A Data Management Framework for Mobile Ad hoc Network (MANET)

By

Noman Islam

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

2013

Supervised By: Dr. Zubair Ahmed Shaikh
This page has been intentionally left blank
Author’s Declaration

It is hereby declared that the work presented in this thesis is my original work and does not contain any material that is already published by another person except where properly cited and referenced. The submitted text contains a description of my research work that I have carried out during my PhD studies and does not include any material that has been submitted to qualify for the award of any other degree or diploma in any university or institution.

Signature _____________________

Name_________________________
Plagiarism Undertaking

It is hereby declared that the work presented in this thesis is not copied or plagiarized from any other source or existing literature, except where due references and citations are made.

Signature _____________________

Name_________________________
This page has been intentionally left blank
Abstract

Mobile Ad hoc Network (MANET) is defined as an infrastructure-less network created spontaneously among a set of nodes. Data management is an issue of prime concern in MANET because of the massive amount of data exchanges among hosts of the network. Data management requires novel mechanisms to confront with issues of data discovery, data representation and cache management etc. To address them, this thesis has proposed a framework called Association Rules Based Network Layer Data and Information Management System for Mobile Ad hoc Network (ALADIN). The novelty of ALADIN lies in utilization of cross-layer design, correlation among data items and semantic data representation. The cross-layer information is utilized for data discovery and maintaining consistency of cached data. The correlation among data items is exploited for cache replacement and enhancing the performance of discovery phase. A multilevel ontology is utilized for ensuring the semantic interoperability among nodes in scalable fashion. The framework has been implemented in JIST/SWANS simulator. A set of experiments with different network settings and mobility models has been done to evaluate the framework. The results show the improvement in hit ratio, latency and scalability of the data management operations with the proposed framework.
Acknowledgments

All the praise and thanks to Allah (SWT), the Lord of whole universe and Peace and Blessings be upon Muhammad (SAW), the last Messenger of Allah. It was not possible to accomplish this milestone without the help of various people. I would like to express my thanks to all of them who have supported me in various ways during the progression of my studies.

First and foremost, I would like to acknowledge Higher Education Commission (HEC), Pakistan for granting me a fully funded PhD scholarship. I am indebted to thank my supervisor Prof. Dr. Zubair A. Shaikh for his guidance, support and encouragements. I am grateful to the Center for Research in Ubiquitous Computing (CRUC), National University of Computer and Emerging Sciences (NUCES), Karachi, Pakistan and the research fellows associated with CRUC for their constant cooperation during my research endeavor.

I am also thankful to Prof. Dr. Asim-ur-Rehman (Director, NUCES) and Dr. Jawwad Shamsi (Head of Computer Science Department, NUCES) for providing the administrative support required for this accomplishment. I would also like to mention the names of Dr. Tariq Mahmood and Dr. Shahid Shaikh, current and ex-coordinator of the Graduate Students Committee (GSC), for their sincere efforts.

I am very thankful to Dr. Panagiotis Karampelas and Prof. Dr. Jan Jürjens, who took their time to review the thesis and provided their valuable comments. I greatly extend my gratitude for Dr. Nasir Tauheed, Dr. Hamid Abdul Basit and Dr. Arshad Ali Shahid for the final assessment of this thesis.

Last but not the least, I would like to thank my teachers and family for their support during this journey.
Table of Contents

TABLE OF CONTENTS ........................................................................................................ V
LIST OF PUBLICATIONS ................................................................................................ IX
LIST OF FIGURES ................................................................................................................ XI
LIST OF TABLES .................................................................................................................. XIII
LIST OF ALGORITHMS ...................................................................................................... XV
LIST OF ABBREVIATIONS AND NOTATIONS ................................................................ XVII
CHAPTER 1: INTRODUCTION .......................................................................................... 1
  1.1 BACKGROUND AND MOTIVATIONS ..................................................................... 1
  1.2 PROBLEM STATEMENT ......................................................................................... 2
  1.3 RESEARCH OVERVIEW ......................................................................................... 3
  1.4 THESIS OUTLINE .................................................................................................. 4
  1.5 THESIS CONTRIBUTIONS ..................................................................................... 4
  1.6 SUMMARY .............................................................................................................. 5
CHAPTER 2: LITERATURE REVIEW ............................................................................... 6
  2.1 INTRODUCTION ........................................................................................................ 6
  2.2 ISSUES IN DATA MANAGEMENT ......................................................................... 6
    2.2.1 Data Representation Issue ............................................................................ 7
    Meta-data Representation Languages ........................................................................ 7
    Schema Management Approaches .............................................................................. 9
    2.2.2 Data Discovery Issue ..................................................................................... 9
    Directory-based Approaches .................................................................................... 10
    Directory-less Approaches ..................................................................................... 10
    Cross-layer Approaches .......................................................................................... 11
    Miscellaneous Approaches ..................................................................................... 12
    2.2.3 Caching Issue ................................................................................................ 13
    Cache Replacement .................................................................................................. 13
    Consistency Management Approaches .................................................................... 15
    2.2.4 Other Issues .................................................................................................. 16
  2.3 DATA MANAGEMENT FRAMEWORKS ................................................................ 17
    2.3.1 Data Management in P2P File Sharing Systems and Sensor Networks .......... 17
    2.3.2 Data Management in MANET ...................................................................... 18
    Alfano and Manzalini (2012) .................................................................................... 18
    Ad-hoc InfoWare ...................................................................................................... 18
    AmbientDB ................................................................................................................ 18
    MoGATU .................................................................................................................... 19
    DRIVE ....................................................................................................................... 19
    CHaMeLeoN ............................................................................................................... 19
    CDMAN ..................................................................................................................... 19
    Other Frameworks ..................................................................................................... 20
  2.4 SUMMARY .............................................................................................................. 20
CHAPTER 3: ALADIN: A DATA MANAGEMENT FRAMEWORK ...............22

3.1 INTRODUCTION .............................................................................................................. 22
3.2 PROPOSED FRAMEWORK ........................................................................................... 22
  3.2.1 Logging Component ................................................................................................. 23
  3.2.2 Association Rules Mining (ARM) Component .......................................................... 24
  3.2.3 Data Discovery Component (DDC) ........................................................................ 25
  3.2.4 Data Representation Component (DRC) ............................................................... 25
  3.2.5 Query Processing Component (QPC) ..................................................................... 26
  3.2.6 Context Manager .................................................................................................... 26
  3.2.7 Cache Management Component (CM) ................................................................. 26
  3.2.8 Replication Component .......................................................................................... 27
  3.2.9 Security Component ............................................................................................... 27
3.3 SUMMARY ...................................................................................................................... 27

CHAPTER 4: A NETWORK LAYER DATA DISCOVERY APPROACH FOR MANET .................................................................28

4.1 INTRODUCTION .............................................................................................................. 28
4.2 DATA DISCOVERY COMPONENT ................................................................................ 28
  4.2.1 Proactive Component ............................................................................................... 29
    Advertiser Component .................................................................................................... 29
    Listener Component ....................................................................................................... 31
  4.2.2 Reactive Component ............................................................................................... 32
    AODV Based On-Demand Discovery Scheme .............................................................. 32
    DSR Based On-Demand Discovery Scheme .................................................................. 34
    Using Association Rules Mining to improve discovery process .................................. 36
4.3 IMPLEMENTATION DETAILS ...................................................................................... 37
  4.3.1 Datasets Generation ............................................................................................... 37
    Random Dataset ........................................................................................................... 37
    Correlated Dataset ....................................................................................................... 38
    YouTube Based Dataset ............................................................................................... 38
4.4 RESULTS ......................................................................................................................... 38
  4.4.1 Experiment Design ................................................................................................. 38
  4.4.2 Proactive Discovery Component ............................................................................. 39
    Analysis of Naïve (Non-Correlated) Advertisements .................................................... 40
    Analysis of Associative Advertisements ....................................................................... 41
  4.4.3 Reactive Data Discovery Component ..................................................................... 43
    Analysis of integration of data and route discovery operation ...................................... 43
    Analyzing the impact of piggybacking of data items .................................................... 45
4.5 DISCUSSIONS ON RESULTS ....................................................................................... 47
4.6 SUMMARY ...................................................................................................................... 49

CHAPTER 5: AN ONTOLOGY-BASED APPROACH TO DATA REPRESENTATION .................................................................................50

5.1 INTRODUCTION .............................................................................................................. 50
5.2 PROPOSED DATA REPRESENTATION COMPONENT (DRC) ....................................... 50
  5.2.1 Routing Table ......................................................................................................... 51
  5.2.2 Data Table .............................................................................................................. 51
  5.2.3 Ontology Component ............................................................................................ 52
5.3 PROPOSED ONTOLOGY .............................................................................................. 54
This page has been intentionally left blank
List of Publications

The work presented in this thesis is based on the following peer-reviewed book chapter, journal and conference publications:

**Book Chapters:**

**Journal Publications:**

**Conference Publications:**


List of Figures

Figure 2.1: Classification of Ontology Languages ........................................................... 8
Figure 3.1: Key Components of ALADIN ..................................................................... 23
Figure 3.2: Circular Linked List for maintaining session records ................................. 23
Figure 4.1: Block Diagram of ALADIN Data Discovery Component ........................... 29
Figure 4.2: Proactive Discovery Component of ALADIN ............................................. 29
Figure 4.3: AODV based Proactive Discovery ............................................................... 33
Figure 4.4: Analysis of Latency of DDC with Non-Correlated (Naïve) Advertisements ........................................................................................................................................ 40
Figure 4.5: Analysis of Hit Ratio of DDC with Non-Correlated (Naïve) Advertisements ........................................................................................................................................ 41
Figure 4.6: Analysis of Latency with Correlated (Associative) Advertisement .......... 42
Figure 4.7: Analysis of Hit Ratio with Correlated (Associative) Advertisement ........ 43
Figure 4.8: Analysis of Latency of AODV based Reactive Discovery Component Vs NOM ........................................................................................................................................ 44
Figure 4.9: Analysis of Latency of DSR based Reactive Discovery Component Vs NOM ........................................................................................................................................ 44
Figure 4.10: Analysis of Latency with the Piggybacking approach ............................. 45
Figure 4.11: Analysis of Hit Ratio with the Piggybacking approach ........................... 46
Figure 4.12: Analysis of Latency of Discovery Component of ALADIN Vs MoGATU ........................................................................................................................................ 47
Figure 4.13: Analysis of Hit Ratio of Discovery Component of ALADIN Vs MoGATU ........................................................................................................................................ 48
Figure 5.1: Block Diagram of ALADIN’s Data Representation Component ................ 51
Figure 5.2: ALADIN’s Multi-level Representation of Ontology ................................... 52
Figure 5.3: Proposed Software Ontology (Islam and Shaikh 2012) ............................... 54
Figure 5.4: Analysis of Latency with Semantic and Non-Semantic Discovery Component ........................................................................................................................................ 57
Figure 5.5: Analysis of Scalability of Multi-level Schema Management .................... 58
Figure 5.6: Analysis of Latency of Multi-level Schema Management ........................ 59
Figure 6.1: Block Diagram of ALADIN’s Caching Component .................................... 62
Figure 6.2: Candidate Set Generation using Proposed Cache Replacement Algorithm. 63
Figure 6.3 Analysis of Hit Ratio of Caching Component with AODV Routing protocol ........................................................................................................................................ 67
Figure 6.4 Analysis of Hit Ratio of Caching Component with DSR routing protocol ... 68
Figure 6.5 Analysis of Latency with proposed Consistency Manager .......................... 69
Figure 6.6 Analysis of Hit Ratio with proposed Consistency Manager ......................... 70
Figure A.1: The format of the Datasets used for Association Rules Mining ............... 83
Figure A.2: Number of Frequent itemsets generated by Association Rules Mining algorithm. ................................................................................................................................................................ 84
This page has been intentionally left blank
List of Tables

Table 2.1: Summary of various Ontology Languages .......................................................... 8
Table 2.2: A Summary of Data Management Issues in MANET ...................................... 16
Table 2.3: A Summary of Data Management Frameworks for MANET ......................... 21
Table 4.1: Simulation Parameters for experiments analyzing Data Discovery Component .............................................................. 37
Table 4.2: Correlation Matrix based on YouTube video requests of a campus’ users ... 39
Table 4.3: Summary of results of experiments analyzing Data Discovery Component ... 48
Table 5.1: Routing Table maintained by nodes .............................................................. 51
Table 5.2: Data Table maintained by nodes .................................................................... 52
Table 5.3: Simulation Parameters for experiments analyzing Data Representation Component ...................................................................................... 56
Table 6.1: Simulation Parameters for experiments analyzing Caching Component ...... 66
This page has been intentionally left blank
List of Algorithms

Algorithm 3.1: ALADIN’s Logging Component ................................................................. 24
Algorithm 4.1: ALADIN’s Advertiser Component ........................................................... 30
Algorithm 4.2: Advertisement in Correlated Mode......................................................... 31
Algorithm 4.3: Advertisement in Non-Correlated Mode.................................................. 31
Algorithm 4.4: ALADIN’s Listener Component............................................................... 32
Algorithm 4.5: ALADIN’s AODV based Reactive Discovery Component ................... 34
Algorithm 4.6: ALADIN’s DSR based Reactive Discovery Component ....................... 35
Algorithm 4.7: Piggybacking of potential future requests by ALADIN’s Discovery Component......................................................................................................................... 36
Algorithm 5.1: ALADIN’s Multilevel Ontology Representation Component ............... 53
Algorithm 6.1: ALADIN’s Correlation based Cache Replacement Algorithm.............. 64
Algorithm 6.2: ALADIN’s Network Layer Consistency Manager Component.......... 65
List of Abbreviations and Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊕ ⊕</td>
<td>An operation to accumulate or store a data advertisement in local data table.</td>
</tr>
<tr>
<td>( \alpha_t )</td>
<td>A data management operation applied over some data.</td>
</tr>
<tr>
<td>( \Sigma )</td>
<td>The alphabet over which data is defined.</td>
</tr>
<tr>
<td>( \epsilon_a )</td>
<td>A correlation coefficient threshold. Two data items are advertised together if the correlation between these data items falls above this threshold.</td>
</tr>
<tr>
<td>( \epsilon_c )</td>
<td>A correlation coefficient threshold. A value above this threshold signifies that a data item should not be picked as victim during cache replacement.</td>
</tr>
<tr>
<td>( \epsilon_l )</td>
<td>A threshold value used during generation of a correlated request session. A request is picked from candidate set ( C' ) probabilistically if a randomly generated number is greater than this value.</td>
</tr>
<tr>
<td>( \epsilon_r )</td>
<td>A correlation coefficient threshold used during replication. A value above this threshold signifies that a data item should be replicated.</td>
</tr>
<tr>
<td>( \epsilon_{s1} )</td>
<td>A correlation coefficient threshold used by security component. A value below this threshold signifies that a data request is abnormal.</td>
</tr>
<tr>
<td>( \epsilon_{s2} )</td>
<td>A correlation coefficient threshold used by security component. A value above this threshold signifies that a session is abnormal.</td>
</tr>
<tr>
<td>( \rho )</td>
<td>Correlation between any two data items.</td>
</tr>
<tr>
<td>( a )</td>
<td>A data item being advertised by advertisement component.</td>
</tr>
<tr>
<td>( a' )</td>
<td>A data item correlated to an advertised data item.</td>
</tr>
<tr>
<td><strong>ALADIN</strong></td>
<td>Association Rules Based Network Layer Data and Information Management System, the proposed framework.</td>
</tr>
<tr>
<td><strong>AODV</strong></td>
<td>Ad hoc On-Demand Distance Vector Routing protocol proposed by IETF for ad hoc network.</td>
</tr>
<tr>
<td>ARM</td>
<td>Association Rules Mining component of proposed framework.</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>$c_i$</td>
<td>A cache slot used for temporarily storing a particular data item.</td>
</tr>
<tr>
<td>$C$</td>
<td>The cache available to store data items temporarily on nodes.</td>
</tr>
<tr>
<td>$C'$</td>
<td>A candidate set used to generate correlated request sessions. The data requests floated during simulation are selected from candidate set.</td>
</tr>
<tr>
<td>CDG</td>
<td>Correlated Dataset Generator that generates correlated request session during simulation.</td>
</tr>
<tr>
<td>CM</td>
<td>Cache Management component of proposed framework.</td>
</tr>
<tr>
<td>DDC</td>
<td>Data Discovery Component of proposed framework.</td>
</tr>
<tr>
<td>DRC</td>
<td>Data Representation Component of proposed framework.</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic Source Routing protocol proposed by IETF for ad hoc network.</td>
</tr>
<tr>
<td>$d$</td>
<td>A data item enfolding some information.</td>
</tr>
<tr>
<td>$d'$</td>
<td>A data item correlated to another data item $d$.</td>
</tr>
<tr>
<td>$d_c$</td>
<td>Deviation count value that specifies how much a session has deviated from norm.</td>
</tr>
<tr>
<td>$D$</td>
<td>The set of data available over the network.</td>
</tr>
<tr>
<td>$D'$</td>
<td>A subset of data $D$ to be represented.</td>
</tr>
<tr>
<td>$D''$</td>
<td>The data represented in a particular format.</td>
</tr>
<tr>
<td>$D_i$</td>
<td>Data items available at a particular node $i$.</td>
</tr>
<tr>
<td>$D_r$</td>
<td>The set of data items to be replicated.</td>
</tr>
<tr>
<td>$f$</td>
<td>A function that quantifies a cache slot.</td>
</tr>
<tr>
<td>$f_c$</td>
<td>Cache replacement operation to pick a candidate slot from the cache.</td>
</tr>
<tr>
<td>$f_D$</td>
<td>The data management operation to discover a data item on the network.</td>
</tr>
<tr>
<td>$f_{ij}$</td>
<td>A frequent item generated by mining algorithm.</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>$f_R$</td>
<td>A data representation operation.</td>
</tr>
<tr>
<td>$F$</td>
<td>A set of all frequent itemsets generated by mining algorithm.</td>
</tr>
<tr>
<td>$F_i$</td>
<td>A frequent itemset generated by mining algorithm.</td>
</tr>
<tr>
<td>$h_i$</td>
<td>A discovery request floated by a node.</td>
</tr>
<tr>
<td>$h_{i,t}$</td>
<td>The time when a discovery request is issued.</td>
</tr>
<tr>
<td>$H_i$</td>
<td>A session that comprises a set of discovery requests.</td>
</tr>
<tr>
<td>IDS</td>
<td>Intrusion Detection System, one of the approaches to secure a network.</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Ad hoc Network, an infrastructure-less network formed between a set of nodes.</td>
</tr>
<tr>
<td>$M$</td>
<td>A correlation Matrix used to generate correlated request sessions.</td>
</tr>
<tr>
<td>$n$</td>
<td>A node of the network.</td>
</tr>
<tr>
<td>$N$</td>
<td>The set of nodes in the network.</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language, an ontology language used for semantic representation.</td>
</tr>
<tr>
<td>$Q$</td>
<td>Quality of Service (QoS) attribute of the data.</td>
</tr>
<tr>
<td>QPC</td>
<td>Query Processing Component of proposed framework.</td>
</tr>
<tr>
<td>$r_i$</td>
<td>A replica item.</td>
</tr>
<tr>
<td>$R_d$</td>
<td>A set of cache items correlated to a newly arrived item $d$ to be cached.</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Format, an ontology language used for semantic data representation.</td>
</tr>
<tr>
<td>$S_d$</td>
<td>A residual set specifying the set of data items non-correlated to a newly arrived data item $d$ to be cached.</td>
</tr>
<tr>
<td>SADV</td>
<td>A data advertisement message.</td>
</tr>
<tr>
<td>SREP</td>
<td>A data reply message generated in response of SREQ.</td>
</tr>
<tr>
<td>SREQ</td>
<td>A data request message.</td>
</tr>
<tr>
<td>VANET</td>
<td>Vehicular Ad hoc Network, a special type of ad hoc network formed between vehicles.</td>
</tr>
</tbody>
</table>
This page has been intentionally left blank
Chapter 1

