandomSize, models the case where no correlation between the access pattern and data size exists.

4.4 PERFORMANCE EVALUATION

This section describes the performance parameters and measures used for simulation and analyzes the results of the simulation.

4.4.1 Performance Parameters

The default values of different parameters used in the simulation experiments are given in Table 4.2. They are chosen to be the same as used in earlier studies [14,44,68].

Experiments are performed using different workloads and system settings. In order to get the true performance for each algorithm, we collect the result data only after the system becomes stable, which is defined as the time when the client caches are full [74,75]. Each simulation runs for 20,000 client issued queries and each result obtained in the experiment is taken as the average of 10 simulation runs with Confidence Interval of 96 percent.

For simulation purpose, we assume that all data items follow the same scope distribution in a single set of experiments. Two scope distributions with 110 and 215 valid scopes are used (see Figure 3.8). Since the average valid scope areas differ for these two scope distributions, different moving speeds are assumed, i.e., the pair of $(MinSpeed, MaxSpeed)$ is set to (1,2) and (5,10) for Scope Distribution 1 and Scope Distribution 2, respectively. The Caching-Efficiency-Based (CEB) [14,50] cache invalidation policy is employed for cache management. For calculating data distance between valid scope (either a polygon or a circle) and current location we select a reference point for each valid scope and take the distance (Euclidean distance) between the current location and the reference point. For polygonal valid scope, the reference point is defined as the endpoint that is closest to the current location and for circular valid scope, it is defined as the point where the circumference and the line connecting the current location and the center of the circle meet. Access probability for each data item is estimated by using exponential ageing method [14,50,74]. Two parameters are maintained for each data item i : a running probability P_i and the time of the last access to item t'_i . P_i is initialized to 0. When a new query is issued for data item i , P_i is updated using the following formula:

$$P_i = \alpha / (t_c - t'_i) + (1 - \alpha) P_i \quad (4.9)$$

where, t_c is the current system time and α is a constant factor to weigh the importance of most recent access in the probability estimate. Note that the access probability is maintained for each

data item rather than for each data value. If the database size is small, the client can maintain these parameters (i.e., P_i and t_i^l for each item i) for all items in its local cache. However, if the database size is large, these parameters will occupy a significant amount of cache space. To alleviate this problem, we set an upper bound to the amount of cache used for storing these parameters (5 percent of the total cache size in our simulation) and use the Least Frequently Used (LFU) policy to manage the limited space reserved for it.

Table 4.2 Configuration Parameters and Default Parameter Settings for Simulation Model

| Parameter | Description | Setting |
|------------------------|---|---|
| <i>Size</i> | size of the rectangle service area | 4000m*4000m, 44000m*27000m |
| <i>ItemNum</i> | number of data items in the database | 500 |
| <i>ScopeNum</i> | number of different values at various locations for each item | 110, 215 |
| <i>S_{min}</i> | minimum size of a data value | 64 bytes |
| <i>S_{max}</i> | maximum size of a data value | 1024 bytes |
| <i>UplinkBand</i> | bandwidth of the uplink channel | 19.2 kbps |
| <i>DownlinkBand</i> | bandwidth of the downlink channel | 144 kbps |
| <i>FloatSize</i> | size of a floating-point number | 4 bytes |
| <i>QueryInterval</i> | average time interval between two consecutive queries | 50.0 s |
| <i>MovingInterval</i> | time duration that the client keeps moving at a constant velocity | 100.0s |
| <i>MinSpeed</i> | minimum moving speed of the client | 1m s ⁻¹ , 5 m s ⁻¹ |
| <i>MaxSpeed</i> | maximum moving speed of the client | 2m s ⁻¹ , 10 m s ⁻¹ |
| <i>CacheSizeRatio</i> | ratio of the cache size to the database size | 10 % |
| θ | skewness parameter for the Zipf access distribution | 0.5 |
| α | constant factor | 0.25 |

4.4.2 Performance Metric

Same as described in section 3.6.2.

4.4.3 Comparison of Location-Dependent Cache Replacement Schemes

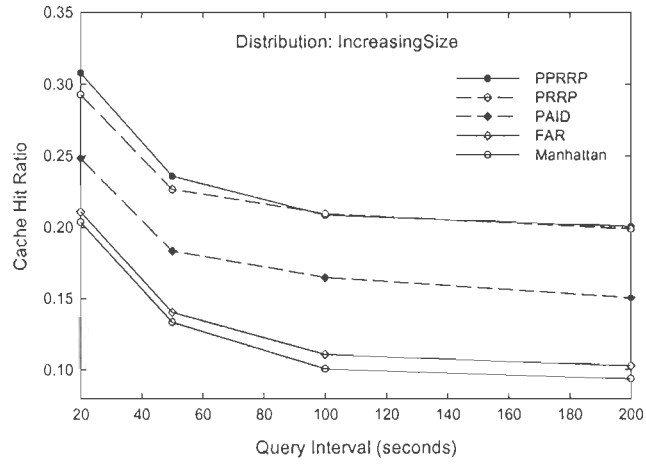
This subsection examines the performance of different location-dependent cache replacement policies, namely, PRRP, PRRRP, WPRRP-1, WPRRP-2 and WPRRP-3 with PAID, FAR and Manhattan. Figures 4.5 to 4.24 show the cache hit ratio for both scope distributions (see Figure

3.8) under various query intervals, moving intervals, cache sizes, client's speed and Zipf's distribution.

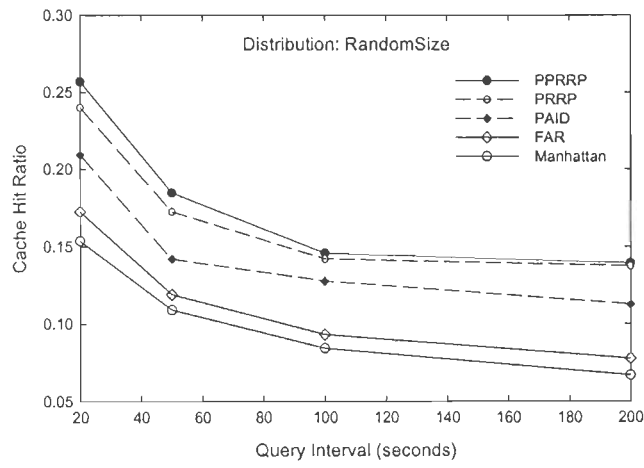
Effect of Query Interval

Query interval is the time interval between two consecutive client queries. In this set of experiments, we vary the mean query interval from 20 seconds to 200 seconds. Figures 4.5 to 4.8 show cache performance for both scope distributions and for the data distributions: IncreasingSize, RandomSize and DecreasingSize.

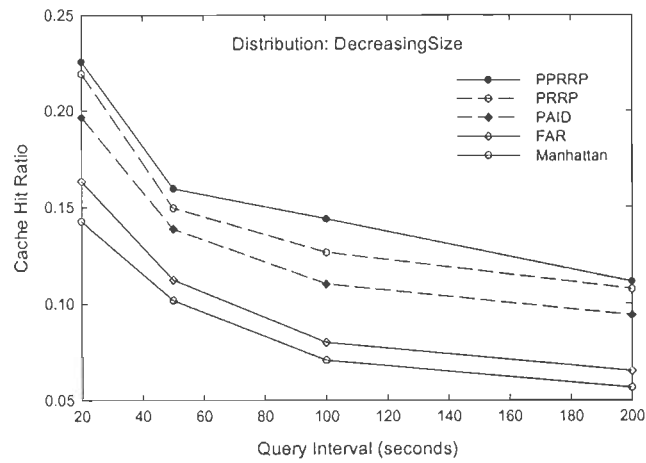
Results show that, when the query interval is increased, almost every scheme gets a worse performance. This is because, for a longer query interval when a new query is issued the client has a lower probability of residing in one of the valid scopes of the previously queried data items. When different cache replacement policies are compared, the proposed policies substantially outperform the existing policies. Figures 4.5 and 4.6 compare the performance of cache replacement policies over query interval for Scope Distribution 1. PRRP, which prefers object within the predicted region over the objects outside the predicted region obtains better performance than PAID with an average improvement of 25%, 17% and 12% for data distribution: IncreasingSize, RandomSize and DecreasingSize respectively. Similarly, PPRRP which improves PRRP by giving priority to the data objects that are nearer to the client's current position within the predicted region also shows better performance than PAID and PRRP. Average improvement of PPRRP over PAID is 28%, 21% and 19% for IncreasingSize, RandomSize and DecreasingSize respectively. As far as weighted predicted region based cache replacement policy is concerned, all WPRRP-1, WPRRP-2 and WPRRP-3 performed better than PAID for all query interval. But, WPRRP-3 gave the best performance over PAID as well as over PRRP and PPRRP. WPRRP-3 shows an average improvement of 29%, 27% and 21% for IncreasingSize, RandomSize and DecreasingSize respectively over PAID policy.



(a)

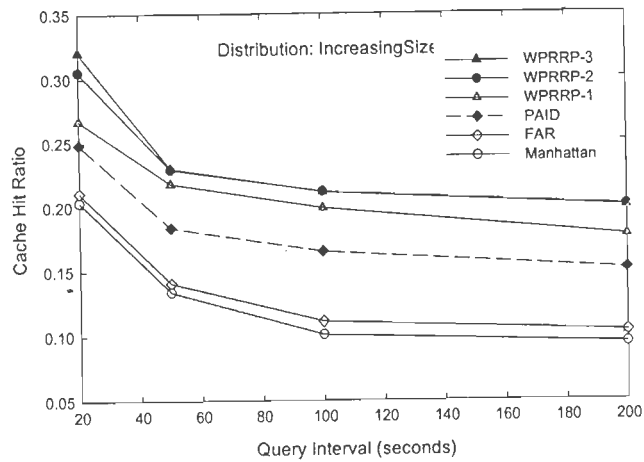


(b)

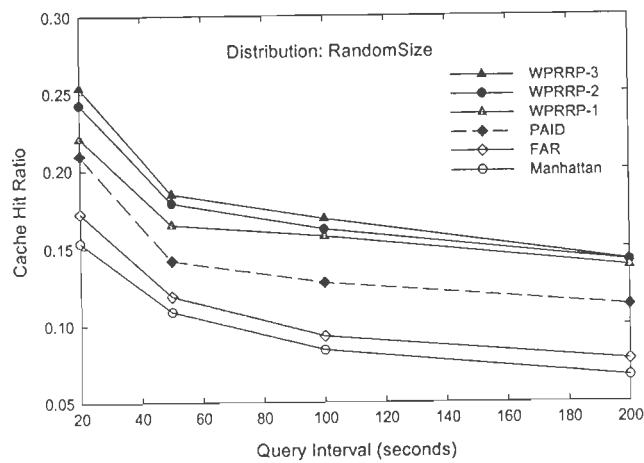


(c)

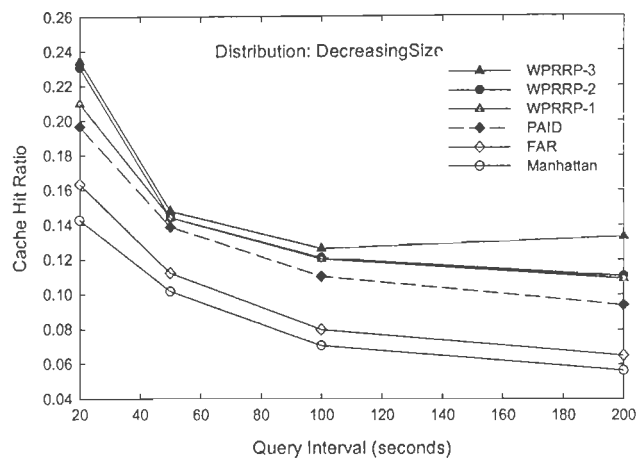
Figure 4.5 Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Query Interval (Scope Distribution 1)



(a)

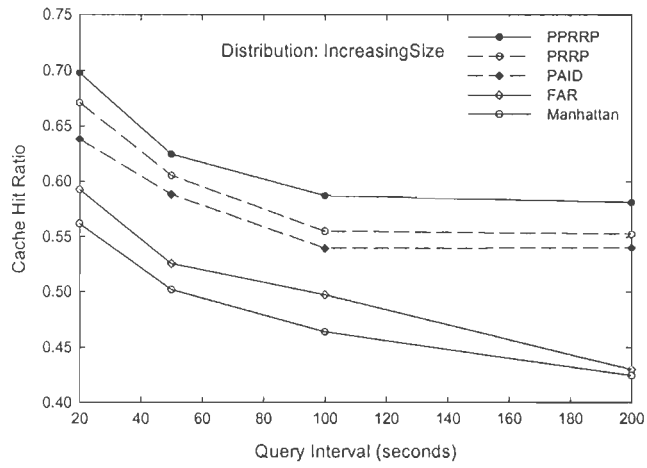


(b)

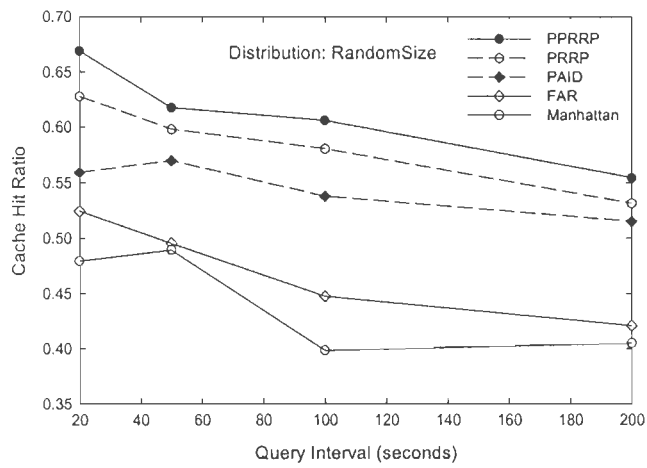


(c)

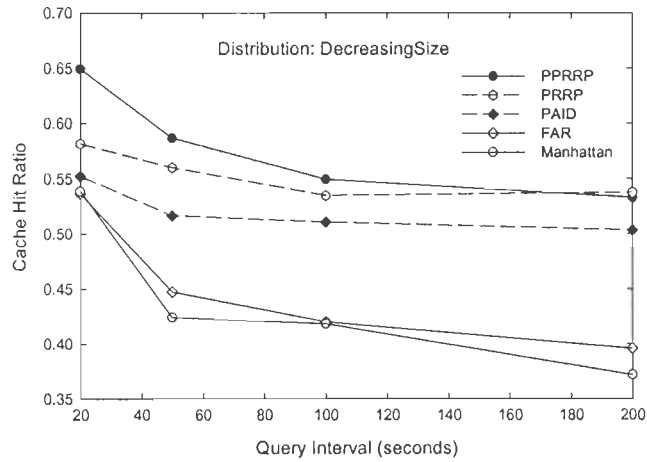
Figure 4.6 Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Query Interval (Scope Distribution 1)



(a)

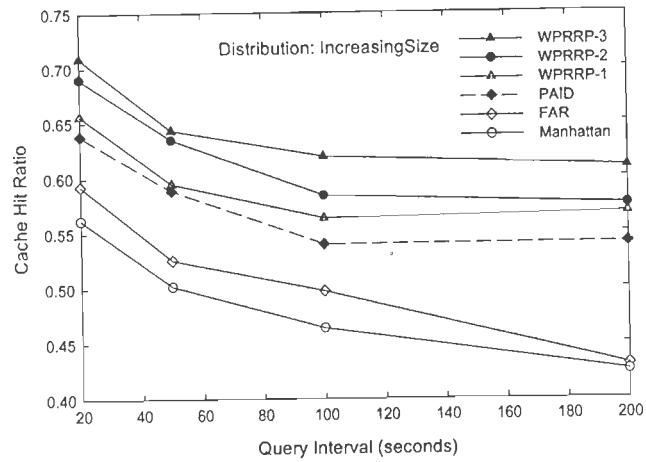


(b)

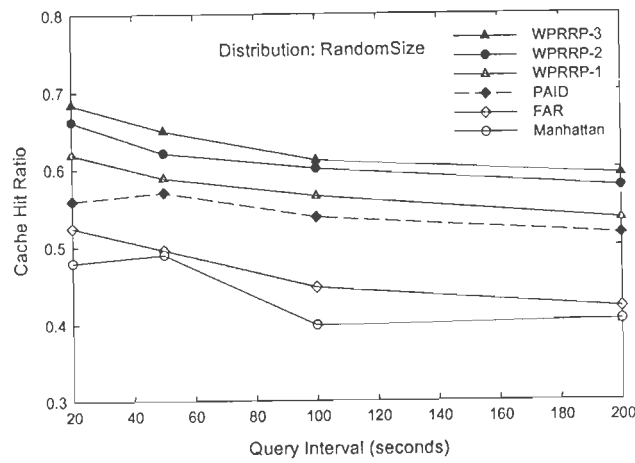


(c)

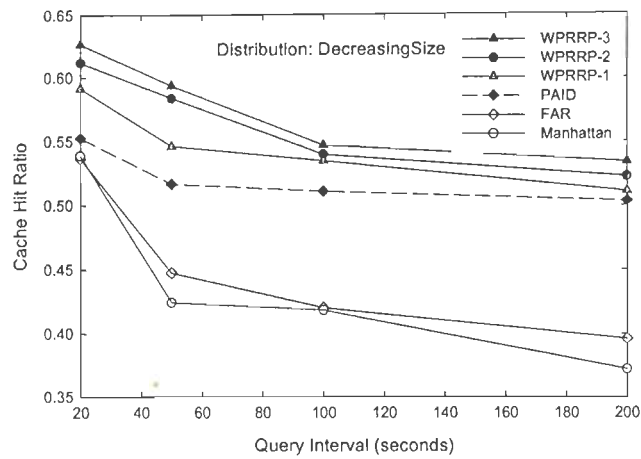
Figure 4.7 Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Query Interval (Scope Distribution 2)



(a)



(b)



(c)

Figure 4.8 Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Query Interval (Scope Distribution 2)

Figures 4.7 and 4.8 show the effect of change in query interval on the performance of cache replacement policies for Scope Distribution 2. It can be observed that the proposed policies show similar gains in performance for Scope Distribution 2 also as they were for Scope Distribution 1. The average improvement of PRRP, PRRP and WPRRP-3 over PAID for Scope Distribution 2 is shown in Table 4.3.

Table 4.3 Average Improvement of WPRRP-3, PRRP and PRRP over PAID on Different Mean Query Intervals (Scope Distribution 2)

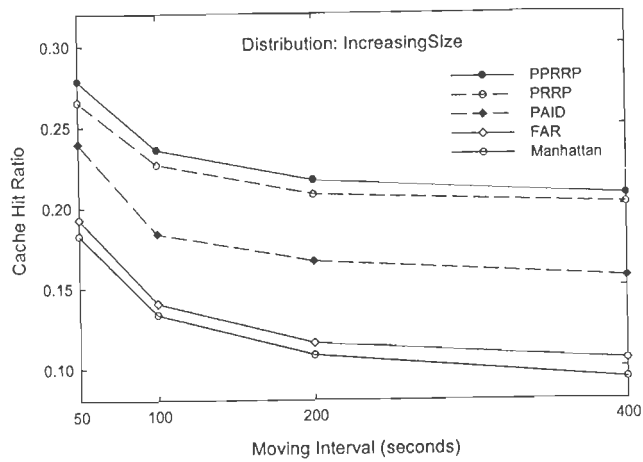
| Proposed Policies | IncreasingSize (%) | RandomSize (%) | DecreasingSize (%) |
|--------------------------|---------------------------|-----------------------|---------------------------|
| WPRRP-3 | 12 | 16.2 | 7.3 |
| PPRRP | 8 | 12.2 | 11 |
| PRRP | 3.3 | 7 | 6.3 |

Effect of Moving Interval

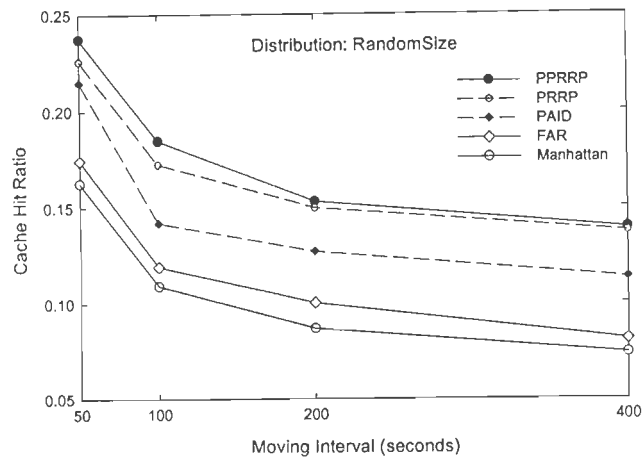
This subsection examines the performance of the replacement policies when the client's moving interval is varied. Longer the moving interval, less frequent are the changes in velocity of the client and, hence, there is lesser randomness in the client's movement. The performance results for IncreasingSize, RandomSize and DecreasingSize of data distribution are shown in Figures 4.9 through 4.12.

We can see that when the moving interval is varied from 50 seconds to 400 seconds, the cache hit ratio decreases drastically. The reason for this is as follows. For a relatively longer moving interval, there is a high possibility of leaving a certain valid region. Consequently, the cached data are less likely to be re-used for subsequent queries, which lead to a worse performance.

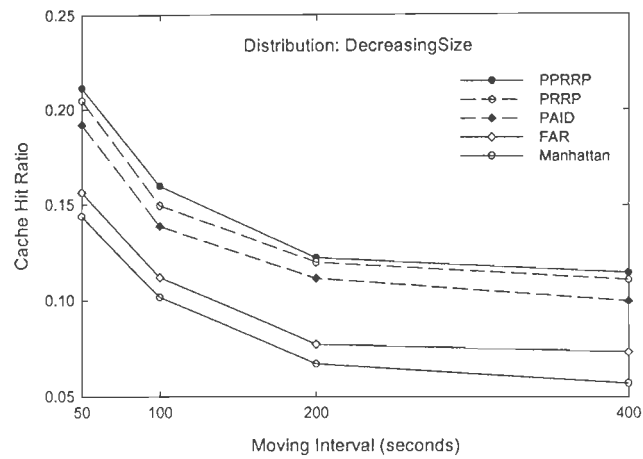
Figures 4.9 and 4.10 compare the performance of cache replacement policies over changing moving interval for Scope Distribution 1. Though for small MI, the randomness in client movement is more as compared to larger MI but PRRP, PRRP and WPRRP perform better than all existing policies for both small and large MI. The predicted region in PRRP helps to keep the data items within the influence of client's movement, thereby reducing the affect of randomness in client's movement.