Introduction

1.1 Background and Motivations

Pervasive Computing promotes the integration of technology into human environment, thus facilitating the users to seamlessly exploit the computing capabilities, anytime and anywhere. The technological progressions during last few decades have enabled the pervasive phenomena where a number of mobile devices organizes themselves without any infrastructure support to form Mobile Ad hoc Network (MANET) (Gehrke and Maddeb 2004; IETF 2013). This flexible networking model provides opportunities for several useful and innovative applications. Examples of such applications are disaster recovery, military operations and intelligent transportation etc. (Wang and Li 2009; Ghosekar, Katkar, and Ghorpade 2010).

MANETs are characterized by mobile, resource limited, unreliable and heterogeneous nodes; low capacity, asymmetric and unstable links; and dynamic, infrastructure-less and hostile environment etc. These unique characteristics call for intensive study on a range of research problems. Among them, data management is gaining significance during last few years. Data management is the set of mechanisms and procedures employed for ensuring timely data access to the consumers (Perich et al. 2004; Perich, Joshi, and Chirkova 2006; Denko et al. 2009). The management of data in MANET is considered a complicated job due to the new challenges that arise in the unique settings of MANET. There is the huge amount of data dwelling among the autonomous mobile devices. These devices have finite memory and processing capabilities. In addition, devices have the independence to employ their own mechanisms to maintain data. Therefore, data management in MANET requires new mechanisms to address various types of issues. To elaborate further, consider the following motivating scenario:

*In an ad hoc network established between vehicles on the road, a collision between few cars has encountered. A particular host in the network...*
realizes this situation and attempts to autonomously find out nearby hospitals for emergency facilities. The host prepares a query for nearest hospitals having facilities for trauma centers, vacant rooms and availability of the doctors etc. Upon the discovery of a specific hospital, the injured persons are starting to be transferred to the hospitals.

Being smart enough, the host also detects the availability of parking space where the vehicles can be parked. The patients are thus examined by doctors and advised the corresponding medicines. Consequently, a discovery request for location of pharmacy is floated with the availability status of desired medicines. The medicines are thus ordered and online payments are made. The host also realizes the probable need for pharmacy in near future and thus silently records the whereabouts of the pharmacy.

The above scenario illustrates the potential of MANET in rescue scenarios. It also highlights different types of data management issues. For example, a number of queries are issued for details about nearby hospitals, parking space and location of pharmacy. This requires the discovery of provider nodes on the network that can answer these queries. A data representation scheme is also desired. The set of nodes interacting with other can be strangers to each other. These nodes can employ their own mechanisms for storing and maintaining data. The issues of interoperability among the node can arise. So, the data representation scheme must provide mechanisms for syntactic and semantic interoperability among the nodes.

Once the data items are discovered, a number of post discovery steps are also performed. To ensure that the discovered information can be utilized in future, caching can be performed. Considering the small memory and the data of different nature available on the MANET, it becomes quite challenging to perform cache replacement in MANET. Additionally, maintaining the consistency of the discovered data also presents a challenge because of the sharp changes in MANET. Finally the problems of context management, query management and data security also require considerations to ensure the proper management of data in MANET.

1.2 Problem Statement
There have been some frameworks proposed in literature for addressing the issues of data management in MANET. After study of these frameworks, it is found out that only few frameworks deal with the issues of data management completely (Xu and Wolfson 2005; Perich, Joshi, and Chirkova 2006). This work fills the gap by proposing a complete data management framework for MANET called ALADIN. ALADIN contributes by providing a cross-layer approach for data discovery based on the algorithm proposed in (Torres and Garcia-Macias 2004). It is observed that inherent relationships among data items can be useful for improving the performance of data management. ALADIN uses the data correlation for advertising data items to neighbors. It also uses piggybacking of correlated information during generation of discovery response. Based on the literature review, it can be said that this is the first research work in this direction. Finally, the employment of a multi-tier representation of semantic information addresses the interoperability problem of MANET. This approach saves the precious memory of the nodes by retaining fractional views of the overall schema.
1.3 Research Overview

The problem of data management has been studied for conventional systems for a while, but the concept leads to several challenges in MANET. ALADIN addresses the prime issues of data discovery, data representation and caching. For this purpose, ALADIN proposes three major components i.e. the data discovery component (DDC), data representation component (DRC) and cache management component (CM). This section provides an overview of the research problems pertaining to each of these data management issues and how ALADIN addresses them.

Because of the mobility of nodes and instability of links, the topology changes very quickly and leads to the spatiotemporal changes in the availability status of the data. Therefore, it becomes necessary to timely discover the data available at various hosts in the network. The ALADIN data discovery component (DDC) component adopts a cross-layer approach to discovery, in which the data discovery and routing operations are performed simultaneously. So, the desired data and routing details for accessing the provider are available in one pass. The DDC also exploits the correlation patterns present among data items for improving hit ratio of the consumer. These correlation patterns are discovered based on historical requests issued by consumers. The DDC uses correlation patterns in two ways. First, it periodically broadcasts the correlated data items together to neighboring nodes that can then store the received advertisements. Secondly, during data discovery, the data items correlated with requested item can be sent together with response. This ensures enhanced hit ratio of the data consumers.

Besides data discovery, another important issue that arises in MANET is interoperability problem. The lack of boundaries and open nature of MANET allow different types of nodes to become part of the network. These nodes can employ their own mechanisms to represent data. This can raise the issues of syntactic and semantic interoperability. In addition, there is an absence of dedicated nodes to store the metadata information about the data dwelling on the network. Hence, individual hosts of MANET are required to maintain the metadata locally. However, the use of monolithic schema for describing the data is certainly not viable due to resource constraints. Hence, a semantic and scalable data representation approach has been proposed. ALADIN adopts a scalable multilevel ontology based approach to describe the data using its data representation component (DRC). The individual host in the network maintains a partial view of the overall schema. This partial view is essential for understanding the semantic of the information lying at a particular host. The schema grows gracefully in on-demand fashion via discovery of ontology documents from the network.

The final issue addressed by ALADIN is cache management. Because of the unreliable nature of links, the hosts in MANET are required to cache and maintain a consistent state of information which is likely to be consumed in near future. The cache replacement algorithms proposed in past turns out to be naïve in MANET. ALADIN employs a correlation based strategy for cache management in MANET. As the queries floated by a consumer in a session are subjected to correlation patterns (present in data), CM prioritizes the data items based on these correlation patterns during cache replacement process. ALADIN also proposes a network layer consistency management component for ensuring the proper state of the cached data items. The component runs at the network layer to recognize topological changes in the network for maintaining the state of the cached data items.
To validate the proposed framework, a prototype implementation has been developed using JIST/SWANS simulator (Barr, Haas, and Renesse 2004). Three types of datasets have been used to generate data request sessions. Request sessions of various characteristics are generated based on these datasets. The first dataset is used to generate sessions of random data requests; while based on the second dataset, sessions of correlated data items are generated using a simulated correlated data set generator. The final dataset is a real dataset that is used to generate sessions consisting of request for correlated data items. Different experiments are performed with varying number of nodes, data items, routing protocols and mobility models. The latency, hit ratio and scalability of the various components are analyzed and compared against complementary solutions available in literature.

1.4 Thesis Outline
This thesis presents a data management framework for MANET. Chapter 2 begins the discussion with an overview of the pertinent literature in the problem area. As data management is a multipart job, hence the literature review talks about the work specifically in the area of data management as well as about those research proposals that target a specific issue of data management. The issues include data discovery, data representation and cache management. Since data discovery is a specific case of service discovery; the terms data discovery, resource discovery and service discovery are used interchangeably in the rest of the chapters of thesis. In Chapter 3, the proposed data management framework i.e. ALADIN is discussed. The solutions for data discovery, data representation and cache management issues are presented in Chapter 4, 5 and 6 respectively. A prototype implementation of the proposed framework, evaluation details and the results of experiments are also presented. The thesis is concluded in Chapter 7 with a discussion on the major contributions of this work, the limitations of the framework and future directions of research.

1.5 Thesis Contributions
The major contributions of this thesis are as follows:

- An extensive data management framework that deals with the major issues of data management.
- This thesis introduces a network layer, correlation based approach to data discovery in MANET. The novelty of this component includes: a) a hybrid approach to network layer data and route discovery b) exploitation of correlation information for improving the discovery process.
- An ontology-based, scalable multilevel scheme for data representation is presented. The scheme differs from traditional approaches by providing syntactic and semantic interoperability. It also retains fractional view of the overall schema and populates the schema on-demand using the discovery component.
- A network layer correlation based cache management scheme is proposed. It comprises: a) cache replacement component on the basis of correlation among data item b) consistency management component that maintains the integrity of data items based on the network layer events.
1.6 Summary
Data Management has been an important topic of research for mobile ad hoc networks. The mobility of nodes and data consumption on the move have given rise to different potential applications. At the same time, several new data management challenges have arisen. This chapter discusses those challenges and highlights the need for a comprehensive data management framework. Next chapter talks about the data management problem in detail.
Chapter 2

Literature Review

2.1 Introduction
The topic of Data Management has been studied extensively in traditional systems. However, there has been shortage of such a framework for MANET. This chapter provides a review of the literature on data management and forms the basis for the proposed framework. It first introduces data management and its key issues in MANET environments. Then currently available data management frameworks for MANET are reviewed. The chapter concludes with the discussion on the need for a novel data management framework for MANET.

2.2 Issues in Data Management
Data Management is a multipart activity that comprises a set of operations employed for managing data as a useful resource. If \( D = \{d_1, d_2, d_3, \ldots \} \) represents the set of data available on the network, \( \alpha \) signifies an operation on data, \( \Sigma \) denotes the alphabet on which data is defined and \( d_i \subseteq \Sigma^* \), then the process of data management can be expressed as:

\[
\text{datamanagement} = \{ \alpha_i(D_i) : D_i \subseteq D \}
\]  

Hence, data management comprises a groups of operations \( \alpha_1, \alpha_2, \alpha_3 \ldots \) performed on subset \( D_i \) of data \( D \). The problem of data management has been studied for traditional systems for a while. However, these traditional approaches can’t be applied directly in MANET. Traditionally, a mobile computing solution based on a client-server model or client-server-proxy model is used (Perich, Joshi, and Chirkova 2006). In the former approach, a mobile device accesses a powerful server machine for its data needs. Despite of its simplicity, this approach raises the issue of scalability as all the clients interact with sever directly. The server is responsible for connection management and transcoding (data conversion) issues. In the client-server-proxy approach, there is a proxy entity residing between client and server machines. The client asks the proxy for
its data and proxies can inquire the server for required information. This approach provides scalability, as the server is now relieved from tasks like connection management and transcoding issues etc.

Unlike these conventional approaches to mobile computing that are essentially based on a client-server model, MANET creates an additional model based on the interaction among the hosts. All the hosts are now peers i.e. producer as well as consumer of information. The advantage of this approach is that nodes can now satisfy their data needs directly by coordinating with peers (without any dependence on prior infrastructure support). However, it presents a host of new challenges arising due to nodes mobility, variations in data availability and heterogeneity etc.

In MANET, data management is not an atomic activity but encompasses range of sub-activities each of which are premeditated to address a particular issue. These issues are data representation, data discovery and caching etc. Following paragraphs discuss each of these issues in detail. A survey on various data management issues and frameworks has been provided in (Islam and Shaikh 2011a).

2.2.1 Data Representation Issue

Let \( D_i = \{d_{i1}, d_{i2}, d_{i3}, \ldots\} \) represents the data items available at a particular node \( i \) in the MANET, and \( D = D_1 \cup D_2 \cup D_3, \ldots \). Data representation can be defined as the mechanism to describe the data such that it can be identified, discovered and accessed by any local or remote consumer. Let \( f_R \) represents the function that represents a set of data items \( D' \subseteq D \) into a particular format \( D'' \in \Sigma^* \), then \( f_R \) can be expressed mathematically as:

\[
\begin{align*}
    f_R(D) &= \{D''|\exists f': f'(D'') = D'\} \\
    \text{Thus, the data representation is a function } f_R \text{ that epitomizes data } D' \text{ into another representation scheme } D''. \text{ There is an inverse function } f \text{ that returns the original data represented by } f_R. \text{ The data representation issue itself can be classified into two sub-issues as discussed below.}
\end{align*}
\]

Meta-data Representation Languages

A number of languages for meta-data representation have been proposed in literature. The classical approaches for data representation are based on identifiers, textual description, attributes or tree like structure for describing the data. For example, JINI uses interfaces and SDP uses identifiers for describing services (Guttman et al. 1999; Gribble et al. 2001). Similarly, SLP and NINJA employ attributes as service representation scheme (Sun Microsystems 2001; Gryazin 2006).

The classical schemes are however not adequate for MANET as the nodes in MANET are heterogeneous in nature. A number of ontology based approaches have thus been proposed in literature (Chakraborty et al. 2001; Chakraborty et al. 2002). These approaches can be categorized as Extensible Markup Language (XML) based and Non XML based languages as shown in Figure 2.1. The Non XML languages adopt their proprietary mechanism for representation and are not generally applicable for MANET. Examples of these languages are Cycl, Ontolingua, Unified Modeling Language (UML) and Web Service Modeling Ontology (WSMO) etc.
Table 2.1 provides the summary of XML based languages (Islam, Abbasi, and Shaikh 2010). Among them, Resource Description Format (RDF) and Web Ontology Language (OWL) are the most widely used language (Cardoso 2007).

Table 2.1: Summary of various Ontology Languages

<table>
<thead>
<tr>
<th>Language</th>
<th>W3C Support</th>
<th>Editors</th>
<th>Other Supporting Tools / Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHOE</td>
<td>×</td>
<td>RFEdit</td>
<td>PIQ SHOE Search</td>
</tr>
<tr>
<td>RDF/RDFS</td>
<td>√</td>
<td>Protégé, NeOnToolkit, DOE, pOWL, Swoop, TopBraid</td>
<td>Jena SPARQL, RD-QL</td>
</tr>
<tr>
<td>OIL</td>
<td>√</td>
<td>OilEd</td>
<td>Fact</td>
</tr>
<tr>
<td>DAML + OIL</td>
<td>√</td>
<td>OntoEdit, Ontolingua, WebODE, IsaViz</td>
<td>DQL, Pellet, Racer</td>
</tr>
<tr>
<td>OWL</td>
<td>√</td>
<td>Protégé, DOE, Altova Semantic Works, WebODE, Swoop, Morla</td>
<td>Jena, OWL-QL, Fact++, Pellet, RacerPro</td>
</tr>
</tbody>
</table>
Resource Description Format (RDF) describes the metadata about any domain in the form of triples (subject, property and object). Web Ontology Language (OWL) provides a rich set of features for expressing classes, properties, relationships, domain, ranges, cardinality restrictions and annotations etc. Most of the data management frameworks utilize OWL ontology to represent the data. For example, MoGATU recommends SOUPA i.e. Standard Ontology for Ubiquitous and Pervasive Applications. SOUPA is based on OWL to describe semantic information about the data items (Chen et al. 2004).

**Schema Management Approaches**

Another issue related to data representation is about maintaining an updated and consistent view of the local data items with respect to the global schema. In MoGATU, every reply to a data item is accompanied with the name of the ontology required to understand the data item (Chen et al. 2004). A query processing component can load the ontology document from the producer node, if unavailable on the local node. AmbientDB utilizes a relational data model for describing the data (Boncz and Treijtel 2004). A global certifying entity is assumed that keeps track of schema changes and provides bi-directional mappings between local and global schema. DRIVE proposed a unified data model for resources in applications related to transportation (Xu and Wolfson 2005). The data representation is based on a model for representing resources, types, relations and information exchanges among the nodes. The conceived model serves as a schema for data and resources available on the network.