(a)

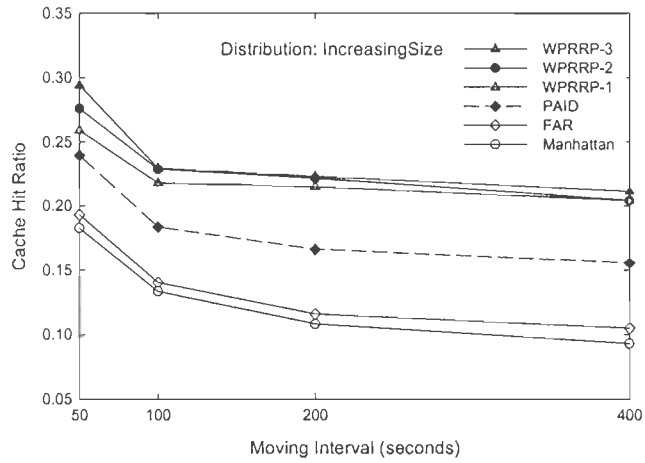


(b)

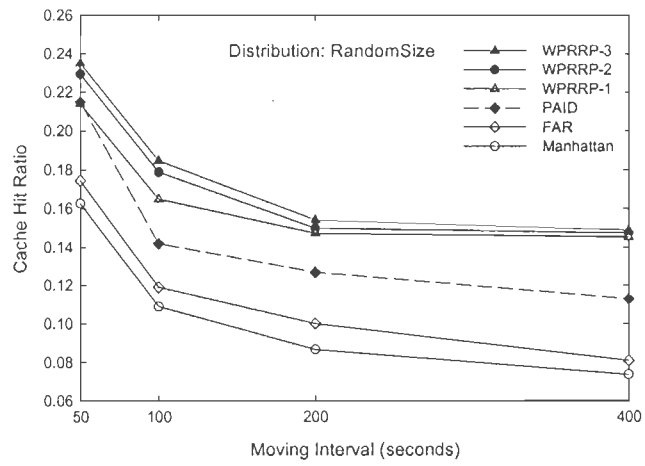


(c)

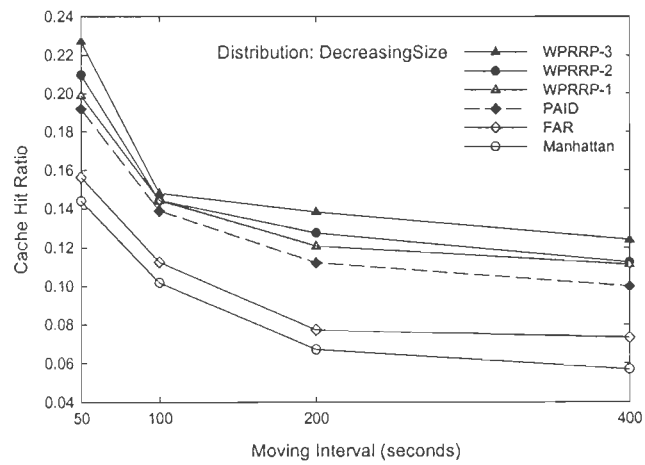
Figure 4.9 Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Moving Interval (Scope Distribution 1)



(a)

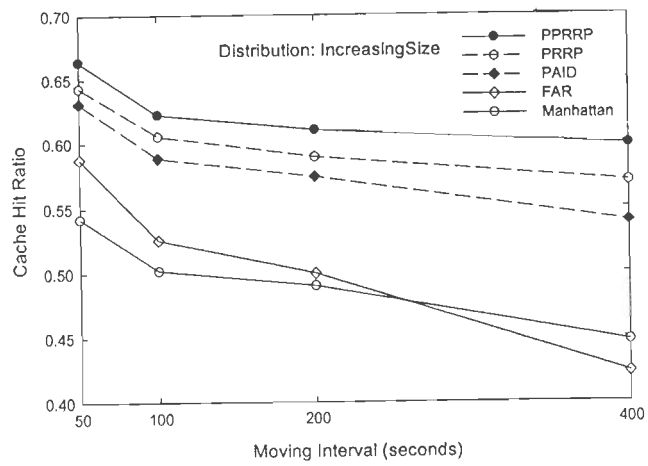


(b)

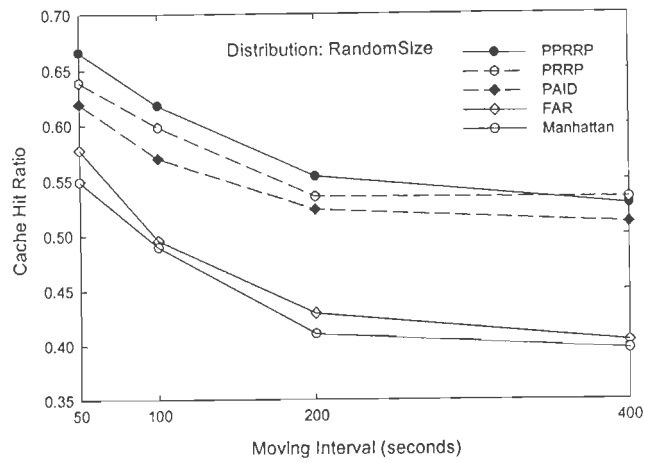


(c)

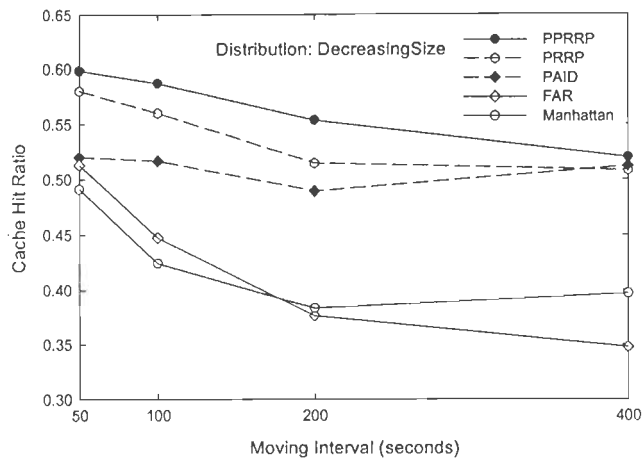
Figure 4.10 Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Moving Interval (Scope Distribution 1)



(a)

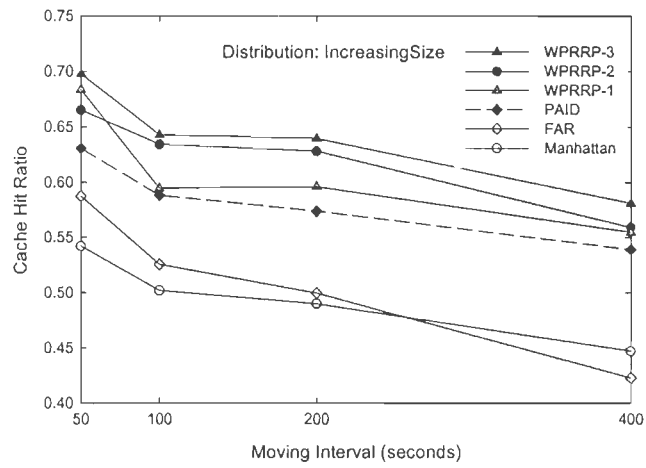


(b)

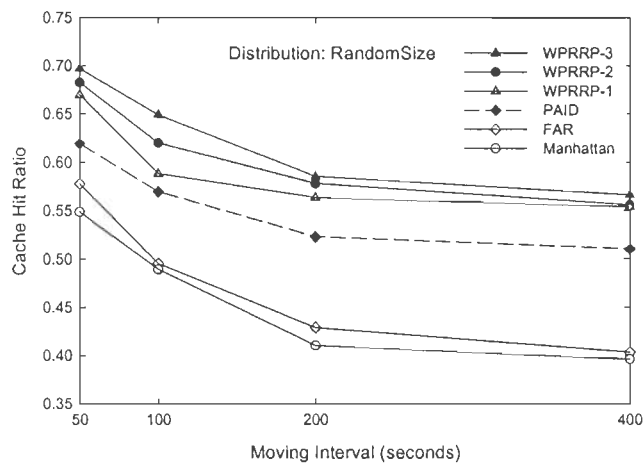


(c)

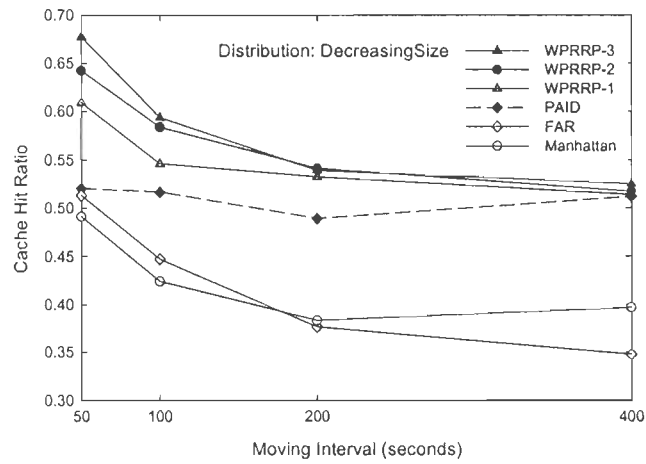
Figure 4.11 Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Moving Interval (Scope Distribution 2)



(a)



(b)



(c)

Figure 4.12 Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Moving Interval (Scope Distribution 2)

Average improvement of PRRP is 22%, 16% and 9% for IncreasingSize, RandomSize and DecreasingSize respectively over the next best policy PAID. Also, average improvement of PRRP over PAID is 27%, 21% and 12% for IncreasingSize, RandomSize and DecreasingSize respectively. Weighted distance with respect to predicted region also plays a major role in overcoming the randomness of the client movement. WPRRP-1, WPRRP-2 and WPRRP-3 also show improvement in performance over PAID along with PRRP and PRRP. As shown in Figure 4.10, WPRRP-3 gives the best performance, that is, performance not only better than PAID but also better than PRRP and PRRP because the average improvement of WPRRP-3 is 30%, 23% and 18% for IncreasingSize, RandomSize and DecreasingSize respectively over the next best policy PAID.

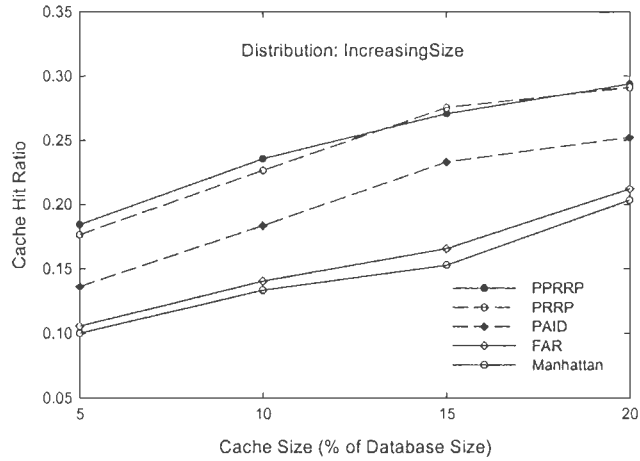
Figures 4.11 and 4.12 compare the performance of cache replacement policies over change in moving interval for Scope Distribution 2. For Scope Distribution 2 also, we get similar improvement in performance of proposed policies as they were for Scope Distribution 1. The average improvement of PRRP, PRRP and WPRRP-3 over PAID is shown in Table 4.4.

Table 4.4 Average Improvement of WPRRP-3, PRRP and PRRP over PAID on Different Moving Intervals (Scope Distribution 2)

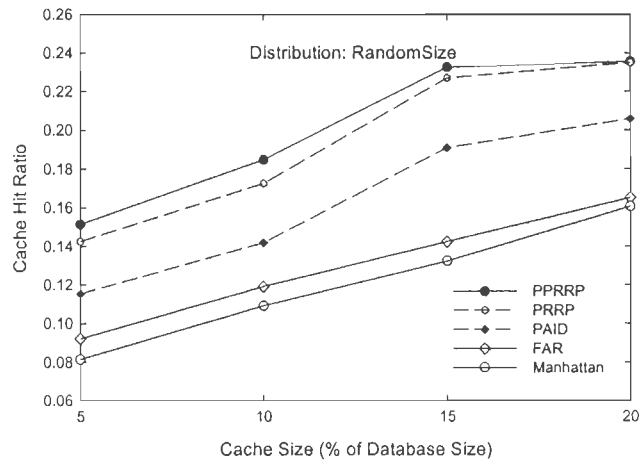
| Proposed Policies | IncreasingSize (%) | RandomSize (%) | DecreasingSize (%) |
|--------------------------|---------------------------|-----------------------|---------------------------|
| WPRRP-3 | 10 | 12.4 | 14.5 |
| PRRP | 7 | 6.3 | 11 |
| PRRP | 3.3 | 3.8 | 8.3 |

Effect of Cache Size

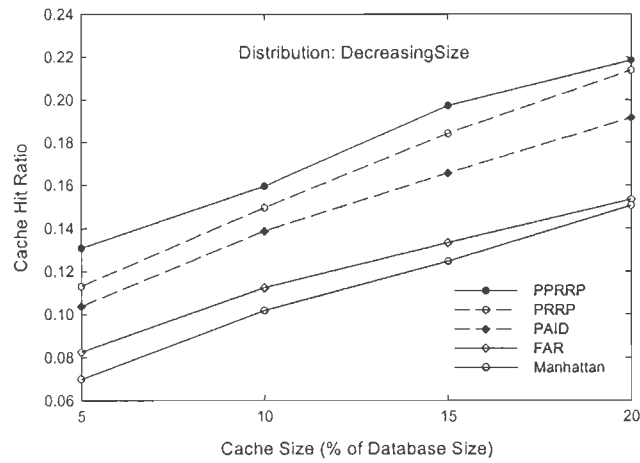
In this set of experiments, we intend to investigate the robustness of the proposed replacement schemes under various cache sizes. Figures 4.13 to 4.16 show the results when CacheSizeRatio is varied from 5% to 20%. As expected, the performance of all replacement schemes improves with increasing cache size. This is because the cache can hold large number of data items which increases the probability of getting a cache hit. Moreover, replacement occurs less frequent in comparison to the case when cache size is low. Figures 4.13 and 4.14 show the performance for Scope Distribution 1. PRRP consistently outperforms the existing policies from small size cache to large size cache. Average improvement of PRRP over PAID is 21%, 20% and 10% for IncreasingSize, RandomSize and DecreasingSize respectively. Similar to PRRP, PRRP and



(a)

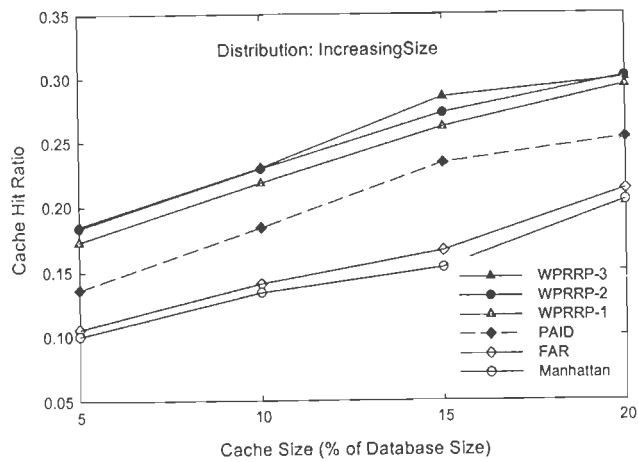


(b)

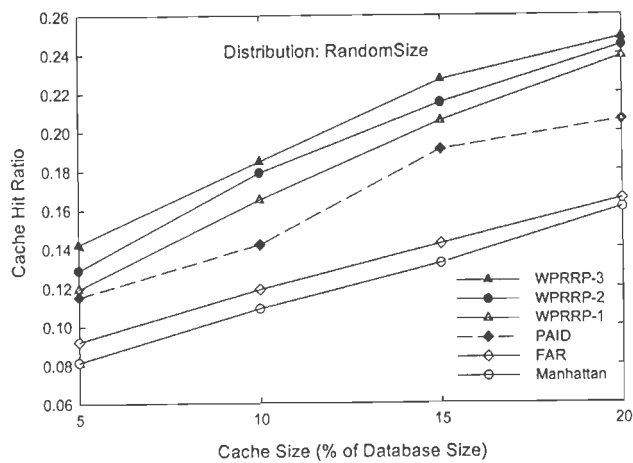


(c)

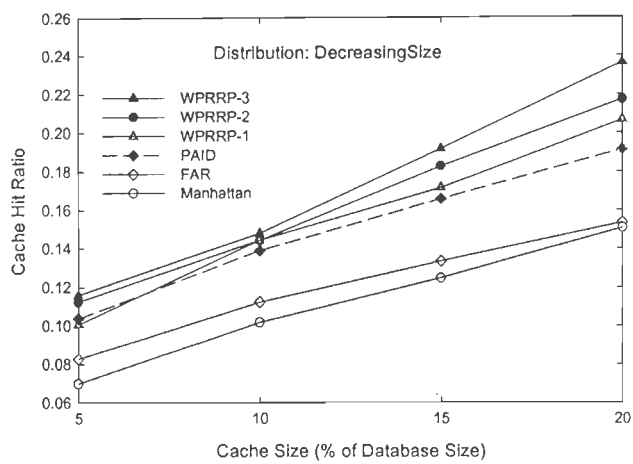
Figure 4.13 Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Cache Size (Scope Distribution 1)



(a)

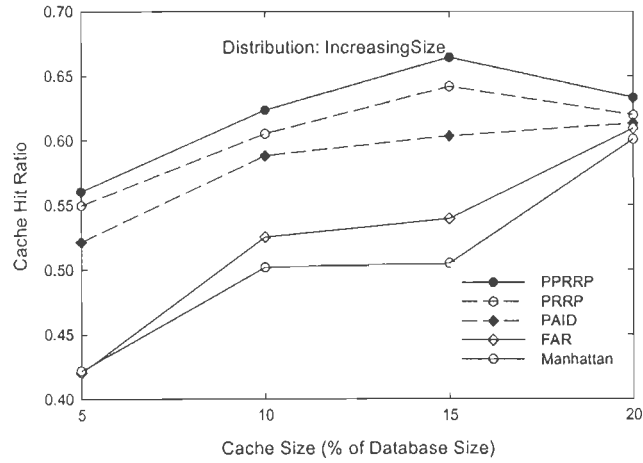


(b)

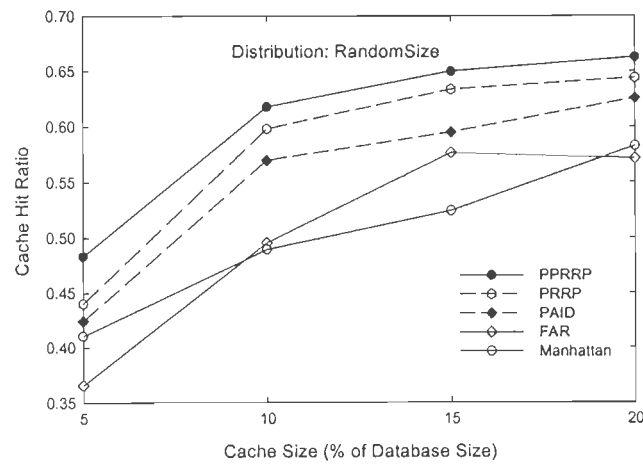


(c)

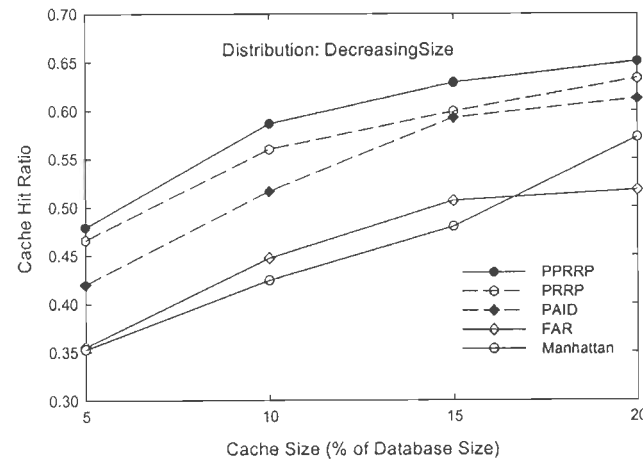
Figure 4.14 Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Cache Size (Scope Distribution 1)



(a)



(b)



(c)

Figure 4.15 Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Cache Size (Scope Distribution 2)

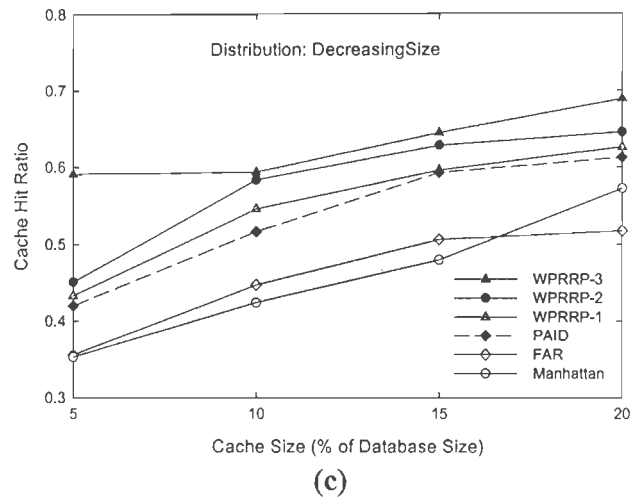
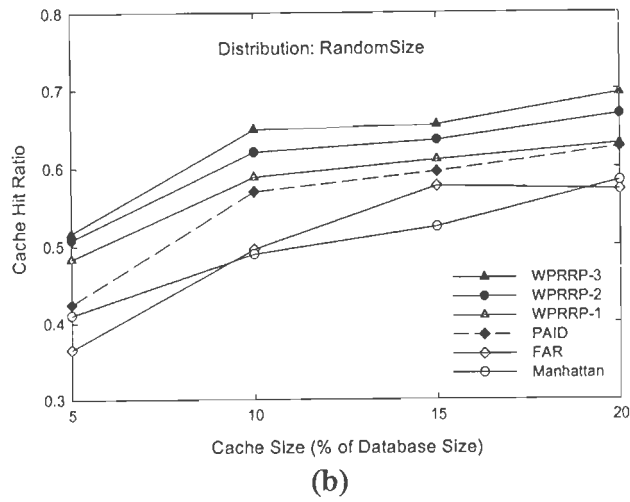
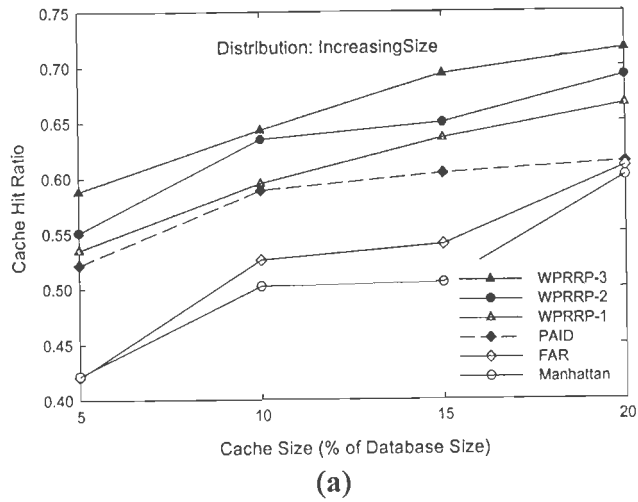


Figure 4.16 Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Cache Size (Scope Distribution 2)

WPRRP policies also consistently outperform the existing policies from small size cache to large size cache. Average improvement of PPRRP over PAID is 25%, 24% and 18% for IncreasingSize, RandomSize and DecreasingSize respectively. Average improvement of WPRRP-3 over PAID is 26%, 23% and 15% for IncreasingSize, RandomSize and DecreasingSize respectively.