The review of literature on data representation concludes that only few frameworks are available based on ontological data representation. These frameworks like MoGATU (Chen et al. 2004) and Ad hoc Infoware (Sanderson et al. 2007) don’t employ multi-level data representation. Also, the scalability of the representation approaches has not been addressed extensively by current frameworks. The proposed framework ALADIN uses an OWL based scalable approach for data representation. For schema maintenance, a core ontology document is maintained at local node. The core ontology provides generally used ontology concepts. There are sub-ontology documents for specialized applications, maintained by other nodes. These ontology documents can be loaded in on-demand fashion using the discovery component of ALADIN.

2.2.2 Data Discovery Issue

The discovery process can be defined as the actions performed for locating the owner of a data block considering the QoS parameters specified by the client. If $N$ is the set of nodes in the network, $d$ is the data to be discovered in the network with description $des$ and quality of service $Q$, then the discovery process can be expressed mathematically as follows:

$$f_D(d) = \{i \mid n_i \in N \text{ and } \exists j: (d_{ij} \in D_i \text{ and } d.des \approx d_{ij}.des \text{ and } d.Q \approx d_{ij}.Q)\} \quad (2.3)$$

The function $f_D$ accepts a data item $d$ to be discovered as an argument. It returns an index $i$ from the available nodes 0, 1, 2, …, $|N|$ such that node $n_i$ hosts a data item $d_{ij}$ whose description and QoS matches with the description and QoS of data item $d$.

The research on data discovery initially focused on devising discovery architectures and consequently two approaches emerged (Mian, Baldoni, and Beraldi 2009): directory-based approaches and directory-less approaches. The next section briefly discusses these two approaches. As the concept of cross-layer discovery has been greatly
emphasized in recent literature, later sections will discuss cross-layer approaches in detail. Finally, a number of miscellaneous researches on data discovery will be discussed. The schemes discussed below use the terms data, services and resources interchangeably.

**Directory-based Approaches**

The *directory-based approaches* are based on a set of dedicated nodes called directory servers that keeps the information about the services. This information includes location, QoS and description etc. about the services. A user agent can query directory servers for these resources. Examples of directory-based approaches are Service Location Protocol (SLP), Salutation, JINI and DReggie etc.

The SLP protocol is a lightweight discovery protocol proposed by Internet Engineering Task Force (IETF) for discovery of services in distributed environment (Guttman et al. 1999). The services are described by attributes and registered to directory agents. There are user agents that can inquire directory agent about the services. In case of no directory servers available on the network, user agent can also ask the service agent directly for the desired services. Similar concept has been proposed in Salutation that comprises a salutation manager for maintaining the services information (Miller and Pascoe 2000). The devices advertise their capabilities to salutation manager. The salutation manager can store this information and serve the requests of service clients.

JINI is an open standard proposed by Sun Microsystems for sharing and discovery of services in distributed systems (Sun Microsystems 2001). Any device upon joining network, registers to lookup server by providing various information. Services are then looked-up using Remote Method Invocation (RMI) protocol. An extension of JINI is DReggie, where a prolog based reasoning engine is attached and services are described using DARPA Agent Markup Language (DAML) (Chakraborty et al. 2001).

In recent literature, Jayapal and Vembu (2011) proposed an adaptive approach to service discovery based on the selection of a core node. The core node is elected based on eligibility factor that is calculated through various parameters (battery power and speed etc.). The core node has the responsibility of maintaining services of nodes in its own area and performs the discovery procedure on behalf of other nodes in its area.

**Directory-less Approaches**

The directory-based approaches are often unworkable for MANET setting due to the stringent requirement of dedicated nodes. In contrast, the *directory-less approaches* function by keeping the details of the resources locally in a catalogue called data, service or resource table. The nodes advertise their services regularly which are used by neighbors to update their tables. Following paragraphs summarize some of the discovery protocols.

NOM is a P2P approach based on reactive discovery of services (Doval and O’Mahony 2002). The nodes look for any incoming NOM message. If the message is a query initiate request, it is simply forwarded to neighboring nodes. If the message is a query, it is first checked locally. If any information is found locally, a reply is generated. In the later case, the message is propagated to neighboring nodes. The query reply message is travelled back to the originator in the similar fashion via propagation of reply to
neighboring nodes. To avoid the network overhead, a NOM message processed earlier or with an expired TTL value is not processed.

The Universal Plug and Play (UPnP) protocol is proposed by Microsoft for connectivity among small appliances in office environments (Microsoft 2000). The devices advertise their capabilities by means of XML messages. Other nodes can call the services using Simple Object Access Protocol (SOAP). Another protocol based on P2P advertisements is IBM’s Deapspace protocol where a service announcement message is sent to neighbors periodically (Nidd 2001). The announcement interval can be intelligently tuned by nodes based on the advertisements sent by other nodes.

Blue-tooth also proposed a service discovery protocol for P2P discovery in small environments (Gryazin 2006). It is based on a client-server based request-response paradigm. This protocol was further extended by Allia that uses the concept of alliances structured among nodes (Ratsimor et al. 2002). Group-based Service Discovery (GSD) protocol employs the concept of grouping among nodes (Chakraborty et al. 2002). The protocol uses DAML to specify the description of services and their groups. Konark is a P2P discovery protocol for m-commerce applications that attempts to accomplish device impendence by specifying the services using Web Service Definition Language (WSDL) (Helal et al. 2003). The protocol utilizes SOAP for consuming the services.

**Cross-layer Approaches**

In a cross-layer discovery approach, information across different layers of protocol stack is exploited for data discovery as well as for enhancing the performance of various networking operations. Among the various cross-layer approaches, there are proposals based on combining the routing and discovery layer (Cheng 2002; Koodli and Perkins 2003; Ma et al. 2003). Torres and Garcia-Macias (2004) proposed a network layer approach to service discovery in MANET via the extension of Ad hoc On-Demand Distance Vector (AODV) Routing algorithm. The service discovery request SREQ is circulated on the network through AODV algorithm such that a temporary reverse route is maintained by intermediate nodes. The SREQ is propagated until it reaches the node that knows of the service. The intermediate nodes then generate a service request reply SREP that is sent back to the source using the temporary reverse route maintained by intermediate nodes. While SREP is routed back to the source node, a forward route is erected. Consequently, the route to reach to the service provider is also discovered during the service discovery phenomenon. This improves the latency incurred in calling the service and enhances the latency of the discovery process. *Latency* can be defined as the time taken when a discovery request is issued to the time the discovery response has been received. This work extends the work of Torres and Garcia-Macias (2004) by adding a proactive advertisement module. This approach gives rise to improvement in the latency of the process as shown in Chapter 4.

An Optimized Link State Routing (OLSR) based discovery scheme has been proposed by Li and Lamont (2005). A new type of message called Service Location Extension (SLE) is proposed for service advertisement, query and response purposes. Queries can be sent to the directories and in the absence of directories, queries can be advertised. Athanaileas, Ververidis, and Polyzos (2007) proposed two service selection schemes after incorporating discovery process in AODV protocol. The first scheme is based on selecting the service whose provider is the least hop away and the second is based on choosing the provider with minimum energy level. Halkes, Baggio, and Langendoen
(2006) performed an evaluation of cross layer service discovery with flooding based discovery schemes.

SCAODV integrates the service discovery along with routing, claiming to reduce the traffic overhead and latency of the discovery process (Zhong et al. 2012). It is based on the broadcasting of SREQ until a node with the desired service is reached. A reply SREP is then generated and propagated towards the requesting node. Any node receiving the reply checks if it has the same service. It compares the effective time of SREP with time out of its service and makes the longer one, the time out of the service. SCAODV also proposes to use a trusted server to provide mutual authentication among the nodes.

Another way of cross-layer exploitation is the utilization of network event information for maintaining services information. In this direction, Kozat and Tassiulas (2004) proposed an approach which comprises two phases i.e. back bone management and distributed service discovery. During the first phase, a set of nodes are selected as back bone nodes. The paths are then found out between nodes to form a mesh. The backbone is continuously maintained with the topological changes. After the first phase, there are service broker nodes such that all the nodes in the network are maximum one-hop away from these nodes. During the discovery phase, the broker nodes are utilized for efficient dissemination of service discovery protocol messages.

Hong, Srinivasan, and Schulzrinne (2007) used the ZeroConf protocol to boost up the discovery process. The algorithm constantly examines the network interfaces for topological variations. Upon any network events, service advertisements are done aptly, thus cutting down the network traffic and accelerating the discovery process.

Pariselvam and Parvathi (2012) proposed a novel cross-layer solution based on Ant Colony Optimization. The proposal is based on swarm-based routing where routing is performed by interaction of forward and backward exploration agents (ants). The backward agents utilize the information gathered by forward agents during their way to destination. Using the swarm-based routing to advertise the service discovery request and replies, the service discovery operations are performed.

**Miscellaneous Approaches**

Besides the schemes discussed above, a number of noteworthy efforts have been put forwarded in miscellaneous directions. Lenders, May, and Plattner (2005) proposed a unique approach for discovery by developing an analogy between the environmental settings of MANET and electrostatic field. A mathematical model is devised based on this correspondence to discover services on the network. The services in the network correspond to positively charged particle and the requests for a particular service correspond to negatively charged bodies. Every network elements are assumed to calculate a potential value and the requests are disseminated towards the nodes with the highest potential values. Talwar, Venkataram, and Patnaik (2007) proposed a scheme where mobile nodes are employed to gather the specifications of the services prevailing on the network. These mobile nodes inquire the flanking nodes about the services hosted by them. Whenever a consumer desires a service, the mobile nodes are contacted to provide the details of the requested service. Mallah and Quintero (2009) proposed a lightweight and scalable scheme for discovery of services. It comprises three phases: back-bone formation, maintenance and registration/discovery phase. The backbone is
formed among a set of stable nodes. The stability of the nodes is calculated based on the parameters like node’s battery power, degree of connectivity and velocity etc. This gives rise to scalability into the discovery process. The maintenance phase keeps track of topological changes. It performs local maintenance in case of disconnection of a backbone node from the network. The backbones are responsible for maintaining and answering the discovery requests by other nodes. In the third phase, the registration of services and subsequent discovery operations are performed. The producer nodes register the services (they want to offer) to the backbone nodes. A consumer node propagates it service discovery request to a set of nodes in a fixed diameter. The closest backbone receiving this request will then generate a reply to this request.

Aguilera and Lopez-de-Ipina (2012) proposed a parameter based service discovery scheme where proactive component of every node periodically broadcasts inputs and outputs of parameters of services. The advertised parameters are then used during service discovery to reduce network traffic overhead.

This work proposes a cross-layer correlation based approach to data discovery. Based on the above discussions, it can be said that no major work has been done in utilizing the correlation among data for discovery purpose. The closest work in this direction is reported by Wang et al. (2010), and Bhole and Ranade (2012). Wang et al. (2010) analyzed the web usage log to determine relationships among web services in web-based environments. This scheme has been integrated into real business for composite web service discovery. The approach by Bhole and Ranade (2012) is based on our initial work reported in (Islam and Shaikh 2009), where association rules are exploited only for piggybacking of potentially useful services during the discovery process. The discovery component used by ALADIN is a step forward in this direction that provides cross-layer proactive and reactive discovery of data items as well as corresponding routes.

### 2.2.3 Caching Issue

After the discovery of the data, caching is done to store the remote data locally such that the time to access the data is improved. In addition, caching ensures the working of the nodes in offline mode of operations. There are two major sub-issues in caching. The first issue is to decide about which data item to be cached. The second issue is to maintain a consistent state of cached information. These two sub-issues are discussed below.

**Cache Replacement**

Let \( C = \{c_1, c_2, c_3, \ldots\} \) denotes the entries of a cache. Suppose a data item \( d \) is to be placed in the cache. Cache replacement problem is then defined as finding a victim slot \( c_d \) in the cache based on some optimization function. Let \( Z \) represents the set of integers and \( f: C \rightarrow Z \) is the optimization function, then cache replacement problem \( f_R \) can be written as:

\[
f_c(d) = \{c_d: f(c_d) = \min (f(c_1), f(c_2), f(c_3), \ldots)\}
\]  

(2.4)

The function \( f \) quantifies the cache slots \( c_1, c_2, c_3, \ldots \) by assigning it an integer value. The cache replacement function \( f_c \) returns a cache slot which has the minimum quantified value.
Different types of cache replacement algorithms have been proposed for traditional systems (Stallings 2009). These are Least Recently Used (LRU), Most Recently Used (MRU) and First in First out (FIFO) etc. In the LRU scheme, the cache removes the least recently used item from the cache where as the MRU algorithms discard most recently used items from the cache. In the FIFO algorithm, the items are removed from the cache in the order they arrived in to the cache. These traditional schemes are not useful for MANET because of their naïve nature (Perich et al. 2004). The researchers are looking into newer algorithms specifically targeted for MANET.

Different research proposals have been put forwarded in miscellaneous directions. Lau, Kumar and Venkatesh (2002) proposed a caching scheme for real-time continuous media applications via an application manager component that uses RTSP protocol to communicate with peer components. Perich et al. (2002) introduced a priority based profile driven caching scheme in which the nodes designate the maximum priority to local nodes, then medium priority to remote nodes and least priority to queries that have been answered previously. In addition, the profiles of the nodes are also kept into consideration for caching the data for future usage.

A set of cooperative caching schemes have been proposed in literature. In these schemes different nodes work in cooperation for deciding the caching data and its location. For example, Cao, Yin and Das (2004) facilitated the nodes in the network to cooperate with each other to reckon about the caching site. The authors proposed three types of schemes. In the CachePath scheme, the nodes in transit between a consumer and producer can cache the routing path towards the data. In the CacheData scheme, the intermediate nodes perform caching of actual data. There is another algorithm called HybridCache that is the combination of the above two schemes.

CoCa is a cooperative caching scheme based on identifying the low activity (LAM) and high activity clients (HAM) in a P2P network (Chow, Leong, and Chan 2004a). The suitable data is hoarded on LAM nodes for easy access by HAM clients. GroCoa extends the work of CoCa by establishing tightly couple group (TCG) via a clustering algorithm (Chow, Leong, and Chan 2004b). These TCGs are characterized by similar mobility signatures and data affinities. The hosts in the same TCGs collaborate with each other for maintaining their caches.

Zone Cooperative Caching is based on construction of zones among a set of nodes that are at 1-hop distance from each other (Chand, Joshi, and Misra 2007). A metric called Least Utility Value (LUV) has been proposed for determining the cache items to be placed in to the cache. LUV is computed based on various factors like distance between the producer and consumers, the size of data items and the access probability of data items etc. Yang and Hurson (2007) introduced a semantic caching for MANET based on Bayesian Probability. A non-flooding based query processing and QoS aware consistency management scheme have been proffered.

Shanmugavadi and Madheswaran (2010) proposed a group based caching scheme. In this scheme a set of nodes forms a group. Any incoming data item is placed into local cache if enough space is available. Otherwise, the node queries its group members for available space and places the data item in the corresponding group node.
Kuppusamy, Thirunavukkarasu, and Kalaavathi (2012) also proposed a cooperative caching scheme based on clusters. The whole network is divided into clusters with cluster head selected based on the power level and nodes’ connectivity. There is a local cache table that contains information about the cached items in the cluster. A global cache table keeps the cached items over the adjacent clusters. A data item request is first searched in to local cache table. If it is not present in local cache, the global cache table is consulted and the corresponding client is then contacted to fetch a copy of data item.

**Consistency Management Approaches**

The mobility in MANET leads to frequent changes in the state information about the nodes and their data items. Hence, it is essential to maintain the consistency of data items. One of the approaches is to propagate any changes by means of distributed events. This approach has been adopted by early distributed systems like Microsoft UPnP (Microsoft 2000). In cooperative caching, any forwarding node can also investigate the data item to update the state of data item locally in adaptive manner (Chand, Joshi, and Misra 2007). One can also maintain a TTL value that when expires, calls for update of the data state and resets the TTL (Shanmugavadivu and Madheswaran 2010). An extension to this approach has been proposed in (Fatima and Khader 2012). They introduced Extended Adaptive TTL (Ex-ATTL) algorithm in which 1-hop neighbors maintain a cache invalidation table for cache invalidation. The invalidation table contains a cached item, its TTL value and creation time. For every data item request, the algorithm checks if it contains the data item. It then calculates the difference between request generation time and the data item creation time. If the result is greater than TTL, the cached item is considered stale and the request is forwarded to find an updated copy of the item. The source of the data item replies to this request with new a TTL value. The retrieved data item is then cached as well as this information is broadcasted to one-hop neighbors.

Different consistency management approaches have been adopted by various data management frameworks. Perich et al. (2003) proposed a neighbor-hood consistent transaction management system based on epidemic voting in which a number of peers work together to perform transaction. Xu and Wolfson (2005) proposed a transaction management component that maintains integrity by means of log records maintained for each transaction. Martin and Demeur (2008) presented various types of consistency management approaches. A pessimistic model can be employed for sequential consistency, optimistic model for emergency conditions and a hybrid model for confined areas.

This work proposes a correlation based cache replacement and a network layer consistency management solution. The replacement scheme gives priority to cached items related to current item. The consistency manager uses the underlying routing protocol to disseminate the topological and other changes. Based on the analysis of the various caching solutions proposed in literature, it can be said that proposed approach provides a novel solution for cache management. We don’t find any solution based on exploiting correlation for cache replacement and only few consistency management solutions are based on utilizing cross-layer information. For example MoGATU uses cross-layer operation for a neighborhood consistent transaction management component and profile driven caching (Perich et al. 2003).
2.2.4 Other Issues
Besides the issues discussed above, data management in MANET also requires dealing with various other issues. Following paragraphs discuss briefly these issues for the sake of completeness. Table 2.2 summarizes the various issues of data management and existing solutions available for these issues.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Challenges</th>
<th>Existing Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Representation</td>
<td>Heterogeneous data sources</td>
<td>Ontology (Chen et al. 2004)</td>
</tr>
<tr>
<td></td>
<td>Finite resources</td>
<td>Attributes (Guttman et al. 1999)</td>
</tr>
<tr>
<td></td>
<td>Absence of standard data representation scheme</td>
<td></td>
</tr>
<tr>
<td>Data Discovery</td>
<td>Lack of universal catalog</td>
<td>Directory based solutions (Jayapal and Vembu 2011)</td>
</tr>
<tr>
<td></td>
<td>Spatiotemporal variations</td>
<td>Directory-less solutions (Gryazin 2006)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cross layer solutions (Pariselvam and Parvathi 2012)</td>
</tr>
<tr>
<td>Caching</td>
<td>Limited resources</td>
<td>Profile driven (Perich et al. 2002)</td>
</tr>
<tr>
<td></td>
<td>Diversity of data sources</td>
<td>Cooperative (Chand, Joshi, and Misra 2007)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Semantic (Yang and Hurson 2007)</td>
</tr>
<tr>
<td>Replication</td>
<td>Network partitioning</td>
<td>Data access frequency based (Hara 2001)</td>
</tr>
<tr>
<td></td>
<td>Power</td>
<td>Correlation based (Hara, Murakami, and Nishio 2004)</td>
</tr>
<tr>
<td></td>
<td>Mobility</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Link failures</td>
<td></td>
</tr>
<tr>
<td>Query Processing</td>
<td>Context awareness</td>
<td>Profile driven querying (Perich et al. 2001; Perich et al. 2006)</td>
</tr>
<tr>
<td>and Optimization</td>
<td>Resource limitations</td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>Open nature</td>
<td>Intrusion detection (Taneja and Kush 2012)</td>
</tr>
<tr>
<td></td>
<td>Lack of security infrastructure</td>
<td>Cryptography (Xiong and Tang 2011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Secure routing (Karlsson 2012)</td>
</tr>
</tbody>
</table>

Replication is an important issue in MANET where a node stores a block of information nearest to the consumer. In MANET it becomes very challenging to decide about the location where a data item has to be replicated. The replication strategies currently available for MANET mainly stressed on the conservation of power, network partitioning and real time transactions (Padmanabhan et al. 2008). Hara (2001) proposed three strategies for data replication namely Static Access Frequency (SAF), Dynamic Access Frequency and Neighborhood (DAFN), and Dynamic Connectivity-based Grouping (DCG). These approaches are based on how frequently the data is consumed and the status of connections. Hara, Murakami, and Nishio (2004) extended these schemes further by considering the correlation among data items for data replication. REPlication in Dense MANETs (REDMAN) is a middleware that identifies dense
regions in MANET and then replicate and maintain data blocks in these regions in a scalable way (Bellavista, Corradi, and Magistretti 2005).

Query Processing and Optimization is another important data management issue. The output of queries in MANET is dependent on the time, space and other contextual information. Hence, new techniques for query processing and optimization are desired. Representative work has been presented in (Perich et al. 2001; Perich et al. 2006). As the MANET is formed among a set of hostile nodes in possibly exotic environment, the nodes suffer from a range of security issues. For example, a malicious node can launch a Denial of Service (DOS) attack by incessantly inquiring for a particular type of service. Similarly, an adversarial node can strike at the routing layers by advertising false routes. The wireless nature of MANET also leads to other security vulnerabilities that can compromise data privacy. Various intrusion detection systems, secure routing approaches and cryptographic solutions have been proposed for MANET (Xiong and Tang 2011; Karlsson 2012; Taneja and Kush 2012).

Besides the solutions for representation, discovery and caching (discussed earlier), ALADIN also outlines solutions for replication and security management. These solutions are based on correlation patterns that are discovered by the framework during other data management operations.

2.3 Data Management Frameworks

The subject of data management has been under study over a long period of time. This includes the work on conventional file based system, relational database management systems and distributed databases etc. (Ricardo 1990). A number of frameworks for P2P file sharing systems and sensor networks have been proposed during last few years. In the domain of MANET, the work is at infancy and only a small number of frameworks have been proposed.

2.3.1 Data Management in P2P File Sharing Systems and Sensor Networks

A great deal of research toil has been devoted for data management mechanisms for the P2P file sharing systems. Different file sharing systems i.e. Gnutella, Chord and Gridella have thus evolved (Girdzijauskas 2009). Gnutella is a decentralized P2P system that employs a flooding-based query routing protocol and a hashing scheme to satisfy the user queries. Chord is a robust and distributed peer-to-peer system that adapts itself gracefully with the topological variations in the network. It employs a consistent hashing strategy to discover the hosts maintaining the requested information on the network. Gridella is P2P grid information management system that extends Gnutella by proposing numerous novel techniques like directed search for improving the search time and bandwidth consumption.

Sensor Network is a special type of ad hoc network established among a group of inert sensor hosts. Due to its strong association with MANET, it is imperative to quote the data management schemes proposed for sensor networks. TinyDB, Direct Diffusion and Cougar were the initial systems proposed for sensor networks (Yick, Mukherjee, and Ghosal 2008). Among other solutions, Garofalakis et al. (2006) proposed Data Furnace, a data management infrastructure for sensor networks based on probabilistic models. A probabilistic model is employed to discover high level semantic events based on the low level uncertain events of the environment.
StonesDB is a two layered architecture comprising sensor nodes and resource rich proxies running at separate tiers on the network (Diao et al. 2007). There is a database executing natively on every sensor nodes and is equipped with a query processing component, a data summarization component and an energy efficient storage substrate. A distributed database management layer is running at the proxies and provides an abstraction over the local databases running at the various nodes of the sensors.

Nittel et al. (2007) proposed an ad hoc sensor network based platform for ocean monitoring. The project is based on realizing the coarseness of the information rampant in traditional ocean monitoring system. By deploying a fleet of drifters in ad hoc configuration based on the region density, the project investigates the issues of routing, energy consumption, long term connectivity and sensing uniformity etc.

2.3.2 Data Management in MANET
The problem of data management becomes a taxing job in dynamic environment of MANET. The issues arise because of mobility, heterogeneity, reliability and security etc. Following are some of the frameworks proposed for MANET.

Alfano and Manzalini (2012)
Alfano and Manzalini (2012) proposed a method for operations of a node in a communication network. The communicating entities can be cell phones, laptops, sensors, PDAs and iPods etc. The proposal comprises a method for erecting a set of information provider nodes. The information provider nodes are entities that contain some data in its secondary storage for possible consumption. The project only talks about data discovery issue; details about other issues of data management are not provided.

Ad-hoc InfoWare
Sanderson, Goebel, and Munthe-Kass (2004) and Sanderson et al. (2007) proposed a knowledge management framework for rescue scenarios based on sparse MANET. The main objective of the project is to ensure seamless exchange of data among the nodes organized as a network during rescue work. The framework discusses various components for knowledge management, distributed event notification service, resource management and security manager. However, no operational details for these frameworks are provided nor any evaluations are reported.

The project suggests that knowledge management should comprise meta-data handling components and supporting tools. The former includes components for data dictionary, profile and context management while the later includes query processing component. For event notification, publish-subscribe model has been proposed. A dissemination based approach has been suggested for resource discovery. A neighborhood awareness strategy is also discussed for predicting the availability of a node in near future. This saves the bandwidth and the energy of the node by anticipating the availability of resources in near future, calculating soft-state intervals for data states and possible replication of the information etc.

AmbientDB
The AmbientDB is a middleware for the realization of ambient intelligence (Boncz 2003; Boncz and Treijtel 2004; Fontijn and Boncz 2004). The ambient intelligence is defined as the approach to provide personalized multimedia contents in a ubiquitous
environment. The AmbientDB project provides a relational database abstraction over a collection of ad hoc network nodes connected via volatile links. The project uses Chord for efficient indexed lookup and performs a three level query translation i.e. abstract algebra, concrete algebra and dataflow algebra. For resolving synchronization issues, instead of providing strict consistency (e.g. two-phase locking), a formalism is provided via which applications can themselves devise synchronization and conflict resolution strategies. To address syntactic and semantic heterogeneity issues, AmbientDB proposes a set of basic operations (for applying and sharing of mappings between local and global schema). In addition, a certifying entity is recommended that keeps track of schema changes and maintaining the corresponding mappings.

**MoGATU**
MoGATU is a data management system for pervasive computing environment (Perich et al. 2004; Perich, Joshi, and Chirkova 2006). There are information managers, information providers and information consumers available at different nodes on the network. An OWL-S based model has been proposed for data representation and query processing. A collaborative query processing component has been proposed that joins the results of a query executed on different information managers. MoGATU exploits the profiles of the consumers for cache replacement. Two caching schemes, LRU + P and MRU + P have been proposed. MoGATU also proposes a neighborhood consistency management for addressing the transactional issues associated with data. For ensuring the security goals, MoGATU maintains a trust value for its peers depending upon the past interactions. In addition, every node maintains a belief value based on the information it has about a particular node. While interacting with the other nodes, a particular node uses these trust and belief values to determine the reliability of the responses generated against its queries.

**DRIVE**
Dissemination of Resource Information in Vehicular Environments (DRIVE) is a software solution for data management in dynamic and mobile P2P networks (Xu, Ouksel, and Wolfson 2004; Sistla, Wolfson, and Xu 2005; Xu and Wolfson 2005; Zhong, Xu, and Wolfson 2008). It embraces a unified model for data representation and an opportunistic approach for dissemination of information among the hosts. The philosophy behind the opportunistic theory is to deliver information to peers as soon as they meet each other. An SQL like querying language is proposed for information acquisition from native database and an economic model based on virtual currencies has been suggested for billing and remittance. DRIVE also proposed a transactional model to cope up with the data integrity issues in the network.

**CHaMeLeoN**
CHaMeLeoN is an attempt towards exchange of real time and continuous media for heterogeneous network (Ghandeharizadeh et al. 2006). The project aspires to cope with the diverse application requirements, device heterogeneity and streaming of continuous media. To accede to this objective, a layered approach is proposed. The framework accepts various parameters (mobility, resource constraints and QoS etc.) and then employs adaptive algorithms for various data management operations.

**CDMAN**
CDMAN is a proposal for collaborative data management in MANET (Martin and Demeur 2008). It stands for Collaborative Data Management in Ad hoc Network. The
work comprises a tree-based XML data structure for storing data segments in a structured format, thus handling the resource limitations. A replication scheme is proposed based on erection of collaborative groups considering the interests of the users and the probability of disappearance of group members from the network. CDMAN introduces three types of models for confronting the data consistency issues. The nodes can utilize a pessimistic model for affirming sequential consistency, an optimistic model for exigent situations or a hybrid model for small and stable groups in confined areas.

Other Frameworks

Wu and Wu (2006) proposed a data management framework for location based services in mobile environment. The authors proposed a cost model and analyzed the proposed mechanisms for caching and replication under various mobility models. Yamasaki et al. (2007) proposed MoDA, a mobility ware data management scheme. The trajectory of the nodes is utilized to determine the replication strategy. The whole network is divided into square grids. The contents associated with a grid are maintained on a mobile node resident in the particular grid.

Query and Disseminate under Global View is a query propagation and dissemination protocol for multidimensional data applications (Michalarias and Becker 2007). The nodes in the network share their views (in the form of data cube) whenever they query for some information. The view contains the information about the node's own status as well as the status of other nodes. If a node receives the view advertisement from some other node, the view merge operation is performed to get a complete view. By employing this P2P broadcasting strategy, the nodes divulge their native information with each other and allay their data requisitions.

Denko, Shakshuki, and Malik (2009) proposed a cross layer data management system based on cooperative caching. The cross layer information is maintained in a data structure and pre-fetching and clustering operations are performed to maintain caches. The network layer provides information about the network traffic status thus promoting timely pre-fetching of cache items. Secondly, the cached item ids are used by network layer to eliminate propagation of request if the requested item is available on the forwarding node.

2.4 Summary

The analytical study of the various data management approaches outlined in literature leads to the conclusion that only a trivial number of data management frameworks are currently available for MANET. Most of the frameworks have been targeted for a specific type of application. There are only few frameworks proposed for general purpose networks. The currently available frameworks focused on particular issues of data management as shown in Table 2.3.

Only a few frameworks attempt to cover the entire issues of data management problem. Therefore, this work proposes ALADIN, a complete solution for data management. The framework is based on semantic data representation, cross-layer design and inherent relationships among data. Among the existing data management solutions, none of them take data correlation into account for confronting the issues of data management. There are only few solutions based on inter-layer communications and ontology-based data representation. Hence, ALADIN provides a distinctive solution in this direction.
### Table 2.3: A Summary of Data Management Frameworks for MANET

<table>
<thead>
<tr>
<th>Framework</th>
<th>Semantic Data Representation</th>
<th>Discovery Component</th>
<th>Cache replacement</th>
<th>Consistency management</th>
<th>Replication</th>
<th>Query Processing and Optimization</th>
<th>Security</th>
<th>Cross-layering</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfano and Manzalini (2012)</td>
<td>×</td>
<td>√</td>
<td>*</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Ad-hoc Infoware</td>
<td></td>
<td>*</td>
<td>×</td>
<td>*</td>
<td>×</td>
<td>*</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>AmbientDB</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>MoGATU</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>DRIVE</td>
<td>×</td>
<td>√</td>
<td>×</td>
<td>√</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>CHaMeLeoN</td>
<td>×</td>
<td>*</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>CD-MAN</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>ALADIN</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

√: Issue addressed, ×: Issue not addressed, *: Issue identified, but not addressed

The details of the proposed framework (ALADIN) are provided in subsequent chapters. The next chapter introduces the proposed framework. In particular the discovery (3.2.3), data representation (3.2.4), query processing (3.2.5), replication (3.2.8), and security (3.2.9) components are introduced. The discovery component is discussed in detail in Chapter 4, while Chapter 5 talks about semantic data representation. The cache management problem is presented in Chapter 6 where a detailed account of cache replacement and consistency management problems is provided.
Chapter 3

ALADIN: A Data Management Framework

3.1 Introduction

This chapter presents ALADIN, an Association Rules Based Network Layer Data and Information Management framework. The theoretical foundation for the framework is provided along with a brief introduction of the framework and essential components.

The framework is developed considering the unique characteristics of MANET and data management challenges discussed in previous chapter. The proposed framework is based on two principles. The first principle (as verified in Appendix A) states that the data items available on the network have inherent relationships. These relationships are also called correlations or associations. The various requests generated by a producer in a session comply with these relationships. By applying different data mining algorithms, these relationships can be discovered. The discovered relations can then be used for optimization of various data management operations.

The resources in MANET are very constrained. The second principle states that the strict layering concept of OSI suite can be relaxed for MANET environments. This cross-layering concept gives rise to enhanced awareness of the lower level network operations, increased resilience, robustness and performance of the MANET protocols (Vashavsky, Reid, and Lara 2005).

3.2 Proposed Framework

Figure 3.1 illustrates the architecture of ALADIN. It comprises different components for data representation, data discovery and cache management etc. ALADIN just outlines the solution for replication and security; hence they are represented by dotted
lines. A number of supporting components for data mining and logging are also provided.

![Key Components of ALADIN](image)

### 3.2.1 Logging Component

The logging component records the request history of the data consumers by means of sessions. A session is defined as: “A chain of discovery requests \( \{h_0, h_1, h_2, h_3, \ldots \} \) floated by a consumer such that time interval between any two consecutive requests is less than a threshold \( \varepsilon \)”. If \( h_j.t \) represents the time when the request \( h_j \) has been made, a session \( H_i \) can be expressed as:

\[
H_i = \{ h_j : ( j = 0 \text{ or } ( h_j.t - h_{j-1}.t < \varepsilon_i ) ) \}
\]  

(3.1)

The log database keeps a partial list of session records in the form of circular linked list (as illustrated in Figure 3.2). A session record is a collection of attributes where each attribute represents a single discovery request.

![Circular Linked List for maintaining session records](image)
Algorithm 3.1 shows the pseudo code for the logging component. Upon receiving a data item request, the current time is compared against the last request from the same source. If this is greater than a threshold, a new session is assumed. If the log database is full, then the oldest record is deleted from the log database. The requested data item is then saved in the database. The computational complexity of the logging component is a constant value i.e. $O(1)$. The space complexity is the average size of session records. If $s$ is the number of sessions and $\bar{H}$ is the average length of session, then space complexity logging component can be written as $O(s \times \bar{H})$.

```java
public class LoggingComponent {
    public void start () {
        do {
            Message m = ReceiveMessage();
            if(m.Type == SREQ) {
                Log log = openLogDatabase ();
                if(lastRequestTime – currentTime > threshold) {
                    startNewSession();
                    if(log is full) {
                        deleteOldestRecord(log);
                    }
                }
            write(m.RequestedDataItem);
        }
    }while(!stop);
}
```

3.2.2 Association Rules Mining (ARM) Component
The ARM component applies the association rules mining algorithm on the past discovery requests. This history of past requests is maintained in the log database. By applying the FP-Growth mining algorithm, ARM component estimates the relationships among different data items (Han et al. 2004). If $H$ denotes the set of sessions on which the algorithm is employed, then ARM component yields a set of frequent itemsets $F={F_1,F_2,F_3,\ldots}$, where $F_i=f_{i1},f_{i2},f_{i3},\ldots$ and all the items $f_{ij}\in F_i$ have been requested in tandem in sessions of $H$ for a significant number of times. The output of the ARM component is stored in an array list called mining results. As the FP-Growth algorithm is based on storing the session records in the form of an FP-tree, the time and space complexity of ARM component depends on the total size of FP-tree i.e. session records (Sujatha and Deekshatulu 2009). Initially, the size of the FP-Tree increases but after some time, the tree size remains constant.
3.2.3 Data Discovery Component (DDC)
The discovery component is responsible for finding out the provider for a particular data item available on the network. For this purpose, a proactive advertising module is proposed. This module periodically disseminates the data items and their routes to neighboring nodes. During advertisement, the correlation among data items is exploited to broadcast correlated items in tandem. This is expected to enhance the hit ratio of local consumers.