Figures 4.15 and 4.16 compare the performance of cache replacement policies under varied CacheSizeRatio for Scope Distribution 2. Results show similar performance gains for all proposed policies for Scope Distribution 2 also. The average improvement of PRRP, PPRRP and WPRRP-3 over PAID is shown in Table 4.5

Table 4.5 Average Improvement of WPRRP-3, PPRRP and PRRP over PAID on Different Cache Size (Scope Distribution 2)

| Proposed Policies | IncreasingSize (%) | RandomSize (%) | DecreasingSize (%) |
|--------------------------|---------------------------|-----------------------|---------------------------|
| WPRRP-3 | 13.4 | 14.2 | 19 |
| PPRRP | 6.7 | 9.4 | 10 |
| PRRP | 3.9 | 4.5 | 6 |

Effect of Client's Speed

This subsection examines the effect of change in client's speed on the performance of the proposed cache replacement policies. Client's cache hit ratio is shown against client speed from Figures 4.17 to 4.20. Four speed ranges [16], 1~5m/s, 6~10m/s, 16~20m/s, 25~35m/s, corresponding to the speed of a walking human, a running human, a vehicle with moderate speed and a vehicle with high speed, respectively are used. It can be seen that very high cache hit ratio can be achieved for walking human. For higher speed range, the cache hit ratio drops as client spends less time at each geographic location and the valid scope of each data item stored in cache becomes less effective. In PRRP, PPRRP, and WPRRP(all cases), higher the speed of client, greater is the predicted region and hence more data items stored in the cache are held in that region. The overall performance of WPRRP-3 is best with respect to other policies in each speed range. Because the weighted distance in WPRRP-3 helps to retain the data items in cache more efficiently than the other proposed policies. Average improvement of PRRP, PPRRP, and WPRRP-3 over PAID for different speed ranges (see Figures 4.17 and 4.18) for Scope Distribution 1 are given in Table 4.6, Table 4.7 and Table 4.8 respectively.

**Table 4.6 Improvement of PRRP over PAID on Different Speed Ranges
(Scope Distribution 1)**

| Speed Ranges (m/s) | IncreasingSize (%) | RandomSize (%) | DecreasingSize (%) |
|-----------------------|-----------------------|-------------------|-----------------------|
| <i>1~5</i> | 39 | 20 | 15 |
| <i>6~10</i> | 37 | 23 | 16 |
| <i>16~20</i> | 34 | 18 | 14 |
| <i>25~35</i> | 26 | 10 | 15 |

**Table 4.7 Improvement of PPRRP over PAID on Different Speed Ranges
(Scope Distribution 1)**

| Speed Ranges (m/s) | IncreasingSize (%) | RandomSize (%) | DecreasingSize (%) |
|-----------------------|-----------------------|-------------------|-----------------------|
| <i>1~5</i> | 43 | 29 | 21 |
| <i>6~10</i> | 41 | 31 | 25 |
| <i>16~20</i> | 40 | 30 | 24 |
| <i>25~35</i> | 37 | 29 | 35 |

**Table 4.8 Improvement of WPRRP-3 over PAID on Different Speed Ranges
(Scope Distribution 1)**

| Speed Ranges (m/s) | IncreasingSize (%) | RandomSize (%) | DecreasingSize (%) |
|-----------------------|-----------------------|-------------------|-----------------------|
| <i>1~5</i> | 47 | 29 | 21 |
| <i>6~10</i> | 43 | 31 | 33 |
| <i>16~20</i> | 40 | 30 | 39 |
| <i>25~35</i> | 41 | 32 | 43 |

**Table 4.9 Improvement of PRRP over PAID on Different Speed Ranges
(Scope Distribution 2)**

| Speed Ranges (m/s) | IncreasingSize (%) | RandomSize (%) | DecreasingSize (%) |
|-----------------------|-----------------------|-------------------|-----------------------|
| <i>1~5</i> | -1.5 | 11.3 | 7.2 |
| <i>6~10</i> | 2.9 | 5 | 8.3 |
| <i>16~20</i> | 2.4 | 4.3 | 3.2 |
| <i>25~35</i> | 1.0 | 2.3 | 7.3 |

Table 4.10 Improvement of PRRP over PAID on Different Speed Ranges (Scope Distribution 2)

| Speed Ranges (m/s) | IncreasingSize (%) | RandomSize (%) | DecreasingSize (%) |
|-----------------------|-----------------------|-------------------|-----------------------|
| <i>1~5</i> | 2.2 | 15.3 | 16.2 |
| <i>6~10</i> | 5.8 | 8.5 | 13.5 |
| <i>16~20</i> | 4.7 | 10.9 | 6.3 |
| <i>25~35</i> | 1.2 | 7.4 | 11.3 |

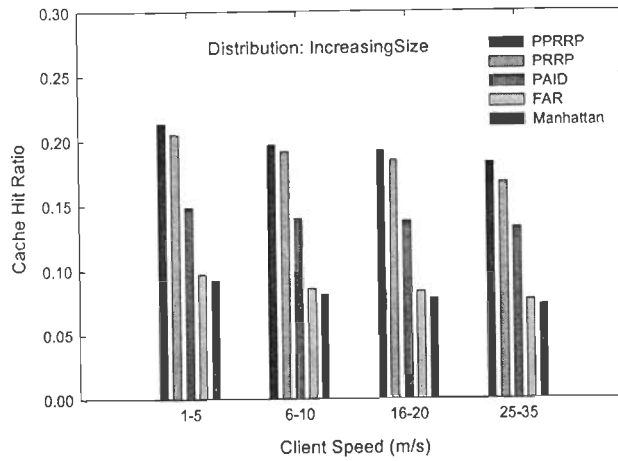
Table 4.11 Improvement of WPRRP-3 over PAID on Different Speed Ranges (Scope Distribution 2)

| Speed Ranges (m/s) | IncreasingSize (%) | RandomSize (%) | DecreasingSize (%) |
|-----------------------|-----------------------|-------------------|-----------------------|
| <i>1~5</i> | 6 | 23.5 | 17.4 |
| <i>6~10</i> | 9.3 | 14 | 14.8 |
| <i>16~20</i> | 11.8 | 12.2 | 10 |
| <i>25~35</i> | 4.1 | 8.2 | 14.1 |

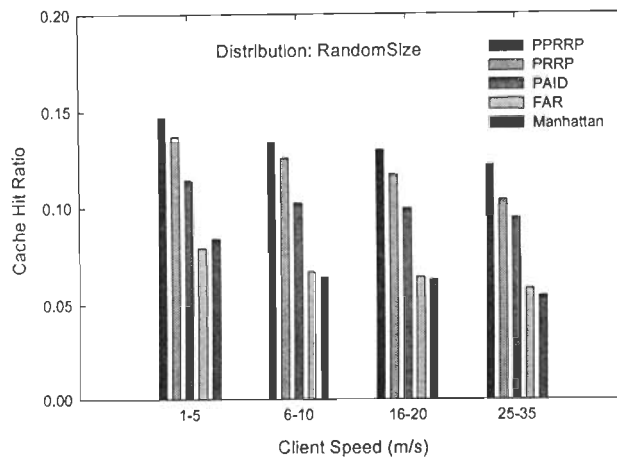
Similarly, Figures 4.19 and 4.20 compare the performance of cache replacement policies under various client's speed for Scope Distribution 2. For Scope Distribution 2 also, the proposed PRRP, PRRP, and WPRRP (all cases) policies have a similar improvement in performance as in case of Scope Distribution 1 and consistently outperform the existing policies. The average improvement of PRRP, PRRP and WPRRP-3 over PAID is shown in Table 4.9, Table 4.10, and Table 4.11 respectively.

Effect of Client's Access pattern

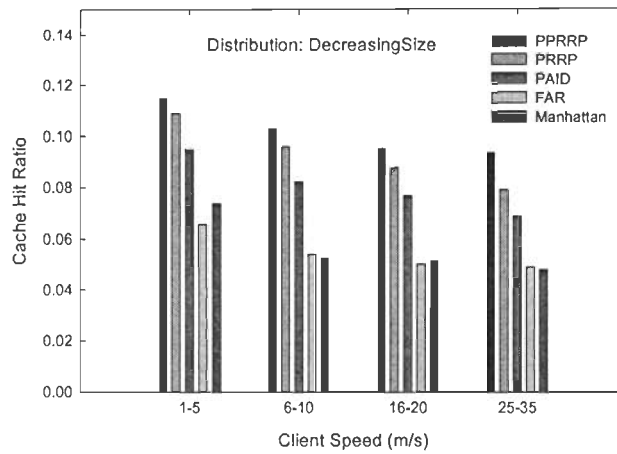
This subsection examines the performance of the replacement policies under various client's access pattern. Client's access pattern is modeled by Zipf's Distribution. The Zipf parameter θ determines the "skewness" of the access pattern over data items. When $\theta=0$, the access pattern is uniformly distributed. When θ increases, more access is focused on few items (skewed). Figures 4.21 to 4.24 show the impact of access pattern on performance of replacement policies for both scope distributions. As desired, performance of PRRP, PRRP, and WPRRP (all cases) along with other replacement policies increases with increase in θ for both Scope distributions over all the three data size distributions. Moreover, proposed policies show an edge over other policies.



(a)

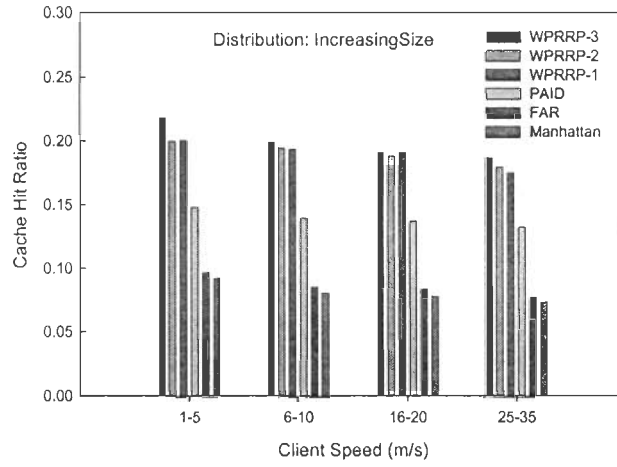


(b)

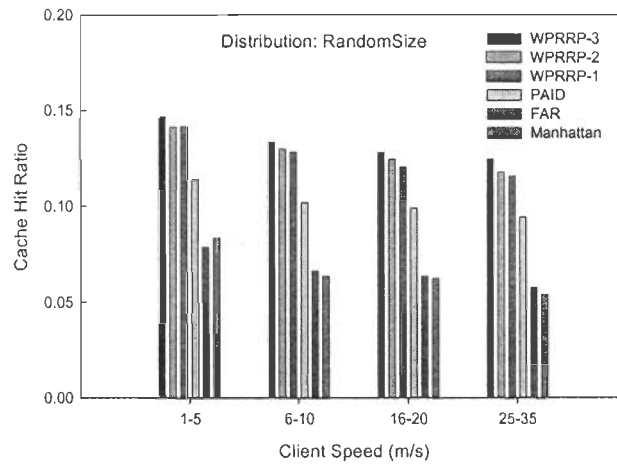


(c)

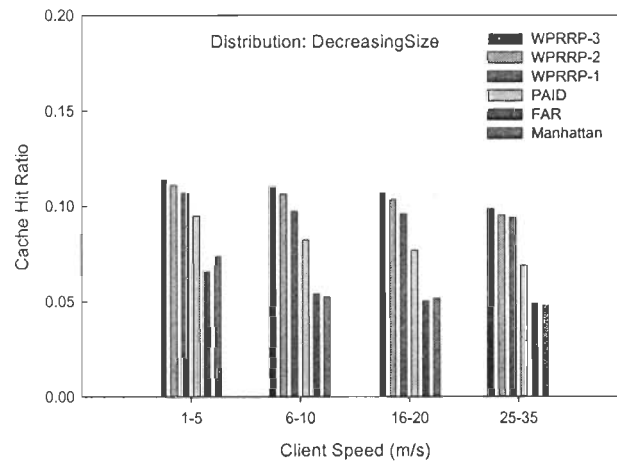
Figure 4.17 Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Client Speed (Scope Distribution 1)



(a)

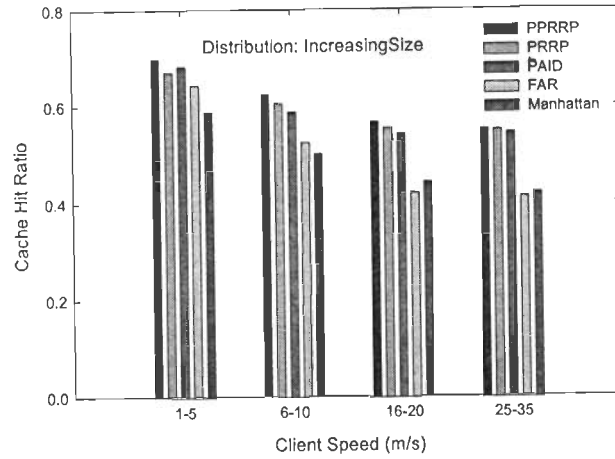


(b)

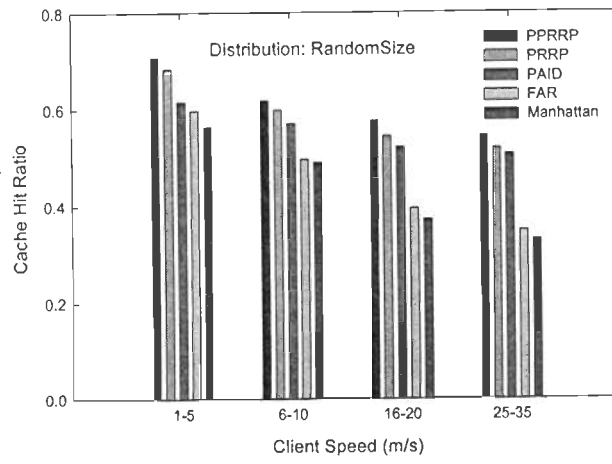


(c)

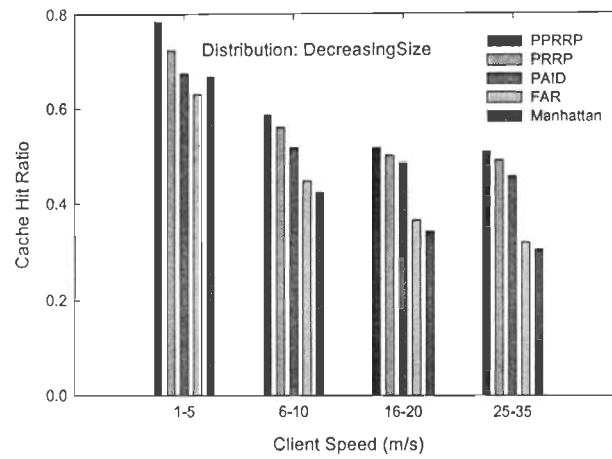
Figure 4.18 Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Client Speed (Scope Distribution 1)



(a)

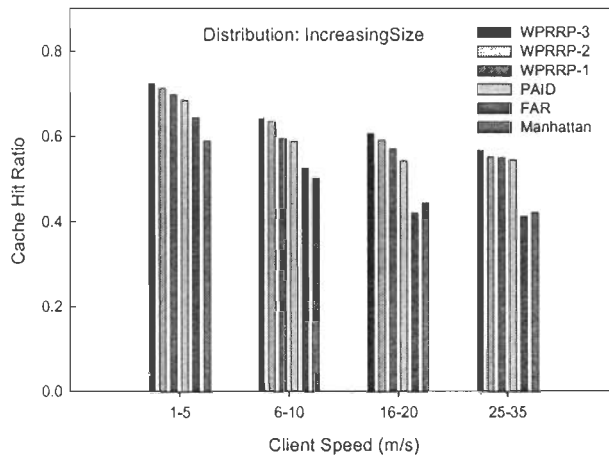


(b)

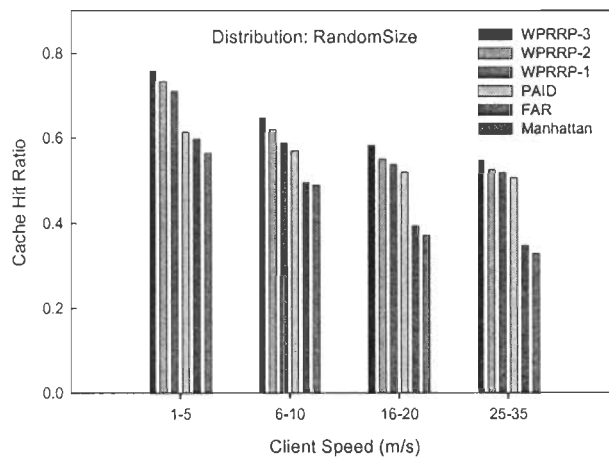


(c)

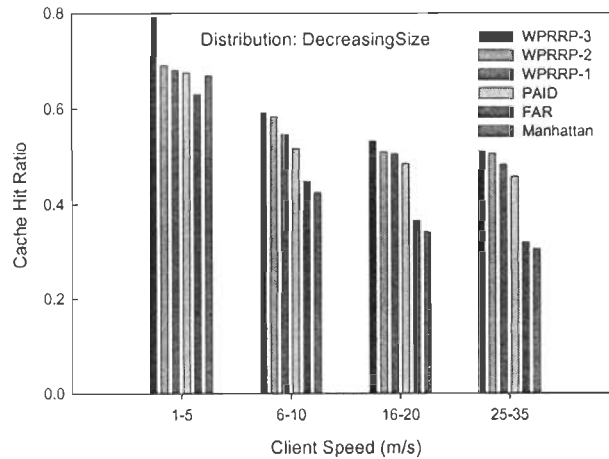
Figure 4.19 Cache Hit Ratio of Replacement Schemes (PRRRP, PRRP) vs. Client Speed (Scope Distribution 2)



(a)

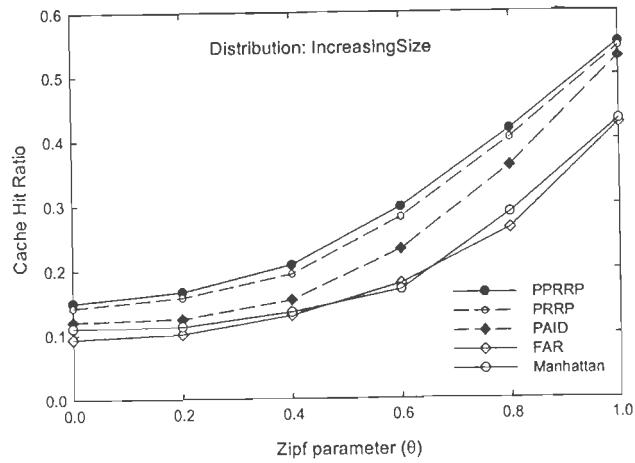


(b)

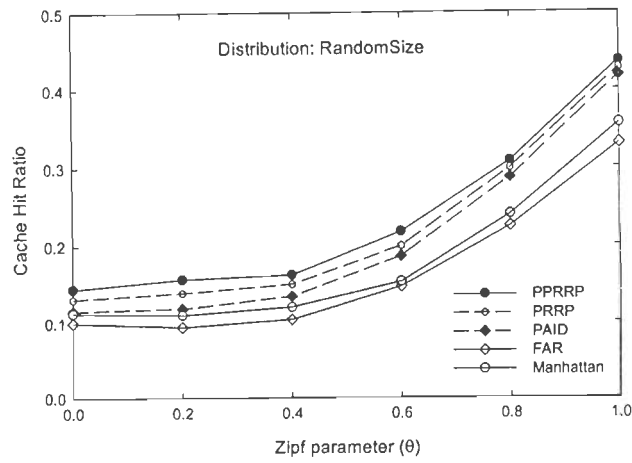


(c)