Neighbors that receive the advertisements cache the information locally, and utilize them for answering queries locally in future. If a query cannot be answered locally, reactive discovery component is used. The reactive module employs an integrated approach based on the underlying routing protocol (AODV/DSR). Any discovery requests are forwarded using the routing protocol until it reaches a node having details about the requested data item. The responses are also sent back using the routing protocol response mechanism. The detailed operations of the proactive and reactive component are provided in next chapter. The advantage of the cross-layer design is that the routes towards the provider are also available with the completion of discovery phase.

To optimize the discovery component, the response of a request also carries anticipated future requests’ details. For this purpose, the correlation patterns computed by the ARM component are utilized. The correlation patterns determine the data items used together in past. The data items correlated with current request is considered as a natural candidate to be requested in future as well. So, the data items correlated with current request is also piggybacked in anticipation of their possible usage in near future. Further details of this strategy are provided in next chapter.

3.2.4 Data Representation Component (DRC)
One of the key issues in MANET is the independence of the nodes to use their own mechanism to represent data. This leads to interoperability issues among the nodes. As the nodes in MANET organize themselves for their goals, they want to interact with each other and share their data and resources. Hence, there must be some mechanism for them to interoperate. However, the resource limitations of MANET make it impossible to describe the whole domain knowledge in the form of a single document.

Hence, ALADIN presents a multilevel ontological approach to data representation that keeps into consideration the scalability demands of MANET. It proposes to maintain two levels of ontology documents. A core ontology document keeps the essential concepts for all types of MANET applications. This core concept is maintained at every node of the network. The nodes can additionally maintain extensions of the core ontology (ext ontology) that describes concepts specialized for any particular application. Any node that needs a particular ontology for understanding the semantics of data can call the discovery component to load that ontology from any remote node of the network. This ensures smooth and scalable growth of storage overhead for metadata management. Further details of the data representation component have been provided in Chapter 5.
3.2.5 Query Processing Component (QPC)
The Query Processing Component and (QPC) is responsible for composition (and
decomposition) of a context-sensitive query based on the user requests. A query
comprises the requested data item (with subsequent details), QoS information and the
list of ontology documents required to comprehend the requested data item. In case an
ontology document mentioned in a particular query is not available locally, it is loaded
from other hosts of the network. Besides processing, QPC can also perform a variety of
optimization tasks to reduce the query execution time and processing overhead etc.
(Perich et al. 2001).

3.2.6 Context Manager
In MANET, it is vital to capture the environment settings via a set of attributes because
of intermittently varying surroundings and context-sensitive queries posed by the
consumers. For example, a user can quest for the nearest location where the current
prayers can be offered. Here, the node needs to maintain the current location as well as
the current time, and furnishes this information to query processing component for the
composition of a context-sensitive request analogous to user’s preferences. These set of
traits that specify the situation of a node residing in MANET is termed as context
(Broens 2004). Examples of contextual information include location, time and QoS
information etc. The Context Manager has the task of capturing the contextual
information and furnishing this information to query components. The contextual
information can be acquired via physical sensors or logical sensors. Besides the simple
context management approaches, advanced techniques for context management can be
incorporated (Abbasi and Shaikh 2009; Aqeel-ur-Rehman et al. 2010).

3.2.7 Cache Management Component (CM)
Caching plays a substantial role in management of data in MANET. It allows the nodes
to operate smoothly with the desired data even when the producers are not available on
the network. After the data has been discovered and consumed, the node can cache the
data locally for future usage. There are two sub-components of cache management i.e.
cache replacement and cache consistency management.

ALADIN proposes a correlation based cache replacement heuristics. The information
computed by the mining component provides underlying relationship among data items.
It is expected that those data items that are correlated should remain together in the
cache. So, if any new data item is to be cached, then those data items that are correlated
with the new data item must not be taken out of the cache. After subtracting the
correlated data item from candidate set, any of the current algorithms can be applied to
the left over candidate data items to determine a victim slot.

To maintain a consistent state of cached data items, a network layer consistency
management scheme has been presented. The routing protocols like AODV and DSR
maintain the consistency of their routes when any topological changes in the network
occur. The proposed work has extended AODV and DSR such that along with route
maintenance, consistency of data items is also maintained. Upon the failure of any link
or node, local route maintenance is tried out. Consequently, the upstream nodes are
notified about these changes. Similarly, any changes in the state of data are also
propagated to upstream nodes. The details of the caching component have been
provided in Chapter 6.
3.2.8 Replication Component

The replication process is an alternative strategy (besides caching) to ensure smooth operation in network failure. Similar to caching, the replication process can make use of associations among the data items to determine the replicas that can be hived up locally for future usage. Let \( r_1, r_2, r_3, \ldots \) be the data blocks available for replication and the set \( D_r \) signifies the data items to be replicated. The set \( D_r \) then contains data blocks \( r_i \) such that \( r_i \) either holds currently used data item \( d \) or contains any data item \( d' \) that is correlated to \( d \). The set of data blocks to be replicated \( D_r \), can thus be determined as follows:

\[
D_r = \{ r_i: (d \in r_i) \lor (\exists d': (d' \in r_i \land \rho(d, d') > \varepsilon_r)) \}
\]  

(3.2)

where, \( \rho \) represents the correlation between any two data items and \( \varepsilon_r \) is a constant threshold value.

3.2.9 Security Component

As the MANET is established among stranger nodes that may perhaps be hostile to each other, the nodes might experience a range of security attacks. There are a number of security solutions proposed in literature that can be used by ALADIN for maintaining the safety and privacy of hosts in MANET. An Intrusion Detection System (IDS) based solution to identify an intruding node in the network on the basis of correlation patterns is sketched below. Further details have been presented in (Islam, Shaikh, and Aqeel-ur-Rehman 2011b; Islam, Shaikh, and Aqeel-ur-Rehman 2011c; Islam and Shaikh 2013).

The proposed solution maintains the deviation count \( (dc) \) of a particular session \( H \) and based on this count, it judges the integrity of the particular session. For every request \( d \), the deviation count \( (dc) \) of a particular session is updated as follows:

\[
dc = \begin{cases} 
    dc & \text{if } \exists h \in H: \rho(d, h) > \varepsilon_{s_1} \\
    dc + 1 & \text{otherwise}
\end{cases}
\]

(3.3)

ALADIN expects that all the data items requested in particular session should be correlated (i.e. correlation value \( \rho \) is greater than a threshold \( \varepsilon_{s_1} \)) and if any non-correlated data request is made, ALADIN assumes this as a deviation from normal behavior and updates the deviation count. If the deviation count of a session exceeds a particular threshold \( \varepsilon_{s_2} \), the session can be marked as belonging to a malicious node. The integrity \( I \) of a session can be estimated as follows:

\[
I(H) = \begin{cases} 
    \text{normal} & \text{if } dc < \varepsilon_{s_2} \\
    \text{malicious} & \text{otherwise}
\end{cases}
\]

(3.4)

3.3 Summary

In this chapter, a data management framework for MANET is presented. The proposed framework is based on multi-level data representation, data correlation and cross-layer design. Various components of the framework have been outlined based on these principles. Next chapters will describe and evaluate the discovery, representation and caching components. The evaluation of other components is left for future work.
Chapter 4

A Network Layer Data Discovery Approach for MANET

4.1 Introduction
Data Discovery is the focal part of any data management framework. For this purpose, ALADIN proposes a hybrid solution comprising proactive advertisements and reactive fetching of data from the network. This chapter discusses the discovery component of ALADIN. The architecture of proposed component is first presented, followed by the algorithmic details of the different sub-components. An analysis of the proposed solution is done using JIST/SWANS simulator. This is followed by discussions on experimental results and conclusion.

4.2 Data Discovery Component
Figure 4.1 shows the block diagram illustrating the working of proposed discovery component. There are two major components. A proactive component is accountable for keeping track of the data available on the network, in advance. Any data item request is first searched in local catalog. If request is not satisfied locally, a reactive discovery component is invoked. The reactive component propagates its requests on the network, which are answered by the producer nodes.
4.2.1 Proactive Component

Figure 4.2 shows the proactive discovery component of ALADIN. It comprises two sub-components: advertiser and listener modules. The advertiser is responsible for dissemination of local node’s information to neighboring nodes while the listener is responsible for storing the received advertisements into local catalog.

**Advertiser Component**

The advertiser periodically advertises a subset of its information to adjacent nodes as an advertisement message (SADV). The SADV message contains ID of the data items being advertised, details of the provider (address, name and QoS etc.) and the routing details to approach the provider.
Algorithm 4.1 shows the pseudo code of the advertiser component. The advertiser upon startup, creates an advertisement message and sends to adjacent nodes. It then sleeps for some time period and then repeats the advertisement process.

There are two modes of operation of advertiser. An advertiser can advertise the data items in correlated mode. In this mode, correlated data items are disseminated together. If $\rho(a, a')$ signifies the coefficient of correlation between data item $a$ and $a'$, then the $SADV$ by a node $i$ contains the following data items:

$$SADV = \{a|a \in D_i \text{ and } (\forall a' \in SADV: \rho(a, a') > \varepsilon_a)\}$$  \hspace{1cm} (4.1)

So, the advertiser module disseminates data items such that correlation between any two advertised data items must be above a threshold $\varepsilon_a$. Alternatively, an advertiser can disseminate the information in simple round robin fashion (Islam and Shaikh 2008). In this case, the advertised message $SADV(s)$ of length $s$ by a node $i$ can be expressed as follows:

$$SADV(k) = \begin{cases} LRU(D_i) & k = 0 \\ SADV(k - 1) \cup LRU(D_i - SADV(k - 1)) & k < s \end{cases}$$  \hspace{1cm} (4.2)

Algorithm 4.2 and 4.3 shows the pseudo code describing the working of advertiser component in two modes. In the correlated mode (Algorithm 4.2), the algorithm picks a least recently broadcasted data $d$ and then selects all those data items that are correlated to this data. The variable $\text{minedResult}$ contains the list of correlated data items, as computed by the mining component.
Algorithm 4.2: Advertisement in Correlated Mode

```java
public void advertise() {
    DataItem d = getLRUItem();
    Advertisement advertisement;
    advertisement.add(d);
    for(i=0;i<minedResult.size();i++) {
        if(minedResult.get(i).contains(d)) {
            advertisement.add(minedResult.get(i).getCorrelatedItems(d));
        }
    }
    broadcast(getSADVPackets(advertisement));
}
```

In the case of non-correlated mode (Algorithm 4.3), the algorithm uses the concept of sliding window. There is a pointer \( p \) that points to the start of the current window. The algorithm picks a set of data items whose size is equal to the width of window i.e. \( \text{windowSize} \). The algorithm then sends this information to neighboring nodes and advances its pointer \( p \) to \( p + \text{windowSize} \).

Algorithm 4.3: Advertisement in Non-Correlated Mode

```java
public void advertise() {
    Advertisement advertisement;
    for(i = p; i < p+windowSize && i<datatable.size(); i++) {
        advertisement.add(datatable.get(i));
    }
    p = p + windowSize;
    if(p> datatable.size()) {
        p = 0;
    }
    broadcast(getSADVPackets(advertisement));
}
```

Listener Component

The listener module periodically listens for incoming SADV messages and can selectively accumulate this advertisement in local data tables. Algorithm 4.4 shows the pseudo code for the working of the listener component. Upon start, the listener component waits for an incoming message. If the message is advertisement, listener extracts the advertised data items and corresponding details (routing and QoS specifications) from the message, and places this information into the local catalog. Let \( \oplus \) denotes the operation to update the data table, then the functioning of listener can be expressed as follows:
Algorithm 4.4: ALADIN’s Listener Component

```java
public class Listener {
    void start() {
        Message m = ReceiveMessage();
        if (m.Type == SADV) {
            storeAdvertisement (m.Advertisement);
        }
    }

    public void storeAdvertisement(Advertisement a) {
        for (i = 0; i < a.size(); i++) {
            datatable.place (a.getDataItem(),
                             a.getProvider(),
                             a.getRoute(), a.getQoS());
        }
    }
}
```

4.2.2 Reactive Component

The reactive component is activated when a node needs a particular data item that is non-existent in its own data table. The reactive component composes a message SREQ which is circulated on the network via any on-demand routing algorithm. Following sections discuss how the SREQ is disseminated on the network using two popular MANET routing algorithms.

**AODV Based On-Demand Discovery Scheme**

The scheme proposed here is an adaptation of the work proposed by Torres and García-Macías (2004). Upon the request of consumer $i$, a node prepares a request SREQ and sends it to adjacent nodes on the network. The information included in SREQ are AODV routing headers, message ID, message type, session ID, length, specifications of requested data item, QoS attributes and the name of ontology documents required to understand the request etc. Any node $j$ that receives this SREQ examines its data table to inquire if it knows anything about the requested data item i.e. $\forall x \in D_j (x.des = d.desc$ and $x.QoS \approx d.QoS)$. If the request can be accommodated, a response SREP is generated. Otherwise, $j$ creates a temporary reverse entry locally that tells about the hop who forwarded this request. The node then transmits the SREQ to its adjacent nodes. To circumvent the network from being overloaded with SREQ, two checks are ensured before propagating the request. The TTL value of the SREQ is ensured to be lesser than the maximum value of TTL. In addition, if a message has already been forwarded, it is not forwarded again. The process of propagation carries on until it reaches a node that hosts data item $d$ or to a node that has the information about the provider of $d$. In both of
these cases, a response SREP is transmitted back to the consumer following the same route that was used for request dissemination. The temporary reverse entry maintained by intermediate nodes determines the next hop to which the SREP is forwarded. When the SREP is being delivered back to source, a forward routing entry is created similar to AODV (Boukerche et al. 2011). Thus, a routing path is established along the whole process of discovery that can be used by the consumer for accessing the data item. The information enclosed in SREP include AODV routing headers, message Id, message type, session Id, length, provider details and the list of ontology documents. Figure 4.3 shows how an SREQ is propagated on the network and how its response is sent back to the requesting node. Node 1 needs a data item that is available at node 7. The node 1 requests 2 and 3 for the data. This request is further send to 4, 5 and then to 7. Since node 7 hosts the desired information, it generates a SREP which is sent back to 5, to 2 and then to 1.

Figure 4.3: AODV based Proactive Discovery

Algorithm 4.5 shows the working of the proposed AODV based discovery. The algorithm listens for incoming message. Once a message is received, it checks if it has already processed this request. If this is a new request and the received packet is SREQ, the algorithm looks in the data table if it contains the information about the request or not. In the former case, the algorithm prepares a SREP and replies to the requesting node. In the latter case, the algorithm propagates the request and creates a temporary reverse entry. Any node that receives the message with type SREP, inquires if the response is destined for itself. The algorithm updates it data table, creates a forward routing table entry and access the data, if the message is intended for the node itself. In the latter case, the algorithm propagates the SREP to its upstream node using the temporary reverse entry maintained during the propagation of SREQ.
Algorithm 4.5: ALADIN’s AODV based Reactive Discovery Component

```java
public class Reactive {
    public void start() {
        do {
            Message m = receiveMessage();
            if(processedList.contains(m.MessageId)) {
                continue;
            }

            if(m.MessageType == SREQ) {
                if( match datatable, m.DataItem, m.QoS, m.Ontologies)) {
                    send(createSREP(m));
                }else{
                    routingEntries.insert(m.Originator, m.Source);
                    sendToAdjacentNodes(m);
                }
            }
            if(m.MessageType == SREP) {
                if(m.destination == localHostAddress) {
                    updateDataTable (m);
                    routingEntries.insert(m.Originator, m.Source);
                    accessData();
                }else{
                    updateDataTable (m);
                    routingEntries.insert(m.Originator, m.Source);
                    sendToUpStreamNode (m);
                }
                processedList.add(m);
            }
        } while(!stop);
    }
}
```

**DSR Based On-Demand Discovery Scheme**

This approach is based on DSR routing protocol and is very analogous to the strategy outlined in previous section for AODV case (Boukerche et al. 2011). Algorithm 4.6 shows the working of the DSR based scheme. The node generates a request SREQ and forwards it to the adjacent nodes, which is propagated by the intermediate nodes, similar to AODV.

However, instead of maintaining a reverse route, it appends the addresses of the hops navigated by SREQ message. The SREQ message includes the headers similar to AODV case like specifications of requested data item, QoS attributes and the name of ontology documents etc. In addition a ‘list of IP address’ field is maintained to keep track of the hops traversed by the request. The SREP message generated by the hosting node (or any intermediate node) is sent back via the list of IP address maintained in the
message. Algorithm 4.6 shows the pseudo code describing the DSR based discovery of reactive component.