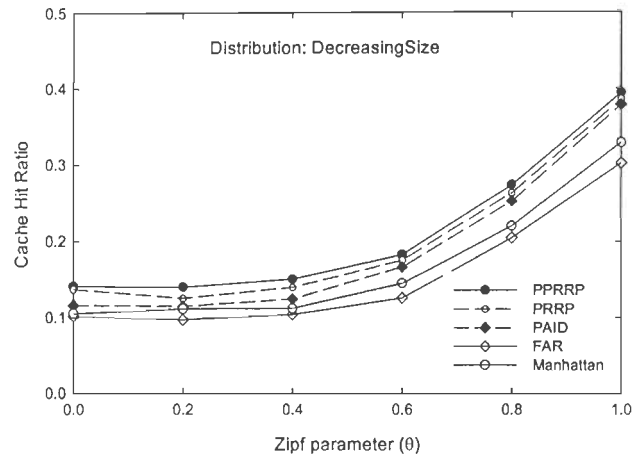
Figure 4.20 Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Client Speed (Scope Distribution 2)



(a)



(b)



(c)

Figure 4.21 Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Zipf Parameter (Scope Distribution 1)

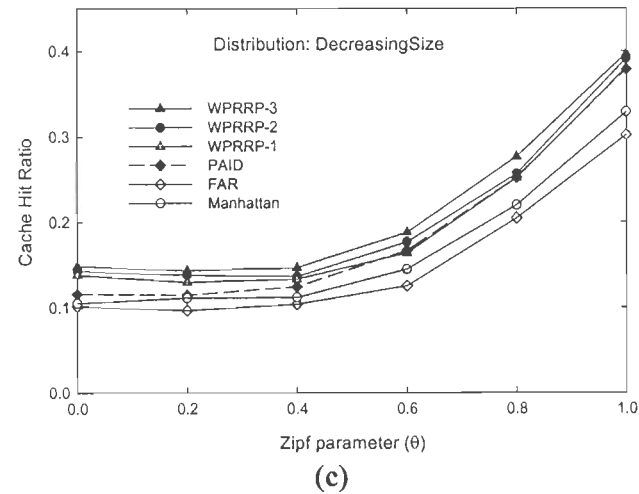
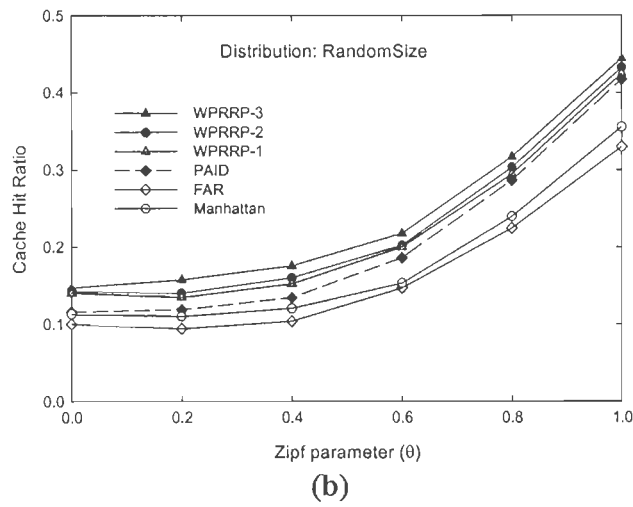
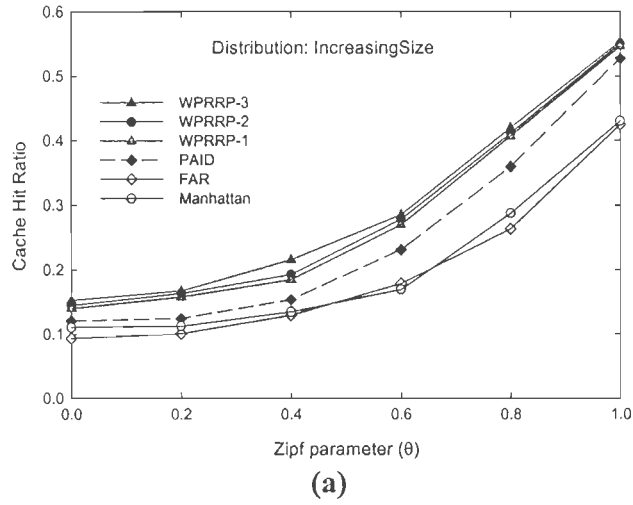
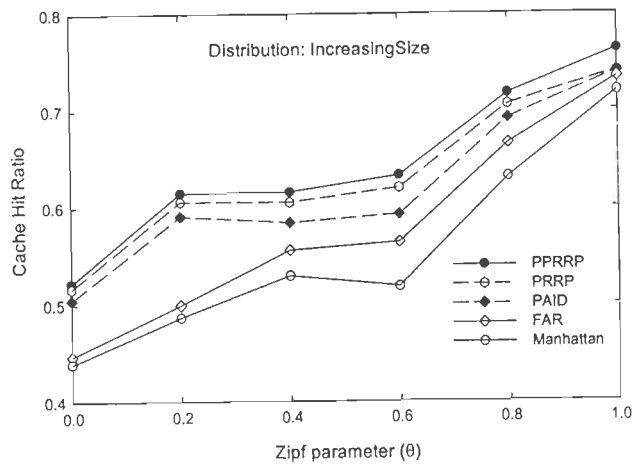
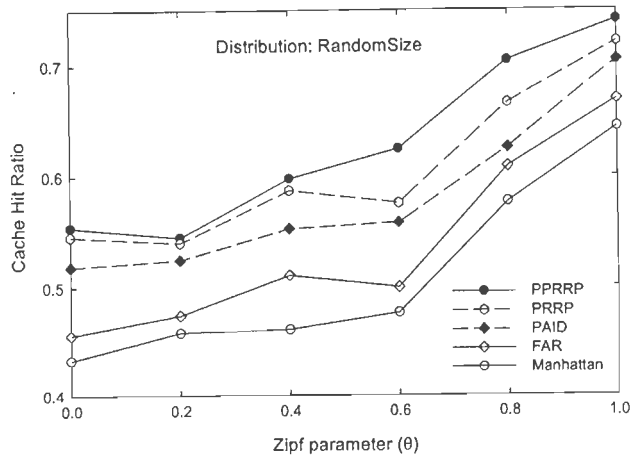


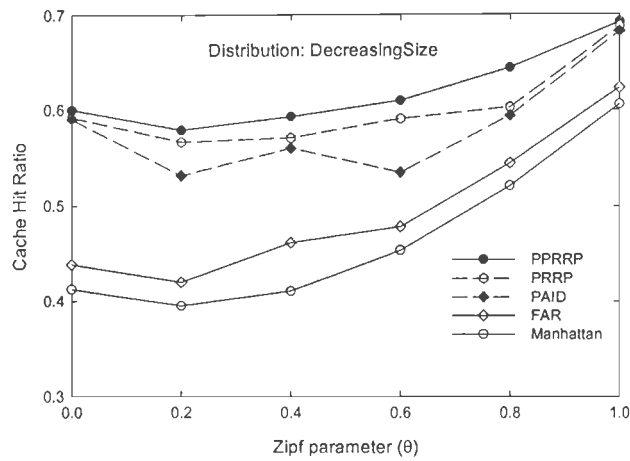
Figure 4.22 Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Zipf Parameter (Scope Distribution 1)



(a)

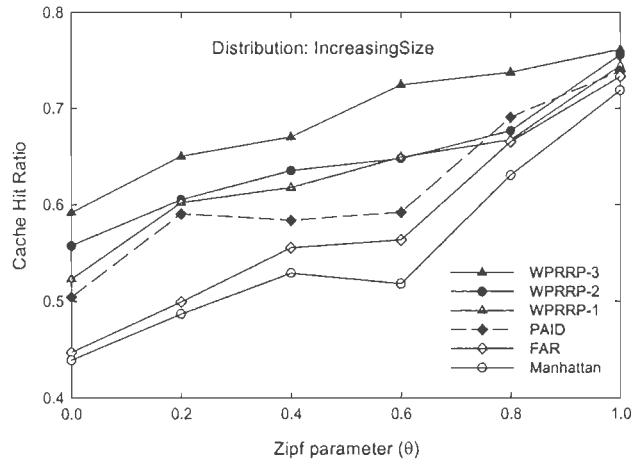


(b)

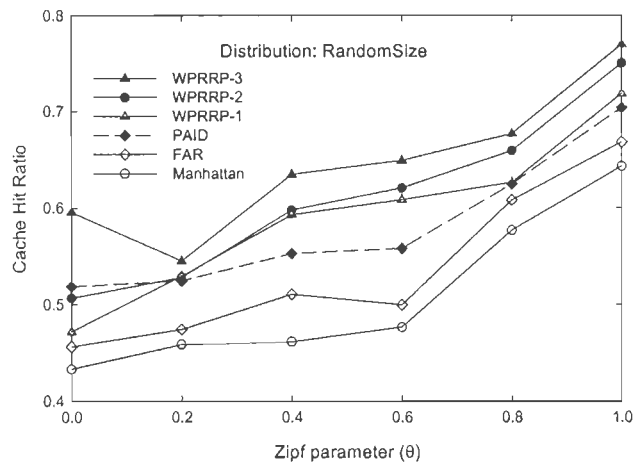


(c)

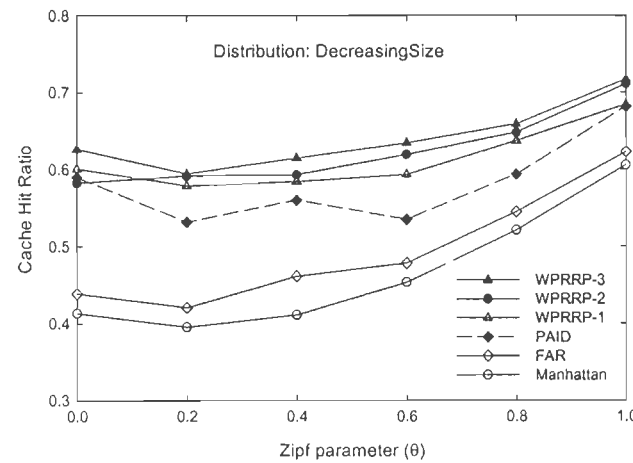
Figure 4.23 Cache Hit Ratio of Replacement Schemes (PPRRP, PRRP) vs. Zipf Parameter (Scope Distribution 2)



(a)



(b)



(c)

Figure 4.24 Cache Hit Ratio of Replacement Schemes (WPRRP) vs. Zipf Parameter (Scope Distribution 2)

4.5 CONCLUSIONS

In this chapter, we proposed the cache replacement policies PRRP, PRRP, and WPRRP that uses predicted region based cost function for selecting data items to be replaced from the cache for location-dependent data. In order to decide which data items to replace from cache, an attempt must be made to predict what items will be accessed in the future. We emphasized on predicting a region around mobile client's current position apart from considering only user's direction or distance. Predicted region plays an important role in improving the system performance. Using the predicted region of user influence, the data items in the vicinity of client's current position are not purged from cache, which increases the cache hit. Proposed policies take into account factors like access probability, data distance from predicted region, valid scope and data size in cache. Out of these factors, the factor data distance from predicted region is unique for PRRP, PRRP, and WPRRP. In PRRP, data distance is calculated such that the data items within the predicted region are given higher priority than the data items outside the predicted region. In PRRP, in addition to giving highest priority to the data items within the predicted region, data items nearer to the client's current position are also favored over other data items in the same predicted region. WPRRP divides the whole areas into different sub regions: in-direction, out-direction, predicted and non-predicted and then associates different weights with each of these sub regions. By changing these weights the schemes can adapt itself to suit to any situation. Section 4.2.3 shows how these schemes can adapt to different situation by using different weight assignments.

A number of simulation experiments have been conducted to evaluate the performance of the proposed cache invalidation schemes. The results show that algorithms WPRRP-3, WPRRP-2, WPRRP-1, PRRP and PRRP, with different system settings, give better performance (improves cache hit ratio) than PAID. Among the proposed policies, the performance of WPRRP-3 was the best. This shows that when client movement is random in nature, predicted region based schemes should be used. Simulation results show that WPRRP-3 achieves an average improvement of more than 25 % for IncreasingSize, more than 20 % for Random Size and more than 15 % for DecreasingSize as compared to existing replacement policy PAID.

Though, the experiments done with different weights assignments for WPRRP show the improvement in performance but a more detailed study is needed to find optimal weight assignment to different regions as it depends heavily on client movement and on other factors.

We compared all policies under various system parameters and for two scope distributions. But in real-world, there can be lots of scope distributions varying from regions to countries. Moreover, we used Euclidean distance to calculate the distance between two points. However, in the real-world, this distance cannot represent the real distance that a user has to cover in order to reach to an object. For example, in City model the distance between two points is calculated using Manhattan distance. Hence, there is a need to explore proposed policies under different real-world conditions. Their performance can vary.

CHAPTER 5

RECENCY/FREQUENCY BASED CACHE REPLACEMENT

5.1 INTRODUCTION

In the last chapter, we focused on the effect of predicted region on location-dependent cache replacement policy. In this chapter, we explore the effects of recency and frequency of data items accessed on location-dependent cache replacement. Least Recently Used (LRU) and Least Frequently Used (LFU) are two most commonly used cache replacement policies based on recency and frequency respectively. We consider these temporal characteristics (i.e., frequency and recency) of data items for location-dependent cache replacement.

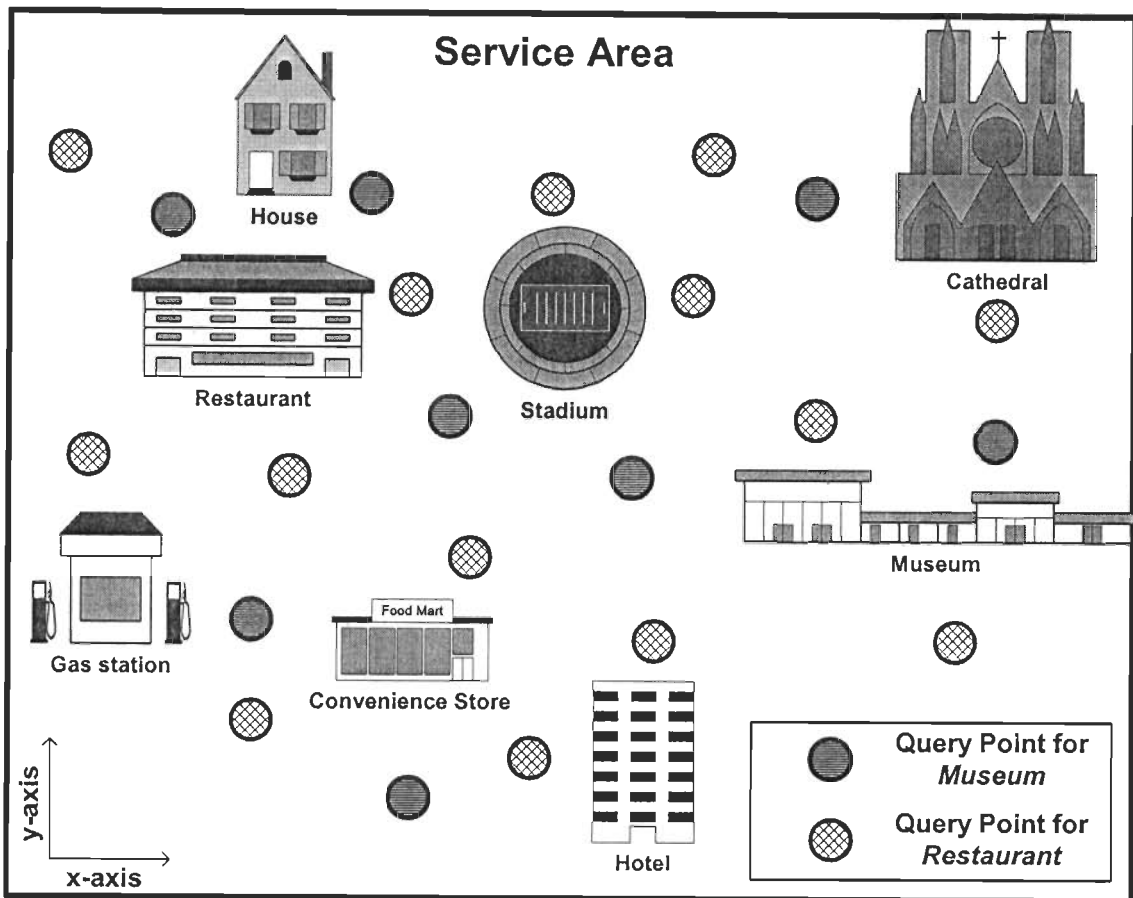


Figure 5.1 Effect of Frequency of Data Item Access

We illustrate our motivation with the help of an example. Consider the scenario shown in Figure 5.1. It shows a snapshot of the nearest-neighbor queries issued for data items- *Museum* and *Restaurant* by a mobile client moving around in a service area during a given time period. The circular symbols reflect access position / query point for *Museum* and *Restaurant* in the service area. This scenario presents the frequency characteristic of data access by the mobile client for data items *Museum* and *Restaurant*. It is observed from the figure that *Restaurant* has been more frequently accessed by the mobile client than *Museum*. This means that the mobile client has more inclination towards *Restaurant* than *Museum*. In other words, the mobile client is more interested in *Restaurant*. Hence, *Restaurant* should be given higher priority over *Museum*, while retaining data items in cache when the cache is full and new data item is to be inserted in the cache.

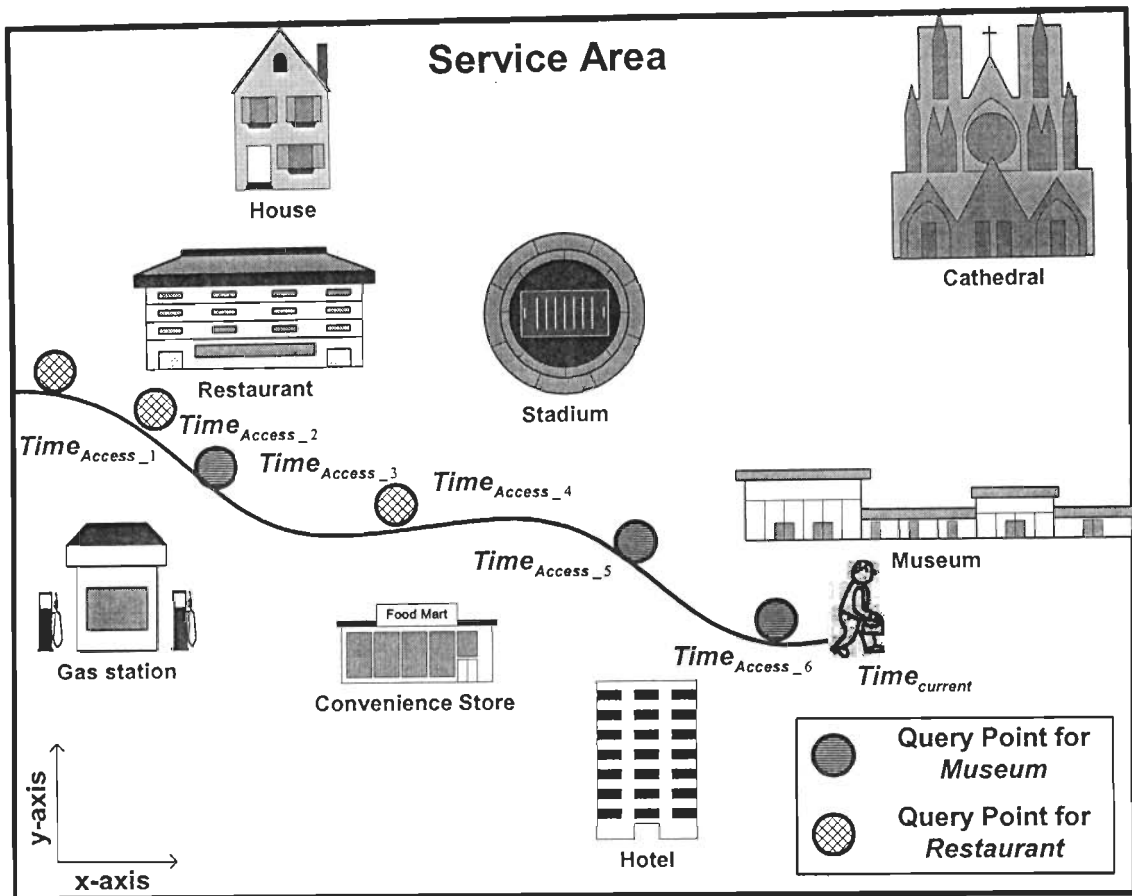


Figure 5.2 Effect of Recency of Data Item Access

Consider the next scenario given in Figure 5.2, which shows the recency characteristic of data access by the mobile client for data items *Museum* and *Restaurant* in the service area. $Time_{Access_i}$ gives the time stamp of user's i^{th} access to data item, where $Time_{Access_1} < Time_{Access_2} < Time_{Access_3} < Time_{Access_4} < Time_{Access_5} < Time_{Access_6} < Time_{current}$. It is observed that *Museum* and *Restaurant* has been accessed same number of times. However, *Museum* has been accessed more recently than *Restaurant* while the mobile client is nearby to the museum in the service area. That is, recency of *Museum* over *Restaurant* is greater with respect to $Time_{current}$. Thus, higher priority should be given to *Museum* over *Restaurant* in retaining data items in client's cache when the cache is full and purging is needed to accommodate a new data item in cache. The recency characteristic shows the current access nature of the user. It may be due to the nearby attractions in which user gets interested or it may be depending upon facilities nearby which matches user's profile.

All earlier approaches for cache replacement of LDD only consider recency as a factor for replacing data items from cache (cache replacement), while we feel that frequency is also important because it shows the user preferences and needs. In this chapter, we propose a replacement policy known as *CRF Area and Inverse Distance Size* (CAIDS), which uses Combined Recency and Frequency value (CRF) [25], valid scope area, data distance and data size of a data item, to select it for replacement. The next section describes our proposed cache replacement policy. The system model used is same as described in section 3.2.

5.2 RECENCY/FREQUENCY BASED CACHE REPLACEMENT POLICY

Efficient and effective cache replacement policies have been the topic of research. Traditional cache replacement policies, due to their temporal nature, consider recency and frequency of data item as the most important factor that affects cache performance. Of these, the Least Recently Used (LRU) and the Least Frequently Used (LFU) replacement policies constitute the two main streams. The LRU policy and its variants base their replacement decision on the recency of references, while the LFU policy and its variants base their decision on the frequency of references. In [25], Lee et al. showed that, between these two seemingly unrelated and independent policies, there exist a spectrum of policies, with LRU and LFU policies as the two extreme points. *Combined Recency and Frequency* value (CRF), proposed by them uses this spectrum and allows a flexible trade-off between recency and frequency of references in making

the replacement decision. The decision to lean toward the recency or frequency is made through the use of a parameter λ , which essentially determines how much more weight we give to the recent history than to the older history. The CRF value of data item i at time t is computed as:

$$C_i(t) = \sum_{j=1}^k F(t - t_{ij}) \quad (5.1)$$

where, $F(x)$ is a weighing function and $\{t_{i1}, t_{i2}, \dots, t_{ik}\}$ are the reference times of data item i and $t_{i1} < t_{i2} < \dots < t_{ik} \leq t$.

The *weighing function* is defined as:

$$F(x) = (1/2)^{\lambda x} \quad (5.2)$$

where, x is the difference between the current time and the time of a reference in the past and λ ranges from 0 to 1.

$F(x)$ essentially reflects the influence of the recency and frequency factors of a data item's past references in projecting the likelihood of its re-reference in the future. In general, $F(x)$ is a monotonically non-increasing function to give more weight to more recent references. This policy differs from the LFU policy in that the contribution of each reference is not always the same, but depends on its recency. The policy also differs from the LRU policy in that it considers not only the most recent reference, but also all the other references in the past.

An intuitive meaning of λ in the weighing function is that a data item's CRF value is reduced to $1/2$ of the original value after every $1/\lambda$ time unit. For example, if λ is 0.0001, a data item's CRF value is reduced to $1/2$ after every 10000 time units. This control parameter λ , allows a trade off between recency and frequency in projecting the likelihood of future references. For example, as λ approaches 0, the CRF value moves towards frequency-based policy. Eventually, when $\lambda=0$ i.e. $F(x)=1$, it becomes simply LFU policy. On the other hand, as λ approaches 1, the CRF value moves towards recency-based policy and when λ is equal to 1 i.e. $F(x) = (1/2)^x$, it degenerates to LRU policy. The spectrum of Recency/Frequency is shown in Figure 5.3.

As far as location-dependent data based cache replacement policies are concerned, policies such as FAR, Manhattan, only considers spatial properties. But PAID considers both spatial and temporal characteristics of data item. In PAID the cost function takes into account the access probabilities (P) of data objects, area of their valid scopes $A(vs)$ and the distance between the client's current position and the valid scope of the object concerned (known as data distance) and

it is given by $P_i A(vs_i)/D(vs_i)$. Temporal properties are taken care by P . But, P as used in PAID only takes into account the recency factor of the data item.

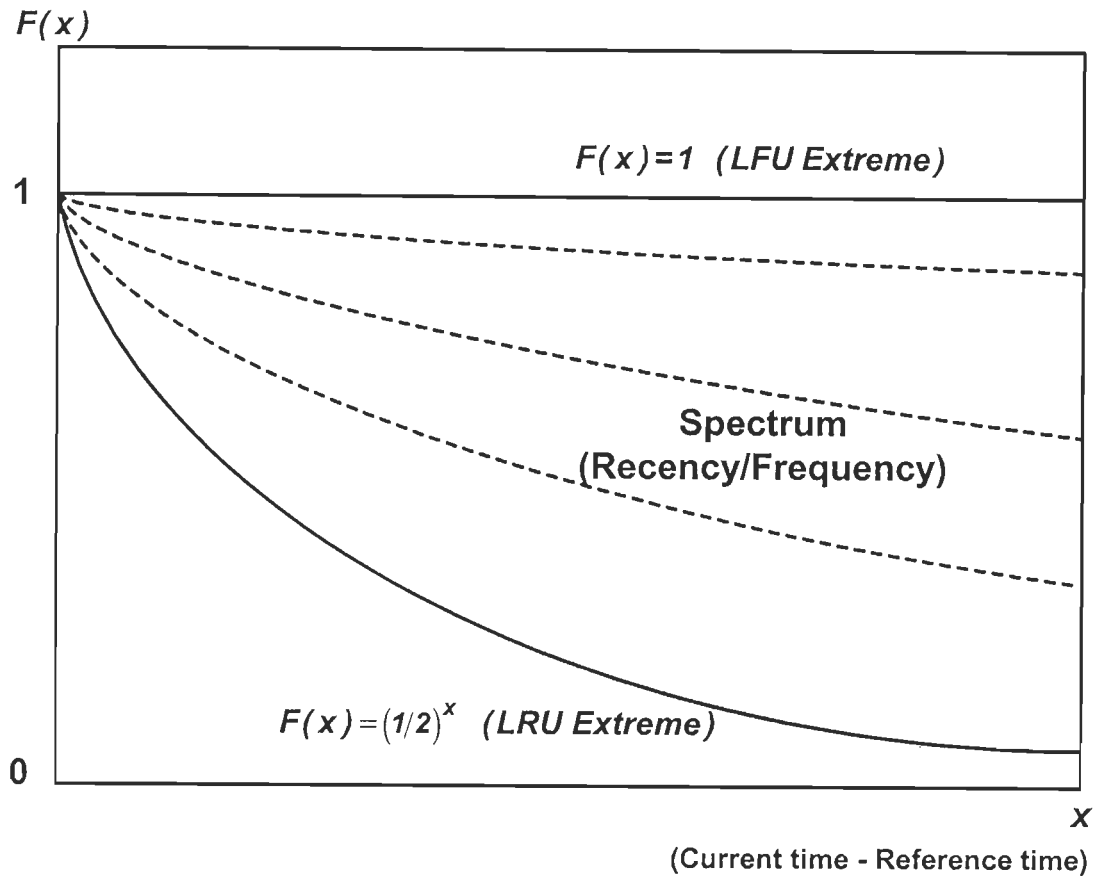


Figure 5.3 Spectrum of Recency/Frequency According to Function $F(x) = (1/2)^x$ where x is (current time – reference time)

We therefore use *Combined Recency and Frequency* (CRF) value, instead of Access Probability P , which quantifies the likelihood of data item that will be referenced in future. Access probability used in PAID for each data item is estimated by using exponential ageing method [14,74]. Two parameters are maintained for each data item i : a running probability $P(i)$ and the time of the last access to item t_i^l . $P(i)$ is initialized to 0. When a new query is issued for data item i , $P(i)$ is updated using the following formula:

$$P_{t_c}(i) = \alpha / (t_c - t_i^l) + (1 - \alpha) P_{t_i^l}(i) \quad (5.3)$$

where, t_c is the current system time and α is a constant factor to weigh the importance of most recent access in the probability estimate.

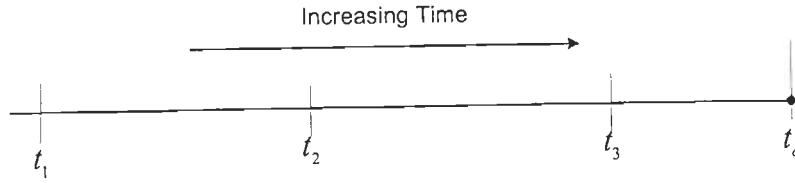


Figure 5.4 Access History of Data Item i

For showing the difference between CRF and P, we take an example. Consider Figure 5.4, which shows the access times t_1, t_2 and t_3 for same data item. Current time is represented by t_c . The CRF and P value at t_c is calculated using eq. (5.1) and eq. (5.3) respectively as shown below:

$$C_{t_c}(i) = F(t_c - t_3) + F(t_c - t_2) + F(t_c - t_1)$$

$$P_{t_c}(i) = \alpha / (t_c - t_3) + (1 - \alpha) P_{t_3}(i)$$

In general, computing the CRF value of a data item requires that the reference times of all the past references of data item to be maintained. It also necessitates re-computing of CRF value of data item at each time step because reference's contribution to CRF values changes over time. From implementation point of view, Lee et al. in [25] proved that if the weighing function $F(x)$ has property $F(x+y) = F(x) * F(y)$, the storage and computational overheads can be reduced drastically such that it becomes not only implementable but also efficient. Authors further gave an equation in which the CRF value at the time of the k^{th} reference can be computed directly from the time of the $(k-1)^{\text{th}}$ reference and the CRF value at that time. Using this property the equation can be reduced to (for derivation please refer [25]):

$$C_{t_k}(i) = F(0) + F(t_k - t_{k-1}) * C_{t_{k-1}}(i) \tag{5.4}$$

Equation 5.4 shows that, at any time, the CRF value can be computed using only two variables t_{k-1} and $C_{t_{k-1}}$ (CRF value at t_{k-1}) for each data item i . This represents history of the data item that needs to be maintained in client's cache.

Our recency/frequency based location-dependent cache replacement policy CAIDS is defined as the product of CRF value of data item, the area of the attached valid scope, inverse of data distance and inverse of data item size. Associated with each cached data object is the

replacement cost. When a new data object needs to be cached and there is insufficient cache space, the object(s) with lowest replacement cost is removed until there is enough space to cache new object. The cost of replacing data value j of data item i is calculated for CAIDS as:

$$CAIDS_Cost_{ij} = \frac{C_i \cdot A(vs_{ij})}{D(vs_{ij}) \cdot S_{ij}} \quad (5.5)$$

where,

C_i : CRF value of data item i ,

$A(vs_{ij})$: area of valid scope vs_{ij} for data value j of data item i ,

$D(vs_{ij})$: distance between the current location of client and the valid scope vs_{ij} , and

S_{ij} : storage space (size) needed to store data value j and its valid scope vs_{ij} .

5.3 PERFORMANCE EVALUATION

The simulation model used to evaluate the performance of the proposed location-dependent cache invalidation policy CAIDS is same as described in section 4.3. This section describes the performance parameters and measures used for simulation. It also analyzes the simulation results of CAIDS.

5.3.1 Performance Parameters

Performance parameters are same as described in section 4.4.1 with following additional detail.

When a new query is issued for data item i , CRF value for each data item is estimated by using eq. (5.4). Two parameters are maintained for each data item i : a CRF value C_i and the time of the last access to item t'_i . C_i is initialized to 0. Note that the CRF value is maintained for each data item rather than for each data value. If the database size is small, the client can maintain the history parameters (i.e., C_i and t'_i for each item i) for all items in its local cache. However, if the database size is large, history information will occupy a significant amount of cache space. To alleviate this problem, we set an upper bound to the amount of cache used for storing it (5 percent of the total cache size in our simulation) and use the LFU policy to manage the limited space reserved for it.

The default values of different parameters used in the simulation experiments are same as given in Table 4.2. Control parameter λ is an additional parameter required for this simulation apart from those mentioned in Table 4.2. The default setting of λ is 0.0001.

5.3.2 Performance Metric

Same as described in section 3.6.2.

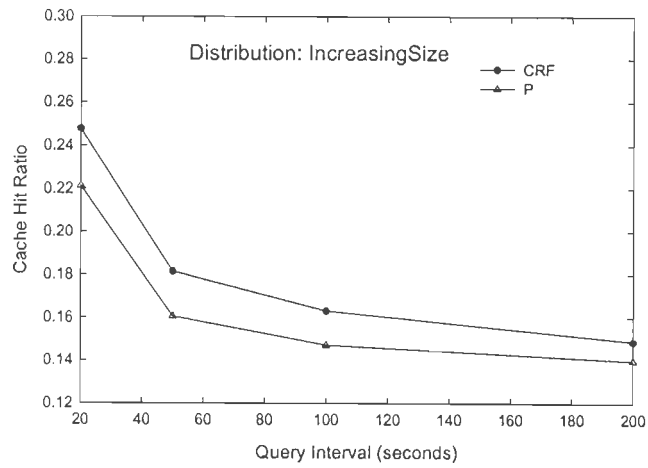
5.3.3 Comparison of Location-Dependent Cache Replacement Schemes

This subsection compares the performance of proposed recency/frequency based location-dependent cache replacement policy CAIDS with PAID. Figures 5.5 to 5.15, show the cache hit ratio for both scope distributions (see Figure 3.8) for various query intervals, moving intervals, cache sizes, client's speed and Zipf's distribution.

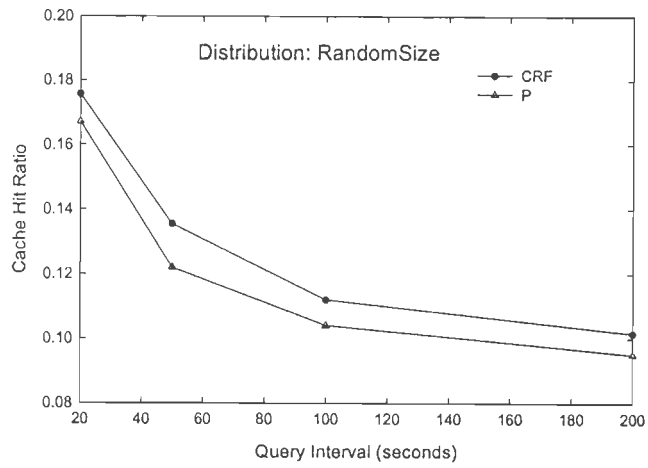
Effect of access probability P and CRF value is shown separately in Figure 5.5 with change in QueryInterval. CRF make best use of the recency/frequency spectrum and gives an average improvement of 10 percent for IncreasingSize, 8 percent for RandomSize and 10 percent for DecreasingSize over access probability. This improvement plays an important role in CAIDS cache replacement policy.

Figure 5.6, shows the improved performance of CAIDS and PAID with respect to change in mean Query Interval for Scope Distribution 1. We observe that the performance of CAIDS is better one. As the query interval increases, cache hit ratio decreases, because the client would make more movements between two successive queries, and thus has low probability at the time of new query to remain in the same valid scope from where the earlier query was being issued. CAIDS gives an average improvement of 33 percent, 28 percent and 17 percent over PAID for IncreasingSize, RandomSize and DecreasingSize respectively.

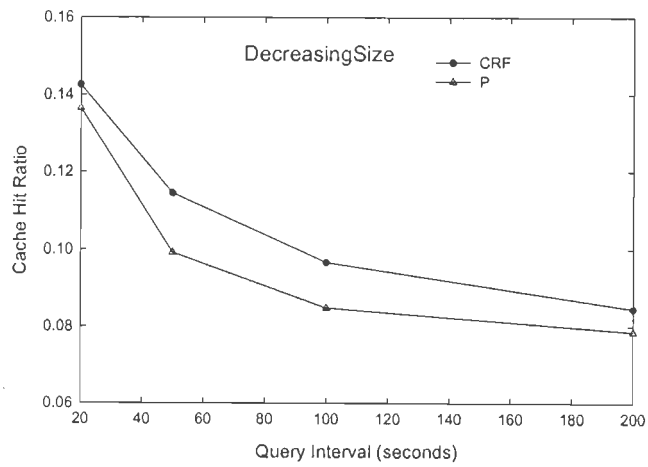
Figure 5.7, shows the effect of Moving Interval (varied from 50 seconds to 400 seconds) on replacement policies for Scope Distribution 1. The longer the moving interval, the less frequently the client changes velocity and direction and, hence, less random is the client's movement. For small MI, the randomness in client movement is more as compared to larger MI. In CAIDS, the frequency factor plays an important role along with the data size for both large and small MI which has not been taken into account by PAID policy. Due to this reason CAIDS performs better than PAID for both small and large MI. As MI increases, the performance decreases.



(a)



(b)



(c)

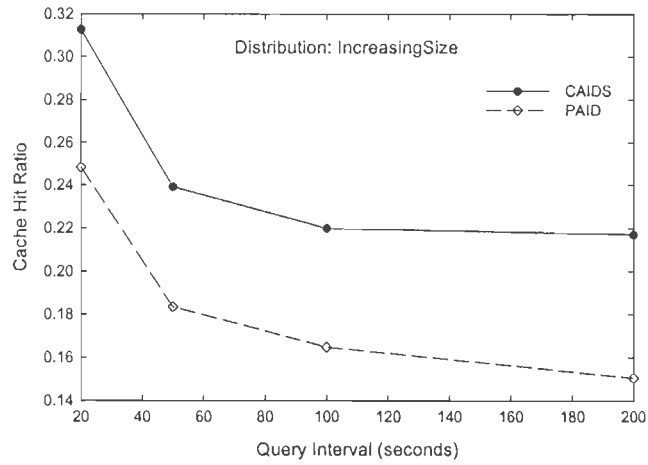
Figure 5.5 Effects of Probability of Access P and CRF on Cache Hit Ratio (Scope Distribution 1)

Because for relatively longer MI, a larger average distance difference is observed for two successive queries, which implies that client has a higher possibility of leaving certain regions. Consequently, the cached data are less likely to be reused for subsequent queries, which lead to a worse performance. However, it appears that this fact deteriorates the performance of PAID more than CAIDS and therefore, CAIDS gives an average improvement of 30 percent, 25 percent and 16 percent over PAID for IncreasingSize, RandomSize and DecreasingSize respectively.

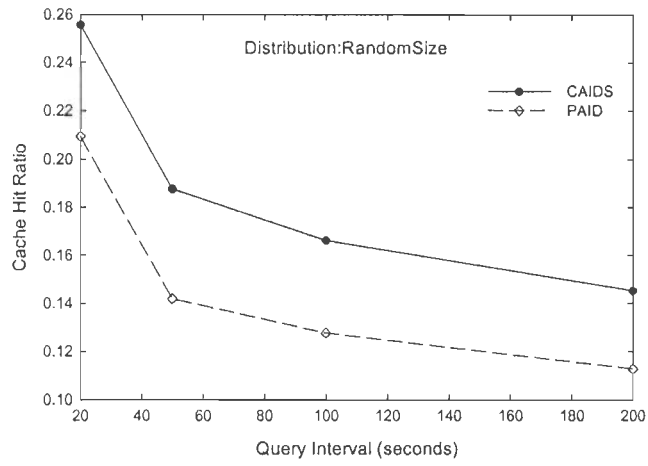
Effect of cache size on performance of replacement policies are shown in Figure 5.8. As expected, the performance of replacement policies improves with increase in cache size. CAIDS consistently out performs PAID policy because CAIDS along with recency and frequency also considers the data item size into its cost function. For IncreasingSize, CAIDS policy gives 30 percent better for small cache sizes and 20 percent better than PAID for large cache sizes. Improvement is 25 percent and 20 percent for small and large cache sizes respectively of CAIDS over PAID for RandomSize. CAIDS gives an average improvement of 16 percent over PAID for DecreasingSize.

Client cache hit ratio is shown against client speed in Figure 5.9. Four speed ranges, 1~5m/s, 6~10m/s, 16~20m/s, 25~35m/s, corresponding to the speed of a walking human, a running human, a vehicle with moderate speed and a vehicle with high speed, respectively are used [16]. It can be seen that very high cache hit ratio can be achieved for walking human. For higher speed range, the cache hit ratio drops as clients spend less time at each geographic location and the valid scope of each data item stored in cache becomes less effective. CAIDS, which interpolates between recency and frequency helps to retain the best suited data item in cache and yields an improvement of over 40 percent for IncreasingSize over 30 percent each for RandomSize and DecreasingSize for all speed ranges over PAID policy.

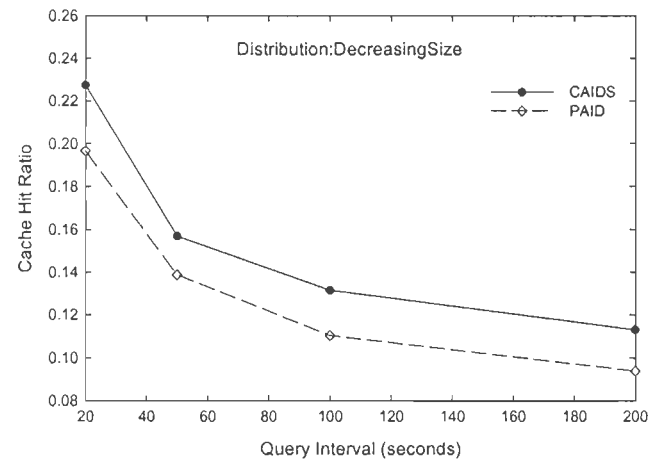
The Zipf parameter θ determines the “skewness” of the access pattern over data items. When $\theta=0$, the access pattern is uniformly distributed. When θ increases, more access is focused on few items (skewed). Figure 5.10 shows the impact of access pattern on performance of replacement policies. CAIDS shows an improved performance over PAID though both increases with increase in θ for all three distributions. Similar performance improvement is observed for Scope Distribution 2 (see Figure 5.15).