Algorithm 4.6: ALADIN’s DSR based Reactive Discovery Component

```java
public class Reactive {
    public void start() {
        do {
            Message m = receiveMessage();
            if(processedList.contains(m.MessageId)) {
                continue;
            }
            if(m.MessageType == SREQ) {
                if(match datatable, m.DataItem, m.QoS, m.Ontologies) {
                    send(CreateSREP(m));
                } else {
                    m.updateHopsTraversed();
                    send(m);
                }
            } else if(m.MessageType == MessageType.SREP) {
                if(m.Destination == localHostAddress) {
                    updateDataTable (m);
                    routingEntries.insert(m.Originator, m.Source);
                    accessData();
                } else {
                    updateDataTable (m);
                    routingEntries.insert(m.Originator, m.Source);
                    listOfHopsTraversed.getNextHop();
                    sendToNextHop(m);
                }
                processedList.add(m);
            }
            while(!stop);
        }
    }
}
```

The algorithm listens for the request packets. If an SREQ message is received, the algorithm checks in its local data table if the request can be accommodated. If it is possible to generate a SREP, the algorithm sends the SREP via the list of hops traversed maintained in the packet. Otherwise, the intermediate node propagates this SREQ to adjacent nodes. In addition, the algorithm appends its local address to message. If an SREP is received by the node, the algorithm checks if the intended destination mentioned in the packet is its local address. In that case, the algorithm updates its data
table, creates a reverse routing table entry and accesses the data. Otherwise, algorithm extracts the next hop from the list of IP address and sends the packet to the next hop.

**Using Association Rules Mining to improve discovery process**

In conjunction with the response of requested data item, DDC also generates the responses for the anticipated future request (Islam and Shaikh 2009, Islam et al. 2010). Anticipated requests are determined based on the relation with currently requested data. DDC attaches these computed responses with the current response. The logic behind this piggybacking is the philosophy that requests issued by nodes are highly dependent on intrinsic relationship among the requested data items (Appendix A.1). Hence, data items that have been requested together previously are also expected to be used together in future as well. DDC capitalizes on this association among data items to predict anticipated future requests, which is then appended with current request. The session Id maintained in the SREQ is used to keep track of the requests belonging to a particular session and these requests are logged via the logging component. If \( d \) represents the requested item by a node \( i \), then SREP contains the information about the following data items:

\[
SREP = \{ d \cup \forall_{j \in D_i} \rho(d, j) > \varepsilon_a \}
\]  

In other words, the SREP includes those data items whose correlation with current data item falls above a threshold \( \varepsilon_a \). Algorithm 4.7 shows how a node generates response SREP for a particular data item request \( d \).

---

**Algorithm 4.7: Piggybacking of potential future requests by ALADIN’s Discovery Component**

```
SREPMetaMessage createSREP(SREQMessage m, DataItem d) {
    SREPMetaMessage r = new SREPMetaMessage();
    // copy routing headers
    r.destination = m.source;
    r.source = InetAddress.getLocalHost();
    r.nextHop = m.previousHop;
    // copy other essential fields
    r.sessionId = m.sessionId;
    // now get data items related to request
    for(int i=0; i<minedResult.size(); i++) {
        if(minedResult.get(i).contains(m.DataItem)) {
            r.RelatedDataItems.add(minedResult.get(i).getCorrelatedItems(d));
        }
    }
    return r;
}
```
The node $i$ first copies the essential details from the request packet and then computes the items related to $d$. It then appends this information with the current response and corresponding details. The SREP is then transmitted back to the requesting node using the underlying discovery protocol.

### 4.3 Implementation Details

In order to analyze the performance of the discovery component, a prototype implementation has been developed. A number of network simulation tools are available for MANET simulation (Stanica, Chaput, and Beyolt 2011). In this work, JIST/SWANS++ has been chosen for simulation (Barr, Haas, and Renesse 2004). It comprises a discrete event simulator called Java in Simulation Time (JIST) and a Scalable Wireless Ad hoc Network Simulator (SWANS++). The discovery component has been evaluated using the simulator via a suite of experiments conducted using AODV and DSR based protocols (Boukerche et al. 2011). The simulations were carried out with two types of mobility models (Stanica, Chaput, and Beyolt 2011): i) the random way point mobility model and ii) the street mobility model. The various parameters of simulation are listed in Table 4.1.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Area (m$^2$)</td>
<td>$500 \times 500$</td>
</tr>
<tr>
<td>Node Placement</td>
<td>Random</td>
</tr>
<tr>
<td>Simulation Time (sec)</td>
<td>5000</td>
</tr>
<tr>
<td>Resolution Time (sec)</td>
<td>60</td>
</tr>
<tr>
<td>Transport Protocol</td>
<td>UDP</td>
</tr>
<tr>
<td>Path Loss Model</td>
<td>Free Space</td>
</tr>
<tr>
<td>Mobility Models</td>
<td>Random Way Point and Street Mobility (Suffolk County Map)</td>
</tr>
<tr>
<td>Correlation Coefficient</td>
<td>0.8</td>
</tr>
<tr>
<td>Caching</td>
<td>LRU</td>
</tr>
<tr>
<td>Data Representation</td>
<td>Data Identifiers</td>
</tr>
</tbody>
</table>

### 4.3.1 Datasets Generation

The simulation experiments conducted for the analysis of ALADIN use three datasets to generate sessions of various characteristics at simulation time. These datasets are random dataset, correlated dataset and YouTube dataset.

**Random DataSet**

Based on this dataset, the generated session $H$ comprises purely random requests generated using the Java Math’s library as follows:

\[
H(0) = \text{random}(0, |D|) \\
H(i) = \{ j: j = \text{random}(0, |D|) \land k < i : j \ H(k) \} \tag{4.5}
\]
**Correlated Dataset**

A correlated dataset is required for generating sessions comprising correlated requests. This session is used during correlated advertisement by proactive advertiser as well as during piggybacking of requests. In order to produce request sessions comprising correlated items, a *Correlated Data Generator* (CDG) has been developed. The CDG generates request sessions based on a randomly generated correlation matrix $M$ that can be defined as follows:

$$M(i,j) = \begin{cases} 0 & \rho(i,j) < \varepsilon_a \\ 1 & \text{otherwise} \end{cases}$$

(4.6)

For simulation purpose, the correlation matrix can be populated with random values as follows:

$$M(i,j) = \begin{cases} 0 & \text{random}(0,|D|) < 0.5 \\ 1 & \text{otherwise} \end{cases}$$

(4.7)

To generate a session $H$, a candidate set $C'$ is first computed. The set $C'$ comprises those items that are likely to be floated during session. For calculating $C'$, a random data item $d$ is picked from the set $D$. The set $C'$ comprises $d$ as well as the items related to $d$ i.e.

$$C' = \{i: (i = d) \lor (M(i,d) = 1)\}$$

(4.8)

The session $H$ is then generated based on selecting a set of request from $C'$ as follows:

$$H = \{C'(i): \text{random()} < \varepsilon_H\}$$

(4.9)

where, $\varepsilon_H$ is a threshold that determines how likely a data item is to be requested on the network.

**YouTube Based Dataset**

Instead of using a randomly built correlation matrix (Eq. 4.3), a hard coded correlation matrix can also be used to spawn request sessions having correlated data. In appendix A, an FP-Growth mining algorithm is applied on a campus dataset. Based on the results generated from the application of FP-Growth algorithm on the YouTube video requests, a correlation matrix has been built. This correlation matrix reflects the relationship among the data. Table 4.2 shows the generated correlation matrix. The first row and column of the matrix represents the video ids. A value of ‘1’ at $(i,j)$ indicates correlation between videos with id $i$ and $j$.

### 4.4 Results

A suite of experiments have been conducted based on the three datasets discussed previously. The next section briefly discusses the experimental design. The results of the experiments are subsequently reported.

#### 4.4.1 Experiment Design

In order to perform complete analysis, the efficacy of individual optimization techniques proposed by discovery component has to be analyzed. The objective of the analysis is to observe the impact of these experiments on two parameters i.e. latency and
hit ratio. *Latency* is the difference of time when a discovery request was issued to the time when its reply is received. *Hit Ratio* is defined as the ratio of the number of discovery requests satisfied locally to total number of discovery requests issued.

### Table 4.2: Correlation Matrix based on YouTube video requests of a campus’ users

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

The analysis starts with investigating the advantage gain by introducing a naïve advertisement module over the reactive discovery process. Then, the additional performance gain by the associative advertisement over naïve advertisement is determined. The reactive discovery module is analyzed by comparing its performance with the case when only P2P advertisements are used for discovery of data. This is further followed with studying the effect of piggybacking on discovery performance. Finally, an overall comparison has been drawn with MoGATU.

### 4.4.2 Proactive Discovery Component

The analysis of proactive discovery component involves assessing the impact on latency and hit ratio when i) a naïve (non-correlated) broadcast module is used for data and route advertisement, and ii) associative advertisement is performed for data and route dissemination.
**Analysis of Naïve (Non-Correlated) Advertisements**

As discussed earlier, Torres and García-Macías (2004) proposed a network layer discovery approach based on AODV protocol. ALADIN extends this approach by adding a periodic advertising module on top of discovery component. Figure 4.4 and Figure 4.5 show the impact of adding a naïve advertisement component over Torres and García-Macías (2004)’s discovery approach. The parameters of the simulation are shown below the results. As evident from Figure 4.4, the naïve advertising module gives rise to improvement in latency about 1.2 sec. Overall, an improvement of \(\frac{1.2}{18.1} = 6.6\%\) is observed. The improvement in latency is observed because the advertised data items are stored by listening nodes. These stored advertisements are then used to satisfy the discovery requests locally in future, thus leading to improved discovery time.

![Graph showing latency comparison](attachment://graph.png)

**Simulation Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Naïve Advertisements (Torres and García-Macías 2004)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>50</td>
</tr>
<tr>
<td>Number of data items</td>
<td>32</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>AODV</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Street Mobility</td>
</tr>
<tr>
<td>Avg. latency</td>
<td>16.9</td>
</tr>
<tr>
<td>Min. latency</td>
<td>0.0</td>
</tr>
<tr>
<td>Max. latency</td>
<td>42.4</td>
</tr>
<tr>
<td>Avg. latency</td>
<td>18.1</td>
</tr>
<tr>
<td>Min. latency</td>
<td>0.0</td>
</tr>
<tr>
<td>Max. latency</td>
<td>42.5</td>
</tr>
</tbody>
</table>

Figure 4.4: Analysis of Latency of DDC with Non-Correlated (Naïve) Advertisements

Figure 4.5 compares the hit ratio due to the incorporation of naïve advertisement against Torres and García-Macías (2004)’s approach. From the results, the improvement in hit ratio due to the naïve advertisement is found out to be 2.1%. This is about \(\frac{2.1}{9.0} = 23.3\%\) improvement with respect to Torres and García-Macías (2004)’s approach. As
discussed above, the improvement is observed because the periodic advertisement leads
to the data item gets available in the local storage, and then these stored advertisements
satisfy most of the data requests locally.

Simulation Parameters

<table>
<thead>
<tr>
<th></th>
<th>50</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>50</td>
<td>64</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>AODV</td>
<td>Street Mobility</td>
</tr>
<tr>
<td>Mobility model</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Naïve Advertisements (%)

<table>
<thead>
<tr>
<th></th>
<th>Naïve Advertisement</th>
<th>(Torres and García-Macías 2004)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. hit ratio</td>
<td>11.1</td>
<td>Avg. hit ratio 9.0</td>
</tr>
<tr>
<td>Min. hit ratio</td>
<td>0.7</td>
<td>Min. hit ratio 1.3</td>
</tr>
<tr>
<td>Max. hit ratio</td>
<td>22.4</td>
<td>Max. hit ratio 19.1</td>
</tr>
</tbody>
</table>

Figure 4.5: Analysis of Hit Ratio of DDC with Non-Correlated (Naïve) Advertisements

Analysis of Associative Advertisements
This section analyzes the effect on latency and hit ratio due to the incorporation of
associative advertisement module. Figure 4.6 shows the latency of associative
advertisement with DSR routing protocol. The improvement in latency is about 1.5 sec
(8.1%) with 50 nodes that are running DSR protocol. There are 32 data items available
in the network. YouTube dataset has been used to generate correlated data requests and
random way point mobility model has been used to control the movement of nodes
during simulation.
Having looked at the latency, Figure 4.7 highlights the impact on hit ratio by performing the associative advertisements at repeated intervals. The experiment has been done with 100 nodes running DSR routing protocol. CDG data set is used to spawn correlation request sessions. There are 32 data items available on the network. The graph shows that the associative advertisement approach clearly dominates the naïve advertisement technique. The average rise in hit ratio is about 3.0%, which is 22.4% as compared to the naïve advertisement technique.
4.4.3 Reactive Data Discovery Component

This section analyzes the behavior of reactive discovery component. The analysis comprises investigating i) the impact on latency due to integration of the routing and data discovery operations, and ii) the impact on latency and hit ratio due to the piggybacking of anticipated data items with current response.

**Analysis of integration of data and route discovery operation**

ALADIN’s integrated discovery scheme is based on the approach discussed in (Torres and García-Macías 2004). As Torres and Garcia-Macias (2004) used NOM (Doval and O’Mahony 2002) for analysis of their proposal, this section uses NOM as the basis for evaluation. The comparison with Torres and Garcia-Macias (2004) has already been reported in section 4.4.2. NOM is a P2P service discovery algorithm, based on peer-to-peer advertisement of service’s information. Figure 4.8 and 4.9 perform the latency analysis by comparing the proposed approach with NOM. The first experiment uses AODV and the second experiment uses DSR as underlying routing protocol. In these experiments, neither proactive advertisement operations are performed nor are piggybacking of data items done. The graph shows that the latency due to network layer discovery is far below the NOM. The average reduction in latency is about 1.8 sec i.e. 7.6%.

### Table 4.2: Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>100</td>
</tr>
<tr>
<td>Number of data items</td>
<td>32</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>DSR</td>
</tr>
<tr>
<td>Correlation matrix</td>
<td>CDG</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Way Point</td>
</tr>
<tr>
<td>Associative Advertisement</td>
<td>(%)</td>
</tr>
<tr>
<td>Avg. hit ratio</td>
<td>16.4</td>
</tr>
<tr>
<td>Min. hit ratio</td>
<td>0.3</td>
</tr>
<tr>
<td>Max. hit ratio</td>
<td>41.5</td>
</tr>
<tr>
<td>Naïve Advertisement (%)</td>
<td>13.4</td>
</tr>
<tr>
<td>Avg. hit ratio</td>
<td>0.4</td>
</tr>
<tr>
<td>Min. hit ratio</td>
<td>36.1</td>
</tr>
</tbody>
</table>

**Figure 4.7: Analysis of Hit Ratio with Correlated (Associative) Advertisement**

---

43
Simulation Parameters

- Number of nodes: 50
- Number of data items: 32
- Routing protocol: AODV
- Mobility model: Street
- N/W Layer (sec): NOM (Doval and O'Mahony 2002) (sec)
- Avg. latency: 20.3
- Min. latency: 0.1
- Max. latency: 46.0

Figure 4.8: Analysis of Latency of AODV based Reactive Discovery Component Vs NOM

Simulation Parameters

- Number of nodes: 100
- Number of data items: 32
- Routing protocol: DSR
- Mobility model: Random Way Point
- N/W Layer (sec): NOM (Doval and O'Mahony 2002) (sec)
- Avg. latency: 23.1
- Min. latency: 1.0
- Max. latency: 49.0

Figure 4.9: Analysis of Latency of DSR based Reactive Discovery Component Vs NOM
Analyzing the impact of piggybacking of data items

To analyze the impact of piggybacking on discovery performance, this feature has been plugged in the network layer discovery module. The implementation of the network layer discovery module is same as proposed in (Torres and García-Macías 2004). Figure 4.10 shows that the latency improvement by the incorporation of piggybacking component is about 2.5 sec. This is about 11.2% of the approach proposed by Torres and García-Macías (2004). There are 50 nodes in the network running AODV protocol, 64 data items are available on the network, and YouTube dataset has been used to generate data request sessions.

Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>50</td>
</tr>
<tr>
<td>Number of data items</td>
<td>64</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>AODV</td>
</tr>
<tr>
<td>Correlation matrix</td>
<td>YouTube</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Street</td>
</tr>
<tr>
<td>Piggybacking sec</td>
<td>(Torres and García-Macías 2004)</td>
</tr>
<tr>
<td>Avg. latency</td>
<td>19.9</td>
</tr>
<tr>
<td>Avg. latency</td>
<td>22.4</td>
</tr>
<tr>
<td>Min. latency</td>
<td>1.0</td>
</tr>
<tr>
<td>Min. latency</td>
<td>1.1</td>
</tr>
<tr>
<td>Max. latency</td>
<td>49.0</td>
</tr>
<tr>
<td>Max. latency</td>
<td>46.0</td>
</tr>
</tbody>
</table>

Figure 4.10: Analysis of Latency with the Piggybacking approach

Figure 4.11 analyses the hit ratio when piggybacking feature has been used with 50 nodes running AODV routing protocol and 64 data items are available on the network. The graph compares the hit ratio with proposed piggybacking approach, Torres and García-Macías (2004)’s approach, and Bhole and Ranade (2012)’s approach. Torres and García-Macías (2004) don’t use any piggybacking strategy, while Bhole and Ranade (2012) performed piggybacking based on Apriori algorithm. It is obvious from the graph that the hit ratio remains almost same irrespective of the data mining algorithm used. So, one can use the FP-Growth mining algorithm proposed in this dissertation or an Apriori based approach to data mining as proposed by Bhole and Ranade (2012).
Simulation Parameters
- Number of nodes: 50
- Number of data items: 64
- Routing protocol: AODV
- Correlation matrix: YouTube
- Mobility model: Street

Average Hit Ratio
- Piggybacking using FP-Growth mining (Torres and García-Macías 2004): 12.5%
- Piggybacking using Apriori mining (Bhole and Ranade 2012): 8.3%
- Piggybacking using FP-Growth mining (Bhole and Ranade 2012): 12.6%

Figure 4.11: Analysis of Hit Ratio with the Piggybacking approach

The graph also highlights the impact of piggybacking on hit ratio. In particular, the piggybacking approach improves the hit ratio of consumers about 4.2%. When compared to (Torres and García-Macías 2004), this improvement is about 50.6%.