(a)

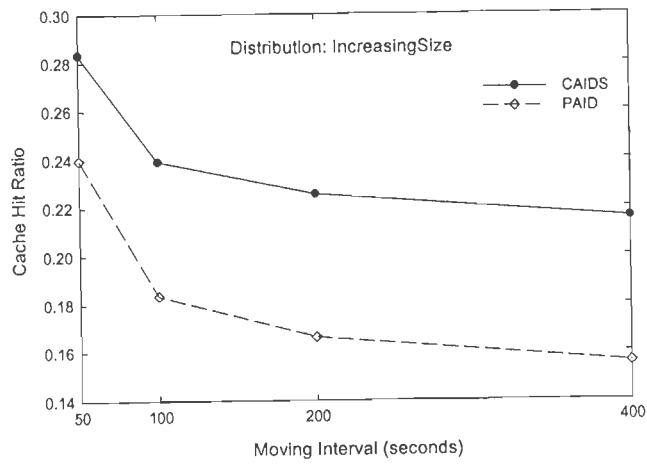


(b)

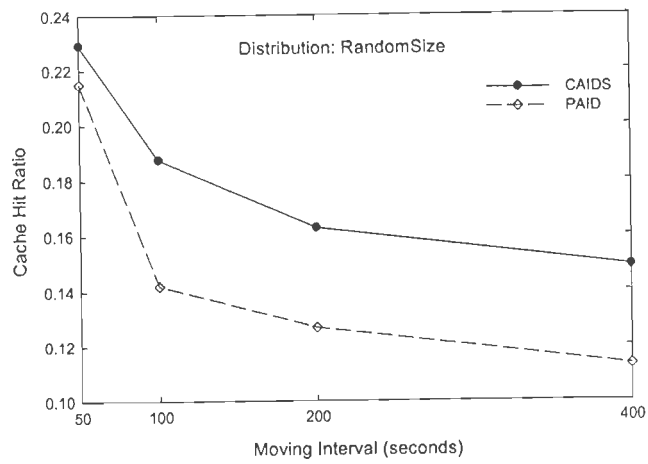


(c)

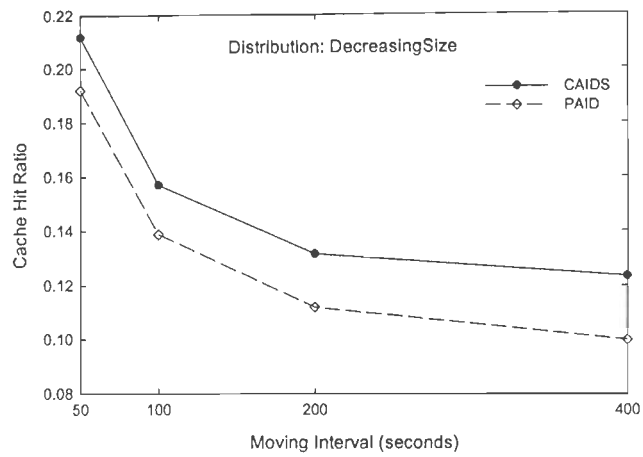
Figure 5.6 Cache Hit Ratio vs Query Interval (Scope Distribution 1)



(a)

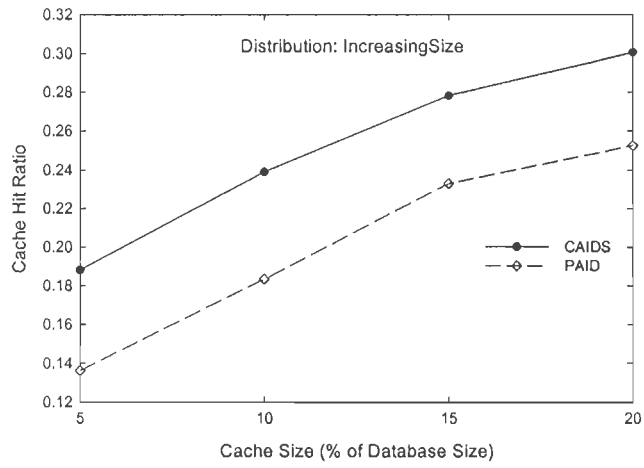


(b)

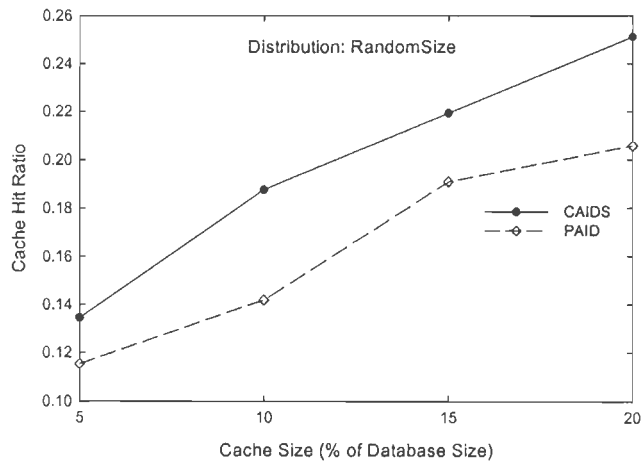


(c)

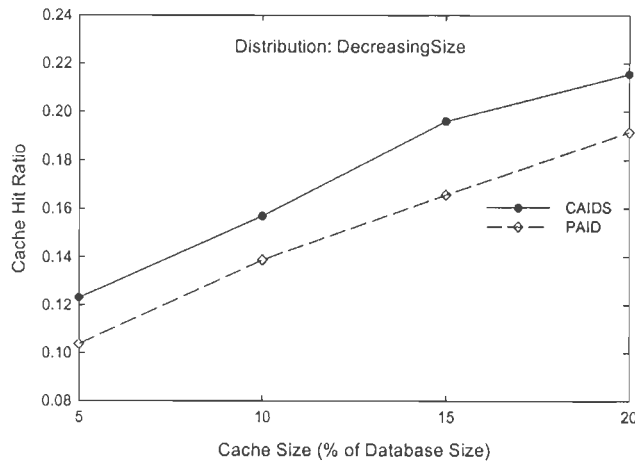
Figure 5.7 Cache Hit Ratio vs Moving Interval (Scope Distribution 1)



(a)

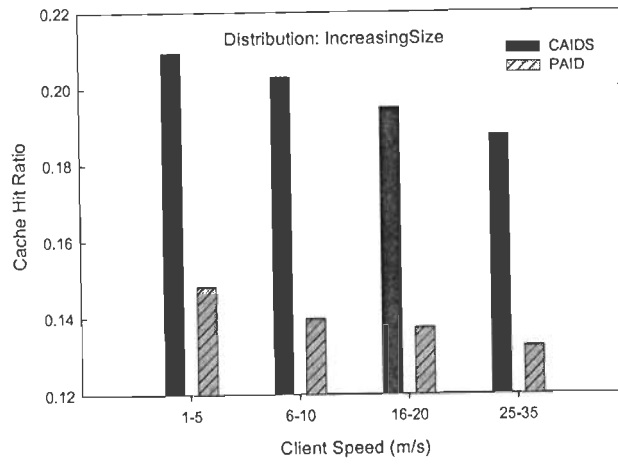


(b)

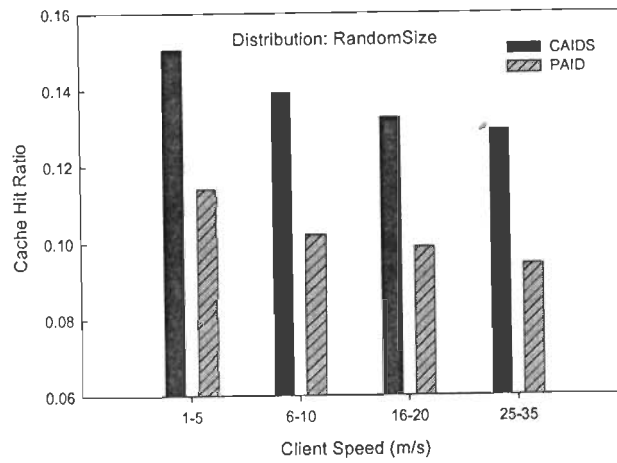


(c)

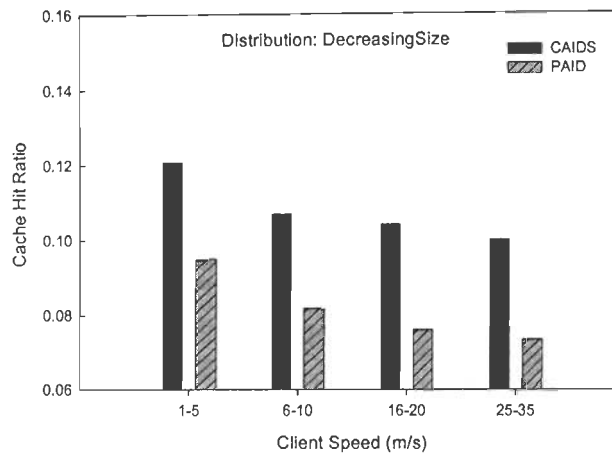
Figure 5.8 Cache Hit Ratio vs Cache Size (Scope Distribution 1)



(a)



(b)



(c)

Figure 5.9 Cache Hit Ratio vs Client Speed (Scope Distribution 1)

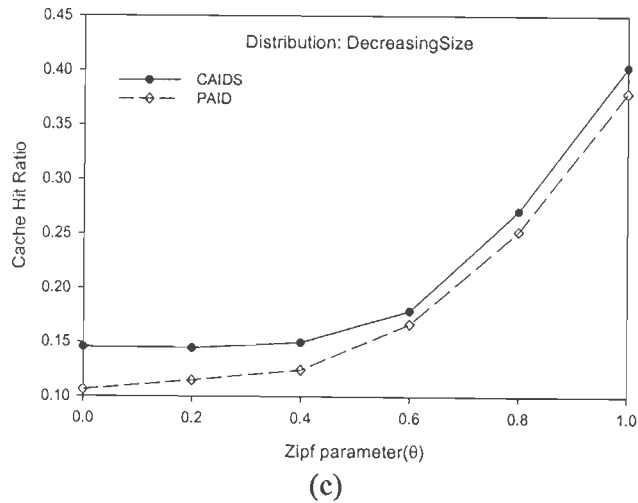
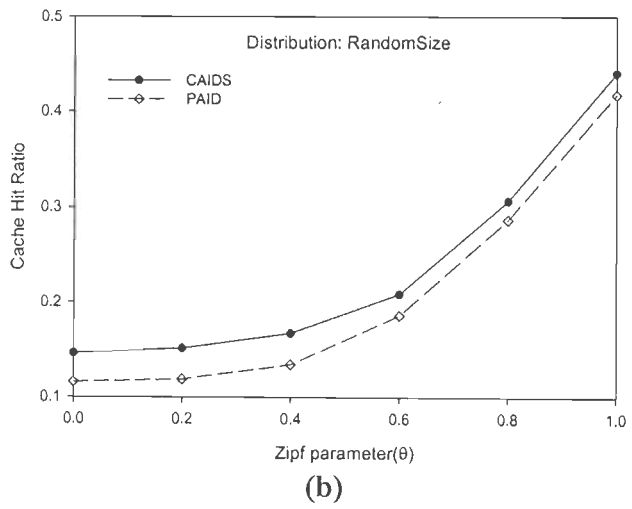
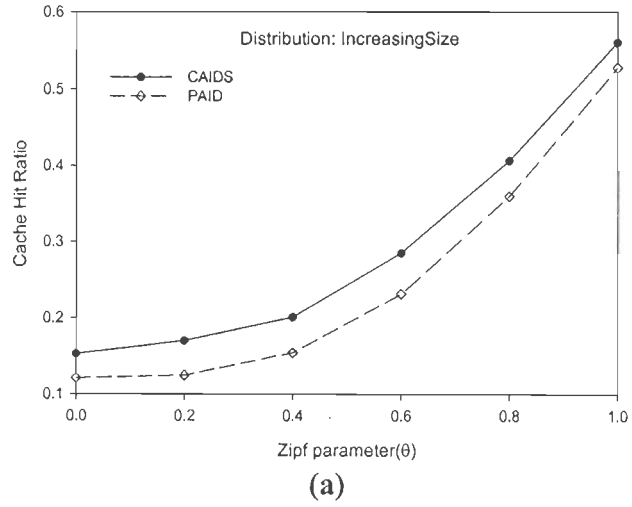
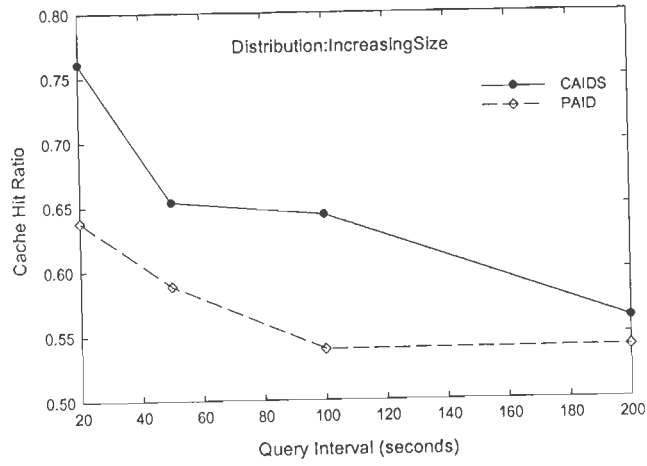
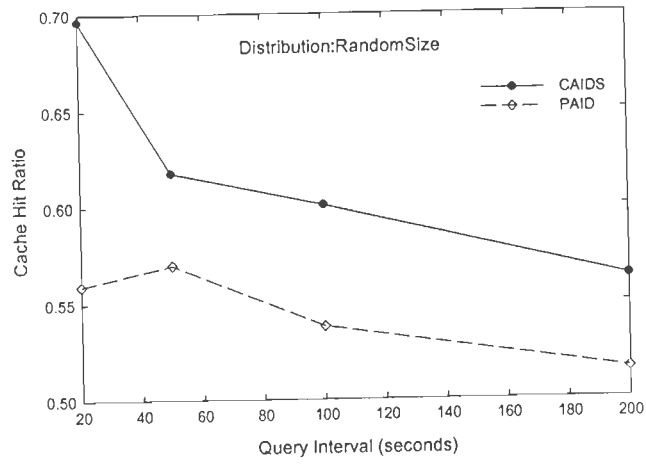


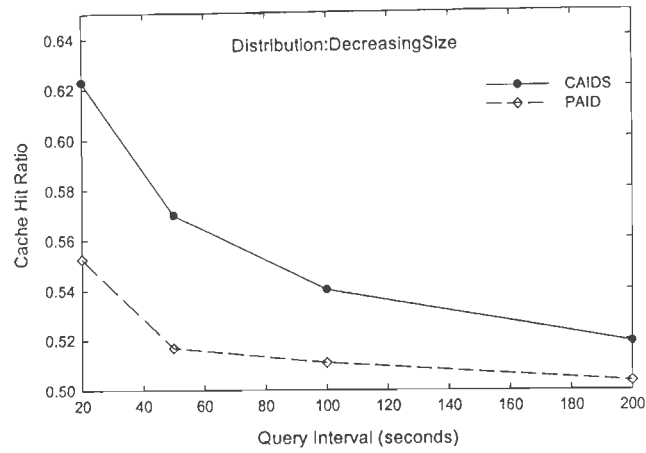
Figure 5.10 Cache Hit Ratio vs Zipf parameter(θ) (Scope Distribution 1)



(a)

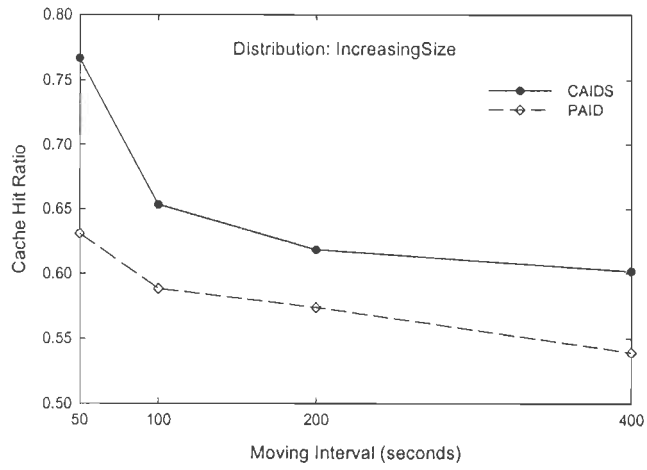


(b)

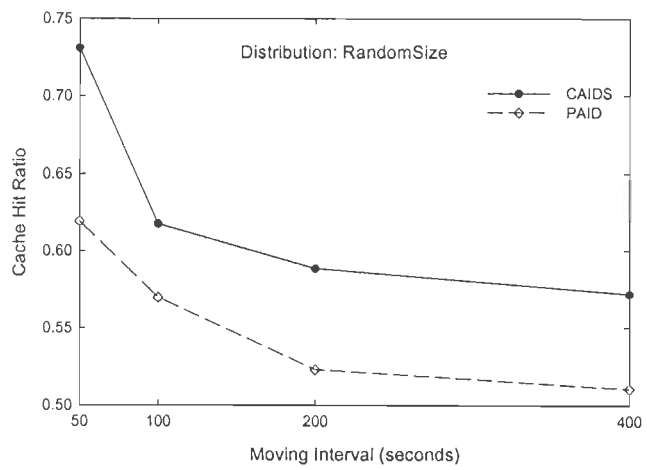


(c)

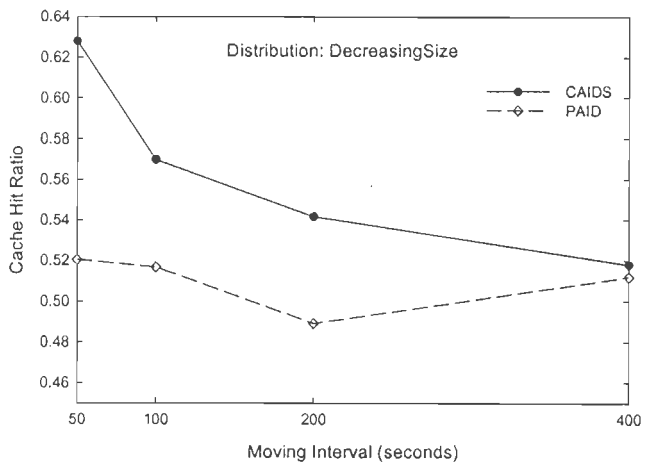
Figure 5.11 Cache Hit Ratio vs Query Interval (Scope Distribution 2)



(a)

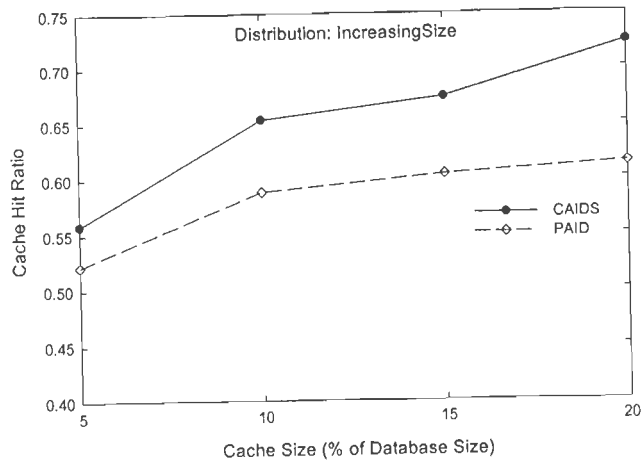


(b)

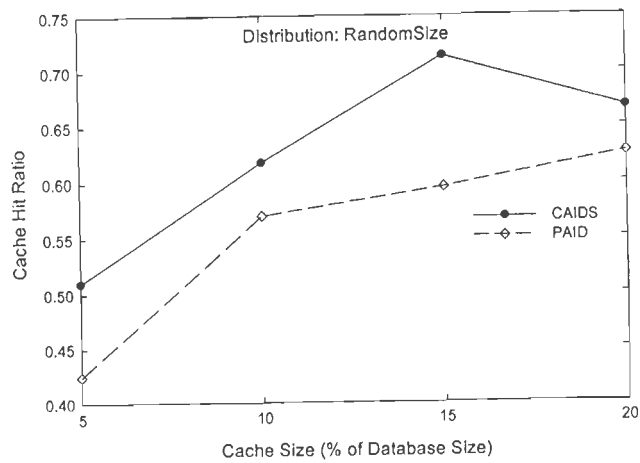


(c)

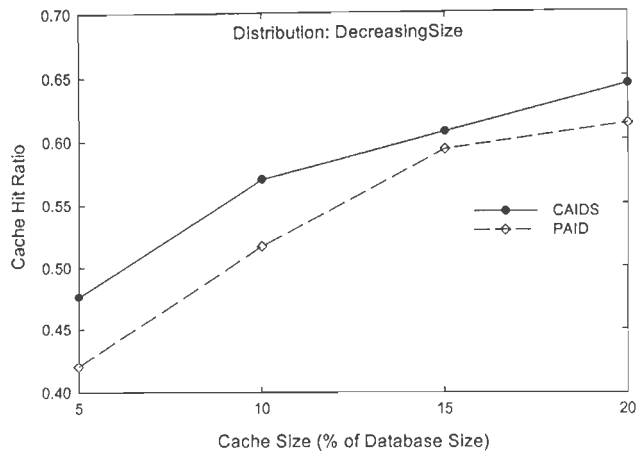
Figure 5.12 Cache Hit Ratio vs Moving Interval (Scope Distribution 2)



(a)

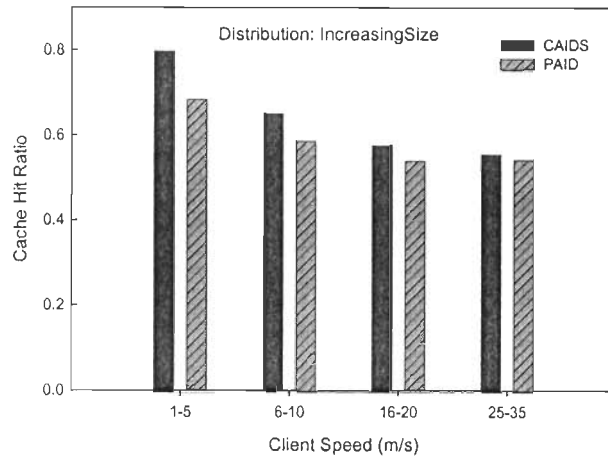


(b)

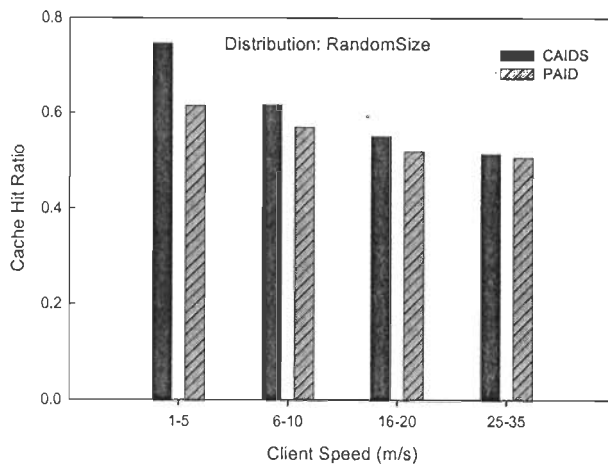


(c)

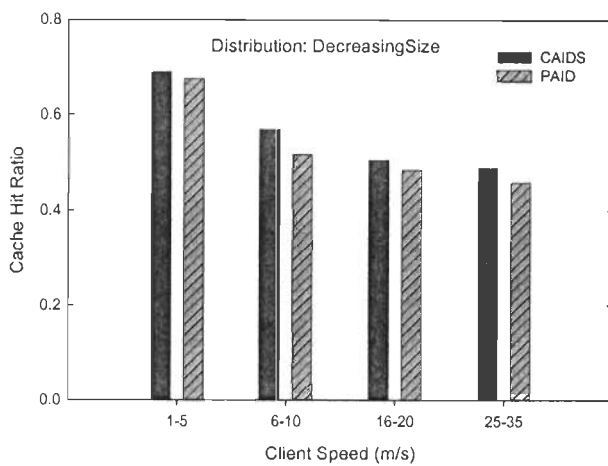
Figure 5.13 Cache Hit Ratio vs Cache Size (Scope Distribution 2)



(a)

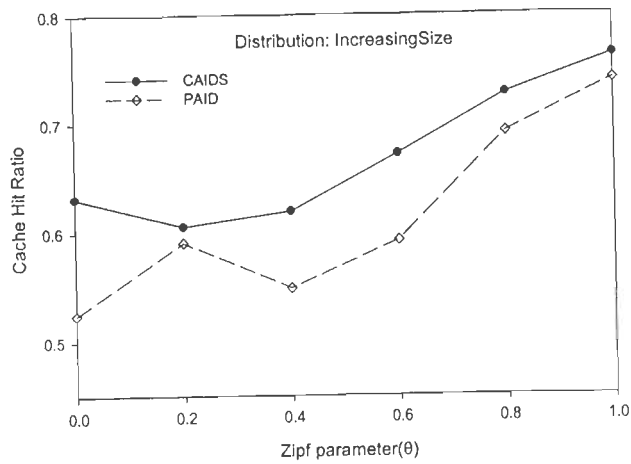


(b)

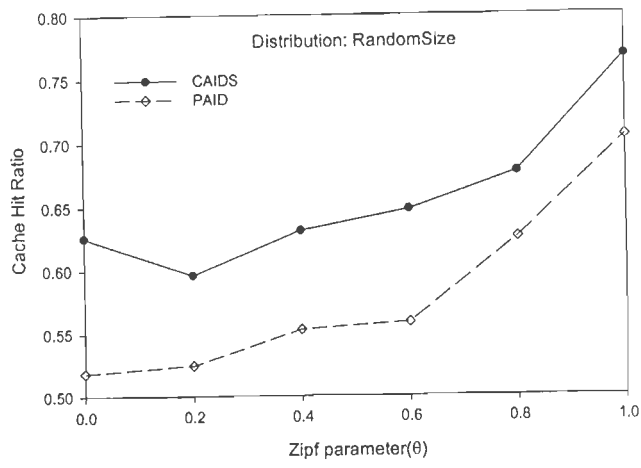


(c)

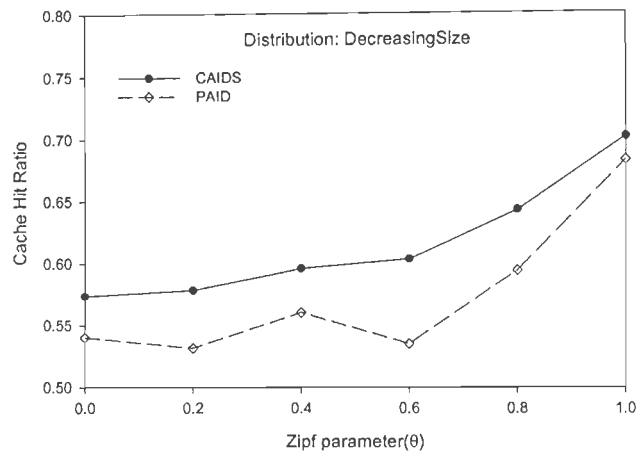
Figure 5.14 Cache Hit Ratio vs Client Speed (Scope Distribution 2)



(a)



(b)



(c)

Figure 5.15 Cache Hit Ratio vs Zipf parameter(θ) (Scope Distribution 2)

Table 5.1 Average Improvement of CAIDS over PAID By Varying System Parameters (Scope Distribution 2)

| System Parameters | Data Item Size Distributions | | |
|-------------------|------------------------------|------------|----------------|
| | IncreasingSize | RandomSize | DecreasingSize |
| Query Interval | 13.5 % | 15.0 % | 8.0 % |
| Moving Interval | 13.0 % | 12.5 % | 10.5 % |
| Cache Size | 11.2 % | 13.5 % | 8.0 % |
| Client Speed | 9.2 % | 9.0% | 6.0 % |

Performance comparisons of CAIDS and PAID is also done on real-data set (Scope Distribution 2) for IncreasingSize, RandomSize, and DecreasingSize distribution, which is shown in Figures 5.11 to 5.15. For Scope Distribution 2 also, we get similar improvement in performance of proposed policies as they were for Scope Distribution 1. The average percentage improvement of CAIDS over PAID for Scope Distribution 2 is given in Table 5.1.

5.4 CONCLUSIONS

In this chapter, we proposed a Recency/Frequency based location-dependent cache replacement policy CAIDS. CAIDS takes into account CRF value of data item, valid scope area, inverse data distance and inverse data item size. CRF plays an important role in improving the performance of LDIS. CRF makes best use of *Recency/Frequency* spectrum and allows a flexible trade-off between recency and frequency of references in making the replacement decision without any extra computational and storage overhead as compared to exponential ageing method. Our experiments show that though the recency factor is important because the access is location dependent but frequency can not be totally ignored because it reflects the user preferences and needs. Simulation results show that improvement of CAIDS over PAID was more than 25 % for IncreasingSize, 20 % for Random Size and 15 % for DecreasingSize.

CHAPTER 6

CONCLUSIONS AND SCOPE FOR FUTURE WORK

In this thesis, we have examined existing cache management policies and proposed new policies for location-dependent data in mobile environment based on geometric location model. This chapter summarizes the work done and discusses some future research directions.

6.1 CONCLUSIONS

A mobile computing environment enables clients to enjoy unrestricted mobility while continuing their computations as they are not required to maintain a fixed position in the network. In such an environment, location information becomes a new important parameter and introduces a new kind of information services, named as Location-Dependent Information Services (LDISs), whose services are dependent on the position of query issuers. Unlike traditional queries, which are assumed to be location-independent, the processing of Location-Dependent Queries has to take into account the client's physical position, which changes continuously in the mobile environment. Due to limitations of mobile environment and battery power of hand held devices, the processing needs to be efficient. Caching at client end is commonly employed to overcome user latency and facilitate data availability even on disconnections.

For maintaining consistency of the cached Location-Dependent Data (LDD), LDIS stores valid scope of the data item along with its value in the client's cache. The valid scope of a data value is needed to validate an answer to a specific location-dependent query which is only valid in a limited region. Valid scope of the data item is represented and stored as a convex polygon on the server. The overhead of storing all end points of the polygon in client's cache is large, so a subset of valid scope is stored that approximates the original valid scope. We observed that, modifying Caching-Efficiency-Based cache invalidation policy [14] to consider more choices in each iteration, gives better results. Based on this observation, we proposed a generalized algorithm CEB_G that selects the best suitable candidate for valid scope to maximize the caching efficiency. Though, CEB_G requires more computation time than CEB, but, as these candidate valid scopes can be calculated and stored at the server only once along with the actual valid scopes, it is acceptable. Moreover, CEB_G improves the precision by selecting more precise

representation of valid scope as compared to CEB. We compared its performance with the existing CEB algorithm. We further introduced a new metric, *Future Access*, which takes into account client's movement behavior. We proposed CEFAB and CEFAB_G algorithms based on the metric, Future Access. The results show that algorithms CEB_G, CEFAB, CEFAB_G with different system settings, give better performance than CEB. Among the proposed algorithms, CEFAB_G gives the best performance. But, computational overhead at server for CEFAB_G and CEB_G is higher than CEFAB. Moreover, in CEFAB and CEFAB_G, client has to send additional information i.e. the end of MI along with the current position, to the server, which requires extra computation at client's end, as compared to CEB_G. Thus, for low resource client CEB_G is preferred. Depending on the resources at the server, choice can be made between CEFAB and CEFAB_G.

Due to the limitation of the cache size, it is impossible to hold all accessed data items in the cache. As a result, cache replacement algorithms are used to find a suitable subset of data items for eviction. Existing cache replacement policies for LDIS only consider the data distance (directional/undirectional), but not the distance based on the predicted region or area where the client can be in near future. With client's random movement patterns, it is not always necessary that client will continue moving in the same direction. Therefore, we consider an area in the vicinity of client's current position, within which the client is likely to be present in near future and give priority to the cached data items that belong to this area irrespective of the client's movement direction. Based on this predicted region, we proposed cost function based Predicted Region based Cache Replacement Policy (PRRP), Prioritized Predicted Region based Cache Replacement Policy (PPRRP) and several Weighted Predicted Region based Cache Replacement Policies. Among the proposed policies, WPRRP-3 gives the best performance, from which we conclude that predicted regions has to be favored irrespective of direction of movement particularly where the client's movement pattern is random.

We further explored the effect of temporal characteristics (i.e., frequency and recency) of data access on location-dependent cache replacement. We proposed a cost function based cache replacement policy known as CAIDS. CAIDS considers both recency and frequency factors along with the data item size, valid scope area and data distance of a data item. Our experiments show that though the recency factor is important because the access is location dependent but frequency can not be totally ignored because it reflects the user preferences and needs.

6.2 SCOPE FOR FUTURE WORK

Location is an increasingly important parameter that was introduced by client's free mobility in mobile computing environment. Research on Location-Dependent Information Services has made great strides in recent years. Several location-aware prototyped applications like tour guides, city guides, and conference companions have demonstrated the potential of taking location information into account. But a number of challenges still remain. We look at some of the issues here and outline directions for future work.

- Existing work done by researchers is based only on simple queries such as “find the nearest restaurant”. Investigation of caching and query processing problems associated with general location-dependent queries such as “find the nearest hotel with a room rate below Rs. 200” is required. The demands become much greater if multiple dimensions are introduced or if complex spatial analysis is involved, or both [30,39]. Any future enhancements may vary considerably in their algorithmic complexity.
- We considered a single server environment for cache management. Investigation of handoff problem and cache management schemes in a global environment [113,119] (multiple server environments) is needed.
- The extent to which data is static or dynamic varies among the kinds of data involved. Geocoding and mapping data [101] are perhaps most static, since addresses and landmarks tend to be constant. Routing information is fairly static, although construction affects the status and the number of roads available. Demographic and yellow pages information is somewhat static, though the rate of any population and business changes affects its accuracy and value. Weather information is relatively dynamic, and traffic information is highly dynamic. Hence, there is a need to study the influence of LDD data and its updates on location-dependent caching strategies based on the observation of the pitfalls of attaching invalidation information to data that are updated frequently. Investigation of cache management schemes involving both location-dependent and time-dependent data is also required.
- Location Based Broadcast is an interesting area where a lot of research can be done. For example, restaurants advertising their menu or discount voucher, hotels announcing room availability, theaters listing last-minute tickets for sale, and many other applications providing information of their services to the users in their respective coverage areas.

Broadcast can be general, grouped or specific, depending upon number of users, time of the day and business season. Hence, there is a need of new indexing schemes [13] to broadcast LDD data. Moreover, the broadcast program needs to be highly adaptive.

- Prefetching of LDD based on spatial association rules is a new area where lots of research is needed. Constructing a prefetch set and using it to pull data items from server with little additional cost in one such area. Further, storing prefetched data may require replacement of existing data from the client's cache. Hence, study of prefetching schemes and cache replacement policies based on spatial data mining technique is needed for LDISs.
- Currently, no positioning system is accessible everywhere. In particular, there are large differences between outdoor and indoor positioning and related applications. For example, typical indoor applications at least require locating targets with the granularity of rooms and floors inside a building, which has so far been impossible to determine via satellite or cellular positioning. In most cases, GPS does not work at all inside buildings, although, there are some initiatives, such as Indoor GPS, for coping with this issue. As a consequence, it would be necessary to develop mechanisms that automatically select the best positioning method from all those that are available on the spot. This includes that the terminal or the network also dynamically switches between different methods for keeping the required level of accuracy, for example, from GPS to WLAN fingerprinting, if the target enters a building.
- Developments in recent years also spawned research in the recent years in the field of spatio-temporal databases and spatio-temporal data streams. For example, medical facilities can track staff and monitor patients for emergency response, and coordinate logistics in case of an emergency (e.g., navigate to the closest hospital or the nearest emergency department that has available capacity). Emergency response centers can help people navigate and evacuate faster in case of disasters such as floods, earthquakes, or terrorist attacks. System needs to consider both the positions of the moving/stationary objects as well as the queries. Thus, there is a need for new real-time spatio-temporal query processing algorithms as well as new association rule mining algorithms for spatio-temporal data streams that deal with large numbers of moving/stationary objects and large numbers of continuous spatio-temporal queries where near-real time response is a necessity.

The number of services potentially relevant for a given location can grow enormously and be very dynamic, with new services appearing and disappearing constantly. The goodness of any

policy lies in the adaptiveness of its algorithm, which adjusts dynamically to the changes in the workload characteristics. The LDIS community is experiencing an increase in the number of initiatives that are addressing these challenges. They are taking place in the form of intensive cooperation between research, industry, and standardization, which are working very hard on making the next and future generations of LDISs a success.

REFERENCES

- [1] A. Balamash and M. Krunz, "An Overview of Web Caching Replacement Algorithms," *IEEE Communications Surveys & Tutorials*, Vol. 6, No. 2, pp. 44-56, 2004.
- [2] A. Guttman, "R-trees: A Dynamic Index Structure for Spatial Searching," In *the Proceedings of the ACM SIGMOD International Conference on Management of Data*, Boston, Massachusetts, USA, pp. 47-57, 1984.
- [3] A. Kahol and S. Khurana, "A Strategy to Manage Cache Consistency in a Disconnected Distributed Environment," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 12, No. 7, pp. 686-700, July 2001.
- [4] A. Kahol, S. Khurana, S. Gupta and P. Srimani, "An Efficient Cache Maintenance Scheme for Mobile Environment," In *the Proceedings of the Twentieth International Conference on Distributed Computing Systems*, pp. 530-537, April 2000.
- [5] A. Madhukar and R. Alhajj, "An adaptive energy efficient cache invalidation scheme for mobile databases," In *the Proceedings of the 2006 ACM Symposium on Applied Computing Session: Mobile Computing and Application (MCA)*, Dijon, France, pp. 1122-1126, 2006.
- [6] A. Nanopoulos, D. Katsaros and Y. Manopoulos, "A Data Mining Algorithm for Generalized Web Prefetching," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 5, pp. 1155-1169, Sept/Oct. 2003.
- [7] A. P. Sistla, O. Wolfson, S. Chamberlain and S. Dao, "Modeling and Querying Moving Objects," In *the Proceedings of the 13th International Conference on Data Engineering (ICDE'97)*, Birmingham, UK, pp. 422-432, April 1997.
- [8] A. K. Elmagarmid, J. Jing, A. Helal and C. Lee, "Scalable Cache Invalidation Algorithms for Mobile Data Access," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 6, pp. 1498-1511, November/December 2003.
- [9] A.Y. Seydim, M. H. Dunham and V. Kumar, "Location Dependent Query Processing," In *the Proceedings of the 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE'01)*, California, USA, pp. 47-53, May 2001.
- [10] A.Y. Seydim, M.H. Dunham and V. Kumar, "An Architecture for Location Dependent Query Processing," In *the Proceedings of the 12th International Workshop on Database*

and Expert Systems Applications (DEXA), Munich, Germany, pp. 549-555, 3rd -7th September 2001.

- [11] B. Zheng and D. L. Lee, "Processing Location-Dependent Queries in a Multi-cell Wireless Environment," In *the Proceedings of the 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE'01)*, Santa Barbra, CA, USA, pp. 54-65, May 2001.
- [12] B. Zheng and D. L. Lee, "Semantic Caching in Location-Dependent Query Processing," In *the Proceedings of the 7th International Symposium on Spatial and Temporal Databases (SSTD'01)*, Los Angeles, CA, USA, pp. 97-116, July 2001.
- [13] B. Zheng, "Indexing of Location-Dependent Data in Mobile Computing Environments," *PhD Thesis*, Hong Kong University of Science and Technology, Hong Kong, 2002.
- [14] B. Zheng, J. Xu and D. L. Lee, "Cache Invalidation and Replacement Strategies for Location-Dependent Data in Mobile Environments," *IEEE Transactions on Computers*, Vol. 51, No. 10, pp. 1141-1153, October 2002.
- [15] B. Zheng, W.-C. Lee and D. L. Lee, "On Semantic Caching and Query Scheduling for Mobile Nearest-Neighbor Search," *Wireless Networks*, Kluwer Academic Publishers, Vol. 10, No. 6, pp. 653-664, November 2004.
- [16] C. Lu, G. Xing, O. Chipara and C. L. Fok, "MobiQuery: A Spatio Temporal Data Service for Sensor Networks," In *the Proceedings of the 2nd International Conference on Embedded Networked Sensor System (ACM SenSys'04)*, Baltimore, USA, pp. 320-334, 2004.
- [17] C. W. Lin and D. L. Lee, "Adaptive Data Delivery in Wireless Communication Environments," In *the Proceedings of the 20th IEEE International Conference on Distributed Computing Systems (ICDCS'2000)*, Taipei, Taiwan, pp. 444-452, April 2000.
- [18] C.C. Chen, C. Lee, C.C. Wang and Y. C. Chung , "Prefetching LDD: A Benefit-Oriented Approach," In *the Proceeding of the 2006 International Conference on Communications and Mobile Computing (IWCMC '06)*, Vancouver, Canada, pp. 1103-1108, July 2006.
- [19] C. Aggarwal, J. L. Wolf and P. S. Yu, "Caching on the World Wide Web," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 1, pp. 94-107, January/February 1999.

- [20] D. Aksoy and M. Franklin, "R x W: A Scheduling Approach for Large-Scale On-Demand Data Broadcast," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 6, pp. 846-860, December 1999.
- [21] D. Barbara and T. Imielinski, "Sleepers and Workaholics: Caching Strategies in Mobile Environments," In *the Proceedings of the ACM SIGMOD Conference on Management of Data*, Minneapolis, USA, pp. 1-12, 1994.
- [22] D. Barbara and T. Imielinski, "Sleepers and Workaholics: Caching Strategies in Mobile Environments (Extended Version)," *MOBIDATA: An Interactive Journal of Mobile Computing*, Vol. 1, No. 1, November 1994.
- [23] D. Barbara, "Mobile Computing and Databases: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 1, pp. 108-117, January/February 1999.
- [24] D. J. Cook and S. K. Das, "Smart Environments: Technologies, Protocols, and Applications," Wiley-Interscience Publications, ISBN 0-471-54448-5, 2005
- [25] D. Lee, J. Choi, J.H. Kim, S.H. Noh, S.L. Min, Y. Cho and C.S. Kim, " LRFU: A Spectrum of Policies that Subsumes the Least Recently Used and Least Frequently Used Policies," *IEEE Transactions on Computers*, Vol. 50, No. 12, pp. 1352-1361, Dec. 2001.
- [26] D. Lee and W. W. Chu, "Semantic Caching via Query Matching for Web Sources," In *the Proceedings of the 8th International Conference on Information and Knowledge Management (CIKM'99)*, Kansa City, USA, pp. 77-85, November 1999.
- [27] D. L. Lee, W.-C. Lee, J. Xu and B. Zheng, "Data Management in Location-Dependent Information Services," *IEEE Pervasive Computing*, Vol. 1, No. 3, pp. 65-72, July 2002.
- [28] D. P. Agrawal and Q. A. Zeng, "*Introducing to Wireless and Mobile Systems*," Second Edition, Thomson, 2006.
- [29] D. Acharya and V. Kumar, "Location based indexing scheme for DAYS," In *the Proceedings of the 4th ACM International Workshop on Data Engineering for Wireless and Mobile Access*, Baltimore, USA, pp. 17-24, June 2005.
- [30] D. Papadias, Y. Tao, K. Mouratidis and C. K. Hui, "Aggregate Nearest Neighbor Queries in Spatial Databases," *ACM Transactions on Database Systems (TODS)*, Vol. 30, No. 2, June 2005.

- [31] E. Pitoura, and B. Bhargava, "Building Information Systems for Mobile Environments," In *the Proceedings of 3rd International Conference on Information and Knowledge Management (CIKM'94)*, pp. 371-378, November 1994.
- [32] E. Yajima, T. Hara, M. Tsukamoto and S. Nishio: "Scheduling and Caching Strategies for Correlated Data in Push-Based Systems," *ACM Applied Computing Review (ACM ACR)*, Vol. 9, No.1, pp.22-28, July 2001.
- [33] G. Cao and C. Das, "On the Effectiveness of a Counter-Based Cache Invalidation Scheme and Its Resiliency to Failures in Mobile Environments," In *the Proceedings of the 20th IEEE Symposium on Reliable Distributed Systems (SRDS'01)*, pp. 247-256, 2001.
- [34] G. Cao, "A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 5, pp. 1251-1265, September/October 2003.
- [35] G. Cao, "On Improving the Performance of Cache Invalidation in Mobile Environments," *ACM Kluwer Mobile Network and Applications*, Vol. 7, No. 4, pp. 291-303, 2002.
- [36] G. Cao, "Proactive Power-Aware Cache Management for Mobile Computing Systems," *IEEE Transactions on Computers*, Vol. 51, No. 6, pp. 608-621, June 2002.
- [37] G. H. Forman and J. Zahorjan, "Challenges of Mobile Computing," *IEEE Computer Society*, Vol. 27, No. 6, pp. 38-47, April 1994.
- [38] G. Yavas, D. Katsaros, O. Ulusoy and Y. Manolopoulos, "A Data Mining Approach for Location Prediction in Mobile Environments," *Elsevier Data and Knowledge Engineering*, Vol. 54, No. 2, pp. 121-146, 2005.
- [39] Gaede and O. Gunther, "Multidimensional Access Methods," *ACM Computing Surveys*, Vol. 30, No. 2, pp.170-231, June 1998.
- [40] H. Shen, M. Kumar, S. K. Das and Z. Wang, "Energy-Efficient Caching and Prefetching with Data Consistency in Mobile Distributed Systems," In *the Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 67-76, 2004.
- [41] H. Shen, M. Kumar, S. K. Das and Z. Wang, "Energy-Efficient Data Caching and Prefetching of Mobile Devices Based on Utility," *ACM/Kluwer Journal of Mobile Networks and Applications(MONET), Special Issue on Mobile Services*, Vol. 10, No. 4, pp. 475-486, August 2005.