Finally, this section compares the proposed data discovery component with MoGATU (Perich, Joshi, and Chirkova 2006). Two experiments have been conducted where MoGATU’s cross-layer discovery component is compared with ALADIN’s data discovery component. Figure 4.12 shows the outcome of the first experiment, where 100 nodes running AODV protocol are used during simulation. The figure illustrates latency incurred by the MoGATU and ALADIN for various discovery requests floated on the network. It can be seen that the proposed approach outperforms MoGATU in terms of average latency by 1.8 sec i.e. 10.3%.
Simulation Parameters
Number of nodes 100 Number of data items 64
Routing protocol AODV Correlation matrix CDG
Mobility model Street

**ALADIN** sec **MoGATU** sec
Avg. latency 15.7 Avg. latency 17.5
Min latency 0.0 Min. latency 0.0
Max. latency 41.1 Max latency 43.3

![Figure 4.12: Analysis of Latency of Discovery Component of ALADIN Vs MoGATU](image)

Figure 4.12 compares the hit ratio of proposed discovery component against the MoGATU, after the proposed techniques of associative advertisements and piggybacking have been implemented. It can be seen that ALADIN’s discovery component yields better hit ratio. In particular an improvement of about 2.6% in hit ratio is observed.

### 4.5 Discussions on Results

Table 4.3 summarizes the results obtained from the various experiments. The proposed discovery approach suggests various techniques to improve the performance of discovery component. First of all, the proposed scheme suggests periodic advertisements that can be performed in non-correlated mode (naïve manner) or correlated mode (associative advertisements). The naïve advertisement gives rise to improvement in latency by 1.2 sec in latency and in hit ratio by 2.1%.
Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>ALADIN</th>
<th>MoGATU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>100</td>
<td>64</td>
</tr>
<tr>
<td>Number of data items</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routing protocol</td>
<td>DSR</td>
<td>CDG</td>
</tr>
<tr>
<td>Correlation matrix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mobility model</td>
<td>Street</td>
<td>Street</td>
</tr>
<tr>
<td>Avg. hit ratio</td>
<td>15.5</td>
<td>12.9</td>
</tr>
<tr>
<td>Min. hit ratio</td>
<td>1.2</td>
<td>1.0</td>
</tr>
<tr>
<td>Max. hit ratio</td>
<td>39.5</td>
<td>37.5</td>
</tr>
</tbody>
</table>

Figure 4.13: Analysis of Hit Ratio of Discovery Component of ALADIN Vs MoGATU

Table 4.3: Summary of results of experiments analyzing Data Discovery Component

<table>
<thead>
<tr>
<th>Technique</th>
<th>Latency Improvement (sec)</th>
<th>Hit Ratio Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve Advertisement</td>
<td>1.2</td>
<td>2.1</td>
</tr>
<tr>
<td>Associative Advertisement</td>
<td>1.5</td>
<td>3.0</td>
</tr>
<tr>
<td>Network Discovery</td>
<td>1.8</td>
<td></td>
</tr>
<tr>
<td>Piggybacking</td>
<td>2.5</td>
<td>4.2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>7.0</strong></td>
<td><strong>9.3</strong></td>
</tr>
</tbody>
</table>

Use of associative advertisements give rise to further improvement in latency and hit ratio, as shown in experiments discussed previously. The minor instances where argument is not valid can be due to reasons like a data item request deviates from the normal correlation patterns or the random broadcasting patterns sometimes match incidentally with the data request patterns etc. Improvements of about 1.5 sec in latency and 3.0% in hit ratio have been observed by using associative advertisements as compared to naïve advertisement. The proposed scheme also suggests network layer discovery; that is compared against NOM, an application layer discovery protocol. An improvement of 1.8 sec has been observed. It is also proposed to piggyback anticipated
future request with current response. This is expected to ameliorate the discovery process further. The argument can be invalid in rare cases where the prediction about future discovery requests is not correct and leads to increased network traffic (as the packet is carrying additional data about future data items). However, average improvements about 2.5 sec in latency and 4.2% in hit ratio have been observed. The overall improvement (deduced from the above discussion) in latency using the proposed discovery scheme is about 7.0 sec with an improvement in hit ratio about 9.3%.

4.6 Summary
This chapter has discussed a network layer scheme for data discovery in MANET. Through various experiments, the performance of the discovery component has been evaluated. The time and space complexity of the data discovery component are dependent on the underlying routing protocol, size of the various tables maintained and the total number of nodes on the network. An analysis of complexity of the routing protocols has been reported in (Broustis, Jakllari, and Repantis 2003). The next chapter discusses the data representation component of ALADIN. The various optimization techniques proposed in this chapter (i.e. correlated advertisements and piggybacking etc.) can also be used with the other discovery schemes proposed in literature.
Chapter 5

An Ontology-Based Approach to Data Representation

5.1 Introduction
Heterogeneity is one of the major concerns in MANET. The lack of boundaries in MANET allows various types of nodes to become part of network. This can give rise to interoperability issues. ALADIN provides an ontological approach to address this issue. A multi-level ontology based data representation scheme is presented that describes the semantics of the data.

Rest of the sections of this chapter provides details of the proposed data representation component. The chapter starts with a brief introduction to data representation component. The sub-component details are then discussed, which is followed by a general purpose software ontology for ad hoc and vehicular applications. The implementation details for the evaluation of the proposed approach are then discussed. The results obtained from various experiments are then reported and the chapter ends with a note on the pros and cons of the proposed solution.

5.2 Proposed Data Representation Component (DRC)
Figure 5.1 illustrates a block diagram illustrating the data representation component of ALADIN along with the interconnection with other components. The data representation component prime objective is to maintain a repository of the data items available over the network. The details of these data items (present locally or at some other node on the network) are maintained in data table. As discussed earlier, during data discovery, the routes to the data provider are also discovered. These routing details are maintained in the routing table. The schema information required for comprehending the maintained data items is stored in the ontology component.
5.2.1 Routing Table

The routing table maintains the information required to access the data provider. Table 5.1 shows the fields maintained by routing table. The sequence number determines the freshness of the routing entries. If a routing update is to be performed, it is ensured that sequence number of the update is greater than the current sequence number of the routing entry.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence Number</td>
<td>Determines the freshness of the routing information</td>
</tr>
<tr>
<td>Destination</td>
<td>The destination to be reached using the routing information</td>
</tr>
<tr>
<td>Hop Count</td>
<td>The number of hop counts to reach the destination</td>
</tr>
<tr>
<td>Next Node</td>
<td>The next node to whom packet will be forwarded</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the route (temporary or a permanent route)</td>
</tr>
<tr>
<td>Precursors</td>
<td>The upstream nodes of a particular route</td>
</tr>
</tbody>
</table>

Other field maintained by the table includes: the destination field for which the routing information is stored, the number of hop counts to reach the destination, next hop to which the routing packet is forwarded, status of the route and the precursors’ information. The precursors are the set of nodes likely to use the current node as a next hop during routing.

5.2.2 Data Table

The data table keeps record of the various data items available over the network. Table 5.2 shows the fields maintained in the data table. The data identifier field uniquely identifies a particular data item. The provider details include identity of the node that
owns the data item. The ontology list contains the name of ontology documents required to understand the meaning of data. The QoS attributes include the quality of service information about the provider maintaining these data items.

### Table 5.2: Data Table maintained by nodes

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Identifier</td>
<td>A unique identifier of the data item</td>
</tr>
<tr>
<td>Data Provider</td>
<td>The address of the node hosting the data item</td>
</tr>
<tr>
<td>Ontology List</td>
<td>The list of ontology documents required to understand data</td>
</tr>
<tr>
<td>QoS</td>
<td>The QoS information associated with the data item</td>
</tr>
</tbody>
</table>

#### 5.2.3 Ontology Component

The ontology component maintains the information necessary to understand the meaning of data. To maintain the meta-information considering the resource limitations of MANET, a multilevel ontology is proposed. The ontology comprises a set of documents residing at two separate levels as illustrated in Figure 5.2.

![Figure 5.2: ALADIN’s Multi-level Representation of Ontology](image)

The first level comprises a static core ontology document that is always available at a particular node, and it describes the primitive concepts of any MANET application. The second level comprises a set of extended ontology documents that grow gradually via discovery of ontology from other nodes. The extended documents are developed based on the core ontology and describes the concepts specialized for a particular application of MANET.
Algorithm 5.1 shows the pseudo code describing how the multilevel ontology representation component works. During semantic matching, the ontologies are used for understanding the meaning of data. When an ontology is not available locally, the discovery component is invoked to find out the ontology from the network. The document is searched over the network in the similar fashion as a data item is discovered. The ontology is then loaded from the provider node and stored in local catalog. Once the ontology is discovered, reasoning and inferencing components are used to analyze the semantics of data. If an ontology is not found on the network, the node can still use the semantic information available in the base ontology to make generalized inferences.

**Algorithm 5.1: ALADIN’s Multilevel Ontology Representation Component**

```java
public class DataRepresentation {

    public boolean match (DataItem queryItem) {

        String ontology = queryItem.requiredOntology;
        if (!extOntology.contains(ontology)) {
            extOntology.add(DiscoveryComponent.find(ontology));
        }

        //construct model and reasoner
        Model m;
        ...

        //initialize reasoner
        Reasoner reasoner;
        ...

        //create corresponding query
        String query;

        //Execute query
        QueryExecution qexec = QueryExecutionFactory.create(query, model);

        //now extract relevant information
        ...
    }
}
```
5.3 Proposed Ontology
Figure 5.3 shows the visual representation of the ontology (developed in Protégé) proposed for semantic data representation. The ontology describes concepts in the domain of mobile and vehicular applications. The proposed ontology comprises a core ontology and a set of extended ontologies. The core ontology contains essential concepts as shown by square boxes, which is further extended (as shown by directed arrows) with three ontology documents related to medical, safety and entertainment applications.

![Ontology Diagram]

**Figure 5.3: Proposed Software Ontology (Islam and Shaikh 2012)**

5.3.1 Core Ontology
The core ontology includes the primary concepts for any ad hoc and vehicular applications. The *Profile* concept provides specification of any generalized entity which is sub-classed by concepts related to different types of profile. The *Computational Profile* includes capability information about any host like storage, processing capabilities and battery etc. The *Driving Profile* describes the driving behavior of a driver. It has attributes to describe driver’s accidental history and traffic rules violation etc. The *Spatial Profile* encapsulates location information about any host of the network. Its attributes comprise GPS coordinates of the host’s location.
To maintain the semantics related to any node, the concept of *Ad hoc Node* is proposed. It contains the description about any host in the network through various attributes. These attributes are velocity, direction, computational profile, spatial profile and the details of data items maintained by the node etc. The Ad hoc Node concept is extended further by *VANET Node* to describe any vehicular node and embraces attributes of vehicle dimensions and driving profile etc. VANET stands for Vehicular Ad hoc Network and is a special type of ad hoc network formed between collection of vehicles. The *Place* concept provides details of any place i.e. its spatial profile. The *Person* concept embraces details about any person like person’s name, age, gender and contact details etc. The *Driver* class extends the *Person* class to provide description about the driver of any VANET Node. The *Service* concept contains information about any data item/service like its name, provider, QoS and routing information etc. The concept *Route* holds the routing path details to reach any particular provider.

### 5.3.2 Extended Ontology

The extended ontology is a collection of ontology documents for specialized types of application. ALADIN currently provides ontology for three types of applications: medical, safety and entertainment application. The medical ontology is useful for e-health care applications. The safety ontology can be exploited for transportation applications. It can provide assistance to drivers through accidental warnings and weather warnings etc. The entertainment ontology provides semantic information in the domain of e-shopping and e-hotel applications. The extended ontology set can be extended further by defining ontology for newer domains based on the core ontology.

#### Medical Ontology

The medical ontology introduces several new concepts by extending the base concepts of core ontology. The *Person* class is extended by the concept *Doctor* that contains attributes for doctor’s qualification, experience and specialty etc. The *Hospital* concept extends the *Place* concept to maintain the information about the available doctors and rooms in the hospital. The *Room* describes detail of a room and is extended by four sub-concepts i.e. *General*, *Operation Theater* and *ICU* and *Pharmacy* etc. A new concept *Health Profile* is proposed to maintain information about health of individuals.

#### Safety Ontology

The concept *Street* maintains information about a particular road like its spatial profile, the list of restaurants and shopping places around it etc. The *Place* concept is extended by *FuelStation* concept which is further extended by two sub-concepts i.e. *PetrolStation* and *CNGStation*. The *Status* concept describes different types of status for safety application. The various types of statuses can be *WeatherStatus*, *TrafficStatus* and *RoadStatus*.

#### Entertainment Ontology

The ontology for entertainment applications proposes following concepts: *Shopping Item*, *Shopping Plaza* and *Restaurant* etc. The *Shopping Item* describes any sellable item in a shop. Its attributes include name, price and quantity of shopping items etc. Similarly, the *Food Item* describes any food prepared and served by a restaurant. It has attributes to describe the type of food, its price and cuisine etc.
5.4 Implementation Details
An evaluation of the data representation component is presented in this section. The proposed component has been implemented in JIST/SWANS network simulator (Barr, Haas, and Renesse 2004). Table 5.3 shows the values of various simulation parameters. For ontology modeling, OWL has been used due to its expressive power and extensive support available in existing editing tools. For ontology editing, Protégé has been selected from the large number of available choices (Noy et al. 2001). The tool was chosen owing to its simpler interface, flexibility and large community base. The JENA Semantic Web Toolkit has been selected for further processing, reasoning and inferencing of ontology at simulation time (McBride 2002).

5.5 Results and Analysis
The analysis of DRC component requires investigating the impact of ontology representation on various data management operations. This section first analyzes the increase in latency due to the discovery process. An important part of the analysis is to investigate the scalability of multilevel data representation. Following the latency analysis, the scalability analysis of the proposed scheme is reported.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Area (m²)</td>
<td>500 × 500</td>
</tr>
<tr>
<td>Node Placement</td>
<td>Random</td>
</tr>
<tr>
<td>Simulation Time (sec)</td>
<td>5000</td>
</tr>
<tr>
<td>Resolution Time (sec)</td>
<td>60</td>
</tr>
<tr>
<td>Transport Protocol</td>
<td>UDP</td>
</tr>
<tr>
<td>Path Loss Model</td>
<td>Free Space</td>
</tr>
<tr>
<td>Mobility Models</td>
<td>Random Way Point and Street Mobility (Suffolk County Map)</td>
</tr>
</tbody>
</table>

Figure 5.4 assesses the additional latency incurred due to the incorporation of ontology for semantic matching. The analysis of the results reveals that the average rise in latency is about 2.8 sec (14.2%). The enhancement in latency is due to the additional processing overhead incurs due to the inferencing and reasoning tasks performed by semantic component.
Simulation Parameters

Number of nodes 100
Routing protocol DSR

Semantic Discovery sec Non-Semantic Discovery sec
Avg. latency 22.5 Avg. latency 19.7
Min latency 0.1 Min. latency 0.0
Max. latency 47.1 Max latency 41.3

Figure 5.4: Analysis of Latency with Semantic and Non-Semantic Discovery Component

Figure 5.5 analyzes the scalability of the proposed component. It compares the distribution of meta-data (ontology tuples) when multilevel ontology is maintained versus the incorporation of a single/unified ontology document. Figure 5.5a shows the ontology tuples maintained by various nodes at different stages of simulation. The figure points out that the schema size grows in a graceful fashion as the simulation progresses. Figure 5.5b further highlights the schema requirements at individual nodes during simulation. The size of the schema rises in linear fashion (as also evident from the trend line) and is far below the unified ontology approach. This affirms the scalability of the proposed solution.
Figure 5.5: Analysis of Scalability of Multi-level Schema Management
A thorough analysis of multilevel proposed approach also requires the examination of latency incurred with the proposed approach against the unified approach. Figure 5.6 shows results of an experiment conducted for this purpose. There are 100 nodes running AODV protocol, and 32 data items are available on the network. To understand the efficacy of the multi-level approach, a comparison with the complementary approaches is drawn. A *unified ontology* approach is based on using a single ontology for maintaining semantic information. Since, MoGATU is one of the most comprehensive data management frameworks; hence a comparison of proposed approach with MoGATU has also been provided (Perich, Joshi, and Chirkova 2006). As can be seen in Figure 5.6, in initial stages of simulation, the latency due to the unified approach is better than multilevel approach.

![Figure 5.6: Analysis of Latency of Multi-level Schema Management](image)

**Simulation Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>100</td>
<td>32</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>AODV</td>
<td>Street</td>
</tr>
<tr>
<td>Avg. Latency (sec)</td>
<td>ALADIN: 22.1</td>
<td>MoGATU: 24.9</td>
</tr>
<tr>
<td></td>
<td>Unified Approach: 27.0</td>
<td></td>
</tr>
</tbody>
</table>

The reason is that the multilevel approach works in cold start fashion. When a particular ontology is not found locally (as only a portion of ontology is maintained locally), it starts the discovery process leading to slight rise in latency. However, the average latency with the multilevel DRC approach is better than the unified approach (about 4.9 sec). Similarly, the proposed approach outperforms the MoGATU due to latency improvement by 2.8 sec.