- [42] H. Song and G. Cao, "Cache-Miss-Initiated Prefetch in Mobile Environments," *Journal of Computer Communications*, Vol. 28, No. 7, pp. 741-753, 2005.
- [43] I. A. Getting, "The Global Positioning System," *IEEE Spectrum*, Vol. 30, No. 12, pp. 36-47, December 1993.
- [44] Il-dong Jung, Young-ho You, Jong-hwan Lee and Kyungsok Kim, "Broadcasting and caching policies for location-dependent queries in urban areas," In *the Proceedings of the 2nd International Workshop on Mobile Commerce*, Atlanta, USA, pp. 54-60, September 2002.
- [45] J. Cai and K. L. Tan, "Energy-Efficient Selective Cache Invalidation," *ACM/Baltes Journal of Wireless Networks (WINET)*, Vol. 5, No.6, pp. 489-502, May 1999.
- [46] J. Jing, A. Helal and A. Elmagarmid, "Client-Server Computing in Mobile Environments," In *the Proceedings of the ACM Computing Surveys*, Vol. 31, No. 2, pp. 117-157, June 1999.
- [47] J. Jing, A. K. Elmagarmid, A. Helal and R. Alonso, "Bit-Sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments," *Mobile Networks and Applications*, Vol. 2, No. 2, pp. 115-127, October 1997.
- [48] J. O' Rourke, *Computational Geometry in C*, chapter 5, Univ. of Cambridge Press, 1998.
- [49] J. W. Wong, "Broadcast Delivery," In *the Proceedings of the IEEE*, Vol. 76, No. 12, pp. 1566-1577, December 1988.
- [50] J. Xu, "Client-side Data Caching in Mobile Computing Environments," *PhD Thesis*, Hong Kong University of Science and Technology, Hong Kong, 2002.
- [51] J. Xu, B. Zheng, W.-C. Lee and D. L. Lee, "Energy efficient index for querying location-dependent data in mobile broadcast environments," In *the Proceedings of the 19th IEEE International Conference on Data Engineering (ICDE'03)*, Bangalore, India, pp. 239-250, March 2003.
- [52] J. Xu, D. L. Lee, Q. Hu and W.-C. Lee, "Data Broadcast," *Handbook of Wireless Networks and Mobile Computing*, Chapter 11, Ivan Stojmenovic, Ed., New York: John Wiley & Sons, ISBN 0-471-41902-8, pp. 243-265, January 2002.
- [53] J. Xu, Q. Hu, W. C. Lee and D. L. Lee, "An Optimal Cache Replacement Policy for Wireless Data Dissemination Under Cache Consistency," In *the Proceedings of the*

- 30th *International Conference on Parallel Processing (ICPP'01)*, Valencia, Spain, pp. 267-274, September 2001.
- [54] J. Xu, Q. Hu, W. C. Lee and D. L. Lee. "SAIU: An Efficient Cache Replacement Policy for Wireless On-demand Broadcasts," In *the Proceedings of the 9th ACM International Conference on Information and Knowledge Management (CIKM)*, Mc Lean, VA, USA, pp. 46-53, Nov. 2000.
- [55] J. Xu, X. Tang and D. L. Lee, "Performance Analysis of Location-Dependent Cache Invalidation Schemes for Mobile Environments," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 2, pp. 474-488, March/April 2003.
- [56] J. Xu, X. Tang, D. L. Lee and Q. Hu, "Cache Coherency in Location-dependent Information Services for Mobile environment," In *the Proceedings of the 1st International Conference on Mobile Data Access (MDA '99)*, Hong Kong, Springer-Verlag LNCS, Vol. 1748, pp. 182-193, December 1999.
- [57] J. Zhang and Le Gruenwald, "Prioritized Sequencing for Efficient Query on Broadcast Geographical Information in Mobile-Computing," In *the Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems*, McLean, USA, pp. 88-93, November 2002.
- [58] J. Zhang, M. Zhu and D. Papadias, Y. Tao, and D. Lee, "Location-based Spatial Queries," In *the Proceedings of the ACM SIGMOD International Conference on Management of Data*, San Diego, USA, pp. 443-454, 2003.
- [59] J. C. Yuen, E. Chan, K. lam and H. W. Lueng, "Cache Invalidation Scheme for Mobile Computing Systems with Real-Time Data," In *the Proceedings of the ACM SIGMOD*, Vol. 29, No. 4, pp. 34-49, December 2000.
- [60] K. C. K. Lee, H. V. Leong and A. Si, "A Semantic Broadcast Scheme for a Mobile Environment Based on Dynamic Chunking," In *the Proceedings of the 20th IEEE International Conference on Distributed Computing Systems (ICDCS'2000)*, Taipei, Taiwan, pp. 522-529, April 2000.
- [61] K. C. K. Lee, H. V. Leong and A. Si, "Semantic query caching in a mobile environment," *Mobile Computing and Communication Review*, Vol. 3, No. 2, pp. 28-36, 1999.
- [62] K. Cheverst, K. Mitchell and N. Davies, "The role of adaptive hypermedia in a context-aware tourist Guide," *Communications of the ACM*, Vol. 45, No. 5, pp. 47-51, 2002.

- [63] K. J. Nesbit and J. E. Smith, "Data Cache Prefetching Using a Global History Buffer," In *the Proceedings of the 10th International Symposium on High Performance Computer Architecture (HPCA'04)*, pp. 96-96, 14-18th February 2004.
- [64] K. J. Cios and L. Kurgan, "Trends in Data Mining and Knowledge Discovery," in *Pal N.R., Jain, L.C. and Teoderesku, N. (Eds.), Knowledge Discovery in Advanced Information Systems, Springer, 2002.*
<http://citeseer.ist.psu.edu/cios05trends.html>
- [65] K. L. Tan and J. Cai, "Broadcast-Based Group Invalidation: An Energy-Efficient Cache Invalidation Strategy," *Elsevier Information Sciences*, Vol. 100, No. 1-4, pp. 229-254, 1997.
- [66] K. L. Tan, J. Cai and B. C. Ooi, "An Evaluation of Cache Invalidation Strategies in Wireless Environments," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 12, No. 8, pp. 789-807, August 2001.
- [67] K. L. Wu, P. S. Yu and M. S. Chen, "Energy-Efficient Caching for Wireless Mobile Computing," In *the Proceedings of the 12th International Conference on Data Engineering (ICDE'96)*, New Orleans, USA, pp. 336-343, Feb. 26-March 1, 1996.
- [68] K. Lai, Z. Tari and P. Bertok, "Location-Aware Cache Replacement for Mobile Environments," *IEEE Global Telecommunication Conference (GLOBECOM 04)*, Vol. 6, pp. 3441-3447, 29th November- 3rd December 2004.
- [69] L. A. Kurgan and P. Musilek, "A Survey of Knowledge Discovery and Data Mining Process Models," *The Knowledge Engineering Review*, Vol. 21, No. 1, pp. 1-24, Cambridge University Press, March 2006.
- [70] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications," In *the Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99)*, New York, USA, Vol. 1, pp. 126-134, March 1999.
- [71] L. Feeney and M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment," In *the Proceedings of IEEE INFOCOM*, Anchorage, USA, Vol. 3, No. 8, pp. 1549-1557, 2001.

- [72] L. Kleinrock, "Nomadicity: Anytime, Anywhere in a Disconnected World," *In the Proceedings of Mobile Network and Applications*, Vol. 1, No. 4, pp. 351-357, December 1996.
- [73] L. Yin and G. Cao, "Adaptive Power-Aware Prefetch in Wireless Networks," *IEEE Transactions on Wireless Communication*, Vol. 3, No. 5, pp. 1648-1658, 2004.
- [74] L. Yin, G. Cao and Y. Cai, "A Generalized Target-Driven Cache Replacement Policy for Mobile Environments," *Journal of Parallel and Distributed Computing*, Vol. 65, No. 5, pp. 583-594, 2005.
- [75] L. Yin, G. Cao and Y. Cai, "A Generalized Target-Driven Cache Replacement Policy for Mobile Environments," *In the Proceedings of the IEEE Symposium on Applications on the Internet*, pp. 14-21, January 2003.
- [76] M. Berg, M. Kreveld M. Overmars and O. Schwarzkopf, "*Computational Geometry: Algorithms and Applications*," chapter 7, New York, NY, USA, Springer-Verlag, 1996.
- [77] M. Erwig, R. H. Guting, M. Schneider and M. Vazirgiannis, "Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases," *GeoInformatica*, Vol. 3, No. 3, pp. 269-296, 1999.
- [78] M. H. Dunham and A. Helal, "Mobile Computing and Databases: Anything Thing New?," *In the Proceedings of the ACM SIGMOD Record*, Vol. 24, No. 4, pp. 5-9, December 1995.
- [79] M. K. Ho Yeung and Y. -K. Kwok, "New Invalidation Algorithms for Wireless Data Caching with Downlink Traffic and Link Adaptation," *In the Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04)*, April 2004.
- [80] M. Satyanarayanan, "A Catalyst for Mobile and Ubiquitous Computing," *IEEE Pervasive Computing*, Vol. 1, pp. 2-5, January-March 2002.
- [81] M. Satyanarayanan, "Fundamental Challenges of Mobile Computing," *ACM Symposium on Principles of Distributed Computing (PODC'95 invited lecture)*, May 1996.
- [82] M. H. Dunham, and V. Kumar, "Location Dependent Data and Its Management in Mobile Databases," *In the Proceedings of the International Workshop Database and Expert System Systems Applications*, Vienna, Austria, pp. 414-419, August 1998.

- [83] M. K. H. Yeung and Y.-K. Kwok, "Wireless Cache Invalidation Schemes with Link Adaptation and Downlink Traffic," *IEEE Transactions on Mobile Computing*, Vol. 4, No. 1, pp. 68-83, Jan./Feb. 2005.
- [84] N. B. Priyantha, A. Chakraborty and H. Balakrishnan, "The Cricket Location-Support System," In *the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobi-Com'2000)*, Boston, MA, USA, pp. 32-43, August 2000.
- [85] N. Chand, "Cache Management in Mobile Computing Environment," *PhD Thesis*, Indian Institute of Technology Roorkee, India, 2006.
- [86] P. Cao, E. W. Felten, A. Karlin and K. Li, "A Study of Integrated Prefetching and Caching Strategies," In *the Proceedings of the ACM SIGMETRICS*, Vol. 23, No. 1, pp. 171-182, May 1995.
- [87] Q. Hu and D. L. Lee, "Adaptive Cache Invalidation Methods in Mobile Environments," In *the Proceedings of the 6th IEEE International Symposium on High Performance Distributed Computing*, August 1997.
- [88] Q. Hu and D. L. Lee, "Cache Algorithms Based on Adaptive Invalidation Reports for Mobile Environments," *Cluster Computing*, Vol. 1, pp. 39-50, 1998.
- [89] Q. Ren and M. H. Dunham, "Using Clustering for Effective Management of a Semantic Cache in Mobile Computing," In *the Proceedings of the International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE'99)*, Seattle, WA, USA, pp. 94-101, August 1999.
- [90] Q. Ren and M. H. Dhunham, "Using Semantic Caching to Manage Location Dependent Data in Mobile Computing," In *the Proceedings of 6th ACM/IEEE Mobile Computing and Networking (MobiCom)*, Boston, USA, pp. 210-221, 2000.
- [91] Q. Ren, M. H. Dunham and Vijay Kumar, "Semantic Caching and Query Processing," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 1, pp. 192-210, January/February 2003.
- [92] R. Alonso, D. Barbara, G. Molina and H. G. Molina, "Data Caching Issues in an Information Retrieval System," *ACM Transactions on Database Systems*, Vol. 15, No. 3, pp. 359-384, September 1990.

- [93] R. Cooley, B. Mobasher and J. Srivastava, "Data Preparation for Mining World Wide Web Browsing Patterns," *Knowledge and Information Systems*, Vol. 1, No. 1, pp. 5-32, 1999.
- [94] R. Cucchiara, M. Piccardi and A. Prati, "Temporal Analysis of Cache Prefetching Strategies for Multimedia Applications," In *the Proceeding of IEEE International Conference on Performance, Computing and Communications (IPCC'01)*, Phoenix, USA, pp. 311-318, 4-6th April 2001.
- [95] S. Acharya and S. Muthukrishnan, "Scheduling On-demand Broadcasts: New Metrics and Algorithms," In *the Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, Dallas, TX, USA, pp. 43-54, October 1998.
- [96] S. Acharya, M. Franklin and S. Zdonik, "Prefetching From a Broadcast Disk," In *the Proceedings of the International Conference on Data Engineering (ICDE'96)*, New Orleans, USA, pp. 276-285, February/March 1996.
- [97] S. Acharya, R. Alonso, M. Franklin and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments," In *the Proceedings of the ACM SIGMOD Conference on Management of Data*, San Jose, USA, pp. 199-210, May 1995.
- [98] S. Dar, M. J. Franklin, B. T. Jonsson, D. Srivastava and M. Tan, "Semantic Data Caching and Replacement," In *the Proceedings of the 22nd International Conference on Very Large Databases(VLDB)*, pp. 330-341, 1996.
- [99] S. Drakatos, N. Pissinou, Kia Makki and Christos Douligeris, "A Context-Aware Prefetching Strategy for Mobile Computing Environments," In *the Proceedings of the International Conference on Communications and Mobile Computing (ICMC'06)*, Vancouver, Canada, pp. 1109-1116, July 3-6, 2006.
- [100] S. Podlipnig and L. Z. Boszormenyi, "Survey of Web Cache Replacement Strategies," *ACM Computing Surveys (CSUR)*, Vol. 35, No. 4, pp. 374-398, December 2003
- [101] S. Upadhyaya, A. Chaudhury, K. Kwiat and M. Weiser, "*Mobile Computing: Implementing Pervasive Information and Communication Technologies*," Kluwer Academic Publisher, ISBN 1-4020-7137-X, 2002.

- [102] S. K. Gupta, V. Bhatnagar and S. K. Wasan, "Architecture for Knowledge Discovery and Knowledge Management," *Springer Knowledge Information System*, Vol. 7, No. 3, pp. 310-336, 2005.
- [103] S.P. Vander Wiel and D.J. Lilja, "When Caches Aren't Enough: Data Prefetching Techniques," *IEEE Computer*, Vol. 30, No. 7, pp. 23-30, July 1997
- [104] S. K. Madria, B. K. Bhargava, E. Pitoura and V. Kumar, "Data Organization Issues for Location-Dependent Queries in Mobile Computing," In *the Proceedings of the East-European Conference on Advances in Databases and Information Systems Held Jointly with International Conference on Database Systems for Advanced Applications: Current Issues in Databases and Information Systems, LNCS*, Vol.1884, pp. 142-156, 2000.
- [105] S.-Y. Wu and K.-T. Wu, "Effective Location Based Services with Dynamic Data Management in Mobile Environments," *Wireless Networks*, Vol. 12, No. 3, Kluwer Academic Publishers, pp. 369-381, May 2006.
- [106] Spatial Datasets. Website at <http://www.rtreeportal.org>, 2005.
- [107] T. Camp, J. Boleng and V. Davies, "A Survey of Mobility Model for Ad Hoc Network Research," *Wireless Communication & Mobile Computing (WCMC): Special Issue on Mobile AdHoc Networking: Research, Trends and Applications*, Vol. 2, No. 5, pp. 483-502, 2002.
- [108] T. Hara, "Cooperative Caching by Mobile Clients in Push-Based Information Systems," In *the Proceedings of 11th International Conference on Information and Knowledge Management (CIKM'02)*, McLean, USA, pp. 186-193, 2002.
- [109] T. Imielinski and B.R. Badrinath, "Mobile Wireless Computing: Challenges in Data Management," *Communications of ACM*, Vol. 37, No.10, pp. 18-28, October 1994.
- [110] T. Sellis, N. Roussopoulos and O. Faloutsos, "The R⁺-Tree: A Dynamic Index for Multi-Dimensional Objects," In *the Proceedings of the 13th International Conference on Very Large Data Bases (VLDB'87)*, Brighton, England, pp. 507-518, September 1987.
- [111] U. Fayyad, G. P.-Shapiro and P. Smyth, "From Data Mining to Knowledge Discovery in Databases," *AI Magazine*, Vol. 17, No. 3, pp. 37-54, 1996.
<http://citeseer.ist.psu.edu/fayyad96from.html>

- [112] W. C. Lee and D. L. Lee, "Signature Caching Techniques for Information Filtering in Mobile Environments," *ACM/Baltzer Journal of Wireless Networks (WINET)*, Vol. 5, No. 1, pp. 57-67, 1999.
- [113] W.-C. Peng and M.-S. Chen, "Design and Performance Studies of an Adaptive Cache Retrieval Scheme in a Mobile Computing Environment," *IEEE Transactions on Mobile Computing*, Vol. 4, No. 1, pp. 29-40, January/February 2005.
- [114] W.-G. Teng, C.-Y. Chang and M.-S. Chen, "Integrating Web Caching and Web Prefetching in Client-Side Proxies," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 16, No. 5, pp. 444-455, May 2005.
- [115] Y. Bao, R. Alhajj and K. Barker, "Hybrid Cache Invalidation Schemes in Mobile Environments," In *the Proceedings of the IEEE/ACS International Conference on Pervasive Services (ICPS'04)*, pp. 209-218, July 2004.
- [116] Y. Saygin and O. Ulusoy, "Exploiting Data Mining Techniques for Broadcasting Data in Mobile Computing Environments," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14, No. 6, pp. 1387-1399, November-December 2002.
- [117] Y.-B. Lin, W.-R. Lai and J.-J. Chen, "Effects of Cache Mechanism on Wireless Data Access," *IEEE Transactions on Wireless Communications*, Vol. 2, No. 6, pp. 1247-1258, 2003.
- [118] Z. Wang, M. Kumar, S. K. Das and H. Shen, "Investigation of Cache Maintenance Strategies for Multi-Cell Environments," In *the Proceedings of the IEEE International Conference on Mobile Data Management (MDM)*, LNCS, Vol. 2574, pp. 29-44, 2003
- [119] Z. Wang, M. Kumar, S.K. Das and H. Shen, "Dynamic Cache Consistency Schemes for Wireless Cellular Networks," *IEEE Transactions on Wireless Communications*, Vol. 5, No. 2, pp. 366-376, February 2006.
- [120] Z. Wang, S. K. Das, H. Che and M. Kumar, "A Scalable Asynchronous Cache Consistency Scheme (SACCS) for Mobile Environments," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 11, pp. 983-995, November 2004.
- [121] Z. Wang, S. K. Das, H. Che and M. Kumar, "SACCS: Scalable Asynchronous Cache Consistency Scheme for Mobile Environments," In *the Proceedings of the 23rd International Conference on Distributed Computing System Workshops (ICDCSW'03)*, pp. 797-802, May 2003.

- [122] Z. Xu, Y. Hu and L. Bhuyan, "Exploiting Client Cache: A Scalable and Efficient Approach to Build Large Web Cache," In *the Proceeding of the 18th International Parallel and Distributed Processing Symposium (IPDPS)*, Santa Fe, New Mexico, pp. 55-64, April 26-30, 2004.

Author's Research Publications

International Conferences

1. Ajey Kumar, Manoj Misra and A.K. Sarje, "A Weighted Cache Replacement Policy For Location Dependent Data In Mobile Environments," In *Proceeding of 22nd ACM Symposium on Applied Computing (SAC'07) Mobile Computing and Applications (MCA) Technical Track*, Seoul, Korea, pp. 920-924, 11th- 15th March, 2007.
2. Ajey Kumar, Manoj Misra and A.K. Sarje, "A Predicted Region based Cache Replacement Policy For Location Dependent Data In Mobile Environment," In *Proceeding of 2nd IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'06)*, Wuhan, China, 22nd -24th September, 2006.
3. Ajey Kumar, Manoj Misra and A.K. Sarje, "An Improved Cache Replacement Policy For Location Dependent Data In Mobile Environment," In *Proceeding of 10th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI'06)*, Orlando, Florida, USA, Vol. 3, pp. 167-172, 16th -19th July, 2006.
4. Ajey Kumar, Manoj Misra and A.K. Sarje, "A New cache replacement policy for location dependent data in mobile environment," In *Proceeding of 3rd IEEE and IFIP International Conference on Wireless and Optical Communication Networks (WOCN'06)*, Bangalore, India, 11th-13th April, 2006.
5. Ajey Kumar, Manoj Misra and A.K. Sarje, "A New Metric for Cache Invalidation of Location Dependent Data in Mobile Environment," In *Proceeding of 4th Asian International Mobile Computing Conference (AMOC'06)*, Kolkata, India, pp. 255-263, 4th-7th January, 2006.
6. Ajey Kumar, Manoj Misra and A.K. Sarje, "Strategies for Cache Invalidation of Location Dependent Data in Mobile Environment," In *Proceeding of International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '05)*, Monte Carlo Resort, Las Vegas, Nevada, USA, pp. 38-44, 27th- 30th June, 2005.
7. Ajey Kumar, Manoj Misra and A.K. Sarje, "Selecting Efficient Valid Scope for Cache Invalidation of Location Dependent Data in Mobile Environment," In *Proceeding of Conference on Wireless Communication and Sensor Networks (WCSN)*, India, 5th-6th March, 2005.

International Journals

8. Ajey Kumar, Manoj Misra and A.K. Sarje, "A New Metric for Geometric Model Based Cache Invalidation of Location Dependent Data in Mobile Environment," *Efficient Heuristics for Information Organization, Journal of Computer Science (Special Issue)* pp. 34-40, 2005.

Under Review (International Journals)

1. Ajey Kumar, A.K. Sarje and Manoj Misra, "Prioritized Predicted Region based Cache Replacement Policy for Location Dependent Data in Mobile Environment," *International Journal of Ad Hoc and Ubiquitous Computing*.
2. Ajey Kumar, A.K. Sarje and Manoj Misra, "Weighted Predicted Region based Cache Replacement Policy for Location Dependent Data in Mobile Environments," *International Journal of Wireless and Mobile Computing (IJWMC)*.