**5.6 Summary**

This chapter presents a scalable approach towards data representation based on multilevel ontology documents. A general purpose ontology approach for addressing the
interoperability issue of MANET is presented. The nodes maintain partial view of the overall schema and ontology grows in on-demand fashion via invoking discovery components as needed. The proposed approach has been simulated in JIST/SWANS simulator. The use of ontology gives rise to semantic interoperability with the cost of slight increase in latency (about 2.8 sec or 14.2%) of the discovery component. The proposed approach therefore leads to improvement in computational and storage overhead. The time and space complexity of the data representation component depend on the various tables (routing and data tables) maintained as well as the average size of the ontology database. Further details on complexity analysis of semantic processing can be seen in (Motik et al. 2012).
Chapter 6

A Correlation Based Caching Scheme for MANET

6.1 Introduction
One of the taxing issues in MANET is to ensure the continuous accessibility of information during nodes and link failures. The simplest solution is to store all the discovered data items in a local catalog. However, it is not practicable due to the limited storage capabilities of node. This issue can be addressed by dedicating a subset of catalog to cache the remote data items and then maintaining a consistent state of the cached information. To accommodate these requirements, ALADIN proposes a caching scheme comprising cache replacement and consistency management sub-components. Each of these components is discussed in sections below. A quantitative comparison with currently available solutions is also performed and discussed in results section.

6.2 Proposed Caching Component
Figure 6.1 shows the block diagram describing the working of proposed caching component. As mentioned earlier, the caching component comprises two sub-components: the cache replacement component and consistency management component. The former exploits the correlation patterns generated by the mining component to place a newly arrived data item into the cache. The consistency management component works at the network layer. It exploits the routing layer capabilities to determine the topological changes in the network. These topological changes are used to maintain consistency of cached data.
6.3 A Correlation Based Cache Replacement Algorithm

The cache replacement component is built on the notion that if data items are related to each other, they are highly likely to be used in tandem in near future. The algorithm assumes that if any data item \( d \) is to be placed in the cache, a victim slot should be picked from those data items that are not correlated to \( d \) (Islam and Shaikh 2010).

Let \( C = \{c_1, c_2, c_3 \ldots \} \) denotes the set of cache entries and \( R_d \) denotes the set of cache slots holding data items related to \( d \), then \( R_d \) can be written as:

\[
R_d = \forall x \in D_i: \rho(d, x) > \varepsilon_c
\]  

(6.1)

In other words, the set \( R_d \) comprises those data items from \( D_i \) whose correlation value (\( \rho \)) with \( d \) is above a particular threshold (\( \varepsilon_c \)). The residual set \( S_d \) is defined as those elements that are not correlated to \( d \) and can be expressed as:

\[
S_d = C - R_d
\]  

(6.2)

Figure 6.2 shows how the residual set is obtained from the cache. The residual set is considered as the candidate set from which an appropriate victim slot \( v \) can be picked i.e.

\[
v = \begin{cases} 
select(S_d) & \text{if } |S_d| \neq 0 \\
select(C) & \text{if } |S_d| = 0 
\end{cases}
\]  

(6.3)
Algorithm 6.1 shows the pseudo code describing the proposed cache replacement algorithm. The algorithm first attempts to find out an empty slot in the cache. If no empty slot is found, then the algorithm attempts to pick a victim slot from the cache using the correlation based algorithm. It first determines the data items related to \( d \) and assigns these data items to the variable \( CR \). Then the set of items not correlated to \( d \) is computed. This residual set is then assigned to the variable \( S_d \) in the algorithm. Then, the algorithm checks if the residual set is empty or not.

In the former case, all the slots in the cache contain items that are correlated to \( d \). So, the algorithm simply applies least recently used algorithm to all the slots of the cache to determine a victim slot. In the case when residual set is non-empty, least recently used algorithm is applied to the residual set. Having determined an appropriate slot where data item can be placed, the algorithm puts the item \( d \) in to the computed slot \( i \). The touch time of the victim slot is also updated to the current system time.
Algorithm 6.1: ALADIN’s Correlation based Cache Replacement Algorithm

```java
int doCacheReplacement(Cache C, DataItem d) {
    int i;
    if(C.hasEmptySlot()) {
        i = C.placeItem(d);
        C.setTouchTime(i, System.currentTime());
    } else {
        CR = getCorrelatedItems(d);
        S_d = C;
        for (i = 0; i < CR.size(); i++) {
            if (S_d.contains(CR(i))) {
                S_d.remove(CR(i));
            }
        }
        if (S_d.size() != 0) {
            i = getLeastRecentlyUsedSlot(S_d);
        } else {
            i = getLeastRecentlyUsedSlot(C);
        }
        C.setTouchTime(i, System.currentTime());
        C.replaceItem(i, d);
    }
}

int getLeastRecentlyUsedSlot(Set S) {
    int i = -1;
    int minTouchTime = Integer.MAX;
    for (k = 0; k < S.size(); k++) {
        if (S(k).touchTime < minTouchTime) {
            i = k;
            minTouchTime = S(k).touchTime;
        }
    }
}
```

6.4 Consistency Manager

As MANET possesses non-deterministic characteristics, it is essential to employ consistency management techniques to keep an up-to-date view of the state of data provider and the corresponding cached items. ALADIN proposes a consistency manager based on the underlying routing protocol. The proposed approach is a network layer solution and the nodes perform immediate actions upon any changes in the network state. For example, upon failure of a node, a new provider for the cached data is found out. In case of any data changes, a node can notify the upstream nodes.
Algorithm 6.2 illustrates how the state of a data provider is maintained by the consistency manager based on AODV protocol (Islam and Shaikh 2012). Every node in the network monitors its neighbors for link failures. If any failure event has occurred, the consistency manager component is initiated. The component first prepares a list of unreachable destinations due to the link failure. This list is built based upon the routing table entries available at the nodes.

Algorithm 6.2: ALADIN’s Network Layer Consistency Manager Component

```c
void consistencyManagement() {
    while(true) {

        switch(Event e) {

            case LinkFailure:
                Message RERR;
                RERR.unreachableDestination = unreachable neighbor
                RERR.unreachableDestinations += destinations in routing table
                    using the unreachable neighbor as next hop

                Message RERR = prepareError(unreachableDestination);
                if(performLocalRouteMaintenance() == “Success”) {
                    RERR.setNBit();
                } else{
                    invalidateLocalRouteAndCache();
                }
                send(RERR);
                break;

            case RERR received:
                if(there are precursors along the route) {
                    retransmit(RERR);
                }

                if( R.Flag(“N”) == false ) {
                    invalidateLocalRouteAndCache();
                }
                break;
        }
    }
}
```

Instead of propagating the failure event to other nodes, local maintenance is first tried out to repair the link. Hence, an alternate route discovery process is initiated. SREQ messages are propagated to build a path from the node to the unreachable destination.
(similar to AODV). If the local maintenance is successful, the upstream nodes are notified and an RERR message is sent with the $N$ bit set to true. The value of $N$ bit indicates if a local maintenance is successful or not. If the local maintenance is unsuccessful, the upstream nodes along this route are notified about the unavailability of the data provider using an RERR message. Before propagating the message, the node invalidates the local routing table entry and caching entries whose provider is one of the unreachable destinations. The RERR message is propagated to upstream nodes using the same mechanism as AODV (Perkins, Belding-Royer, and Das 2003). The philosophy behind propagating the message to upstream nodes is that the proposed solution maintains a path from consumer to producer of information. Hence, the upstream nodes are the natural consumers of the data available at the provider.

When any node receives an RERR message, it updates its cache and routing table entries and propagates the RERR message. In case of any changes in data item states, the upstream nodes are also notified in the similar fashion. The message then contains the list of data items requiring updating. All the consumer nodes along the upstream path will receive this message and can update the state of the cached data item.

This section has discussed in detail the mechanism for AODV. However, if the underlying routing protocol is DSR, same procedure can be adopted. Upon the detection of any link failure, the DSR notifies the sender via the route information enclosed in the packet. The upstream nodes and sending nodes along the path receive the message and then invalidate their routing and caching entries.

### 6.5 Implementation Details

The proposed caching scheme has been implemented in JIST/SWANS simulator (Barr, Haas, and Renesse 2004). Table 6.1 shows the simulation parameters. Three types of datasets were used for analysis as mentioned in Chapter 4. The first dataset is a Random Dataset that comprises data requests randomly distributed. A Correlated Dataset is generated using random correlation matrix based on correlated data generator. Finally, a YouTube Dataset is also generated based on the same correlation matrix presented in Table 4.2.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Area (m$^2$)</td>
<td>500 $\times$ 500</td>
</tr>
<tr>
<td>Node Placement</td>
<td>Random</td>
</tr>
<tr>
<td>Simulation Time (sec)</td>
<td>5000</td>
</tr>
<tr>
<td>Resolution Time (sec)</td>
<td>60</td>
</tr>
<tr>
<td>Transport Protocol</td>
<td>UDP</td>
</tr>
<tr>
<td>Path Loss Model</td>
<td>Free Space</td>
</tr>
<tr>
<td>Mobility Models</td>
<td>Random Way Point and Street Mobility (Suffolk County Map)</td>
</tr>
<tr>
<td>Correlation Coefficient</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 6.1: Simulation Parameters for experiments analyzing Caching Component
6.6 Results
This section analyzes the hit ratio of the cache replacement component. A higher hit ratio as compared to other schemes shows the effectiveness of the caching component. For the consistency management component, the resultant latency and hit ratio of the data management operations is analyzed against the currently available solutions.

6.6.1 Analysis of Cache Replacement Algorithm
A set of experiments were conducted under various network settings. These experiments analyze the hit ratio of caching component under different mobility models and network configurations. Figure 6.3 analyses the hit ratio with 50 nodes and 16 cache slots. The nodes are running AODV routing protocol, and the data requests are generated using a random correlation matrix. The graph clearly shows that the correlation based caching has better hit ratio than LRU and LRU+P algorithm (Perich et al. 2004).

![Figure 6.3 Analysis of Hit Ratio of Caching Component with AODV Routing protocol](image)

<table>
<thead>
<tr>
<th>Simulation Parameters</th>
<th>Correlation Based Caching</th>
<th>LRU</th>
<th>(Perich et al. 2004)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>50</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Cache size</td>
<td>AODV Correlation matrix</td>
<td>YouTube Correlation matrix</td>
<td>Mobility model Street</td>
</tr>
<tr>
<td>Avg. Hit Ratio</td>
<td>%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Correlation-based caching</td>
<td>16.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LRU</td>
<td>12.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LRU+P (Perich et al. 2004)</td>
<td>15.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.4 also illustrates the rise in hit ratio with DSR routing protocols and larger number of nodes. The number of nodes has been increased to 100 and the cache size is
changed to 64 data items. The hit ratio of the proposed correlation-based caching is still better than the other two schemes. It can be concluded that the average hit ratio of the correlation based caching falls clearly above the other algorithms. In some data points of the graph, the performance of proposed caching algorithm is below the other two techniques. This can be due to sudden network changes like link failures and congestion etc. The other reason can be that data request patterns of a consumer deviates from the expected correlated patterns. Overall, an improvement of 3.2% of hit ratio is seen over traditional LRU caching algorithm.

![Graph showing hit ratio variations](image)

**Simulation Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>100</td>
</tr>
<tr>
<td>Cache size</td>
<td>64</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>DSR</td>
</tr>
<tr>
<td>Correlation matrix</td>
<td>CDG</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Avg. Hit Ratio</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation-based caching</td>
<td>15.6</td>
</tr>
<tr>
<td>LRU</td>
<td>12.4</td>
</tr>
<tr>
<td>LRU+P (Perich et al. 2004)</td>
<td>14.5</td>
</tr>
</tbody>
</table>

**Figure 6.4 Analysis of Hit Ratio of Caching Component with DSR routing protocol**

---

**6.6.2 Analysis of Consistency Management Component**

ALADIN presents a robust consistency management solution based on network layer operations. Two parameters i.e. latency and hit ratio have been selected for assessment of the consistency manager. Figure 6.5 compares the latency of various requests after a network layer consistency manager is used against the case where a lazy consistency manager is used. The lazy consistency manager maintains data state while data is being consumed. The analysis shows that the latency has improved by 3.5 sec due to the incorporation of proposed consistency manager. The reason behind the improvement is that the network layer consistency manager leads to maintenance of data item state as
soon as any change occurred in the network. In case of a lazy consistency manager, the state of data item is determined while it is being consumed. This gives rise to some sharp peaks in the graph and leads to increased average latency.

![Graph showing latency over time](image)

**Simulation Parameters**

<table>
<thead>
<tr>
<th></th>
<th>With Consistency Manager</th>
<th>Without Consistency Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>50</td>
<td>16</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>AODV</td>
<td>Street</td>
</tr>
<tr>
<td>Avg. latency</td>
<td>18.8</td>
<td>22.3</td>
</tr>
<tr>
<td>Min latency</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Max latency</td>
<td>44.5</td>
<td>44.5</td>
</tr>
</tbody>
</table>

**Figure 6.5 Analysis of Latency with proposed Consistency Manager**

Figure 6.6 analyses the hit ratio of ALADIN’s consistency management component against two other consistency management solution proposed in literature by Perich et al. (2003), and Fatima and Khader (2012). The overall hit ratios of the three approaches remain almost same. ALADIN has a slightly better performance than other two approaches.

### 6.6 Summary

This chapter presents a cache replacement and network layer consistency management scheme for MANET. The replacement algorithm prioritizes the cache contents based on their correlation with currently arrived data item. The generated candidate set can be used to pick a victim slot based on any caching scheme like LRU, MFU or MRU etc.
Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>50</td>
</tr>
<tr>
<td>Number of data Items</td>
<td>32</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>AODV</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Street</td>
</tr>
</tbody>
</table>

**Figure 6.6 Analysis of Hit Ratio with proposed Consistency Manager**

The results signify the efficacy of correlation based scheme when LRU is applied on the residual set. However, a combination of correlation based scheme with other scheme proposed in literature (cooperative caching, semantic caching and profile driven caching) can also be used. This chapter also proposes a network layer consistency management scheme that maintains the integrity of data items based on the network layer events information. The analysis of the results proves the robustness of consistency management scheme. The time and space complexity of the cache component are dependent on the size of cache as well as the size of data table.
Chapter 7

Conclusion

7.1 Introduction
This chapter summarizes the thesis by providing an abridgement of the work presented in previous chapters. It highlights the contributions made in different areas and also points out the limitations of the proposed work. The chapter ends with discussions on research avenues for future work.

7.2 Research Overview
Data Management can be considered as the key enabler for ubiquitous environments. The data management mechanisms allow the end-users in MANET to access data of their interests anytime and anywhere. The study of pertinent literature in chapter 2 reflects that data management is a complex job and embraces a number of research issues (Perich et al. 2004; Perich, Joshi, and Chirkova 2006). Each of these research issues themselves present challenges during their application in MANET (Toh, Mähänen, and Uusitalo 2005). Chapter 2 discusses these challenges in detail and reviews currently available solutions. This research focuses on three key issues of any data management system i.e. data representation, discovery and cache management. However, based on the proposed framework, solutions for other issues have also been outlined.

The first issue addressed by this thesis is data discovery. A hybrid correlation based network layer discovery scheme is presented. The advantage of the network layer approach is the availability of data provider details along with the path to reach to the provider. The benefit obtained from incorporation of correlation information is the increased hit ratio of data consumers. A number of experiments have been carried out (as reported in Chapter 4) that reflect the improvement in latency and hit ratio with the proposed scheme.
The second issue that has been addressed in this thesis is related to data representation. Because of the non-interoperability issue in resource constrained MANET, the concept of multilevel schema management is employed. A two-level software ontology has been proposed that enables the hosts in the network to maintain partial schema. The benefit of this approach is the syntactic and semantic interoperability among nodes. In addition, this resolves the issue of limited resources in MANET. The results obtained from various experiments also substantiate the scalability of this approach (Chapter 5).

Finally, a cache management scheme is proposed in Chapter 6. The scheme employs a correlation based approach for cache replacement. The victim slot is determined by applying the LRU algorithm on the items not related to the current item. This leads to improvement in hit ratio of the data consumers. The scheme also proposes a network layer consistency manager that enhances the robustness of various data management operations.

### 7.3 Contributions and Lessons Learned

ALADIN is a data management framework for the MANET environments designed by keeping into consideration the nature of MANET. The framework provides the following contributions:

1. Network layer discovery with improved latency and hit ratio.
2. Semantic interoperability in scalable fashion.
3. The use of data associations for advertisement and piggybacking during discovery.
4. The use of correlation during cache replacement.
5. The use of network layer consciousness for maintaining the consistency of data items.

Unlike the existing solutions proposed by Perich, Joshi, and Chirkova (2006), and Xu and Wolfson (2005); ALADIN introduces several new notions. It adopts the concept of network layer discovery. This approach leads to improved robustness and ensures seamless data management operations under dynamically varying conditions. The framework recommends keeping ontology schema information in on-demand fashion. This resolves the issue of semantic interoperability in a scalable way. ALADIN proposes the use of inherent data associations during discovery, cache management and security management etc. The use of data associations is a novel concept that has not been investigated extensively by researchers specifically during discovery and cache management. This approach leads to improved latency and hit ratio as well as caters with the unreliable links among hosts by letting the nodes to operate in offline mode.

### 7.4 Future Work

Although, the proposed framework addresses the issues of data management in a comprehensive fashion, there are some limitations associated with this framework. The subsequent paragraphs mention these limitations and also highlight future research directions.

The proactive component of DDC currently broadcasts the information indiscriminately without keeping into account the network bandwidth, node’s battery and security
attributes. The component can be optimized by considering these attributes before performing advertisements. It is also necessary to determine the optimum size of idle period before advertisement. Comparative analysis should also be done in determining the performance of periodic advertisements with non-periodic advertisements. Researches should also be done in estimating the optimum size of advertisement packet under various settings. The reactive component also demands further investigation. This thesis discusses extensions for two routing protocols (AODV and DSR). Other reactive routing protocols like Optimized Link State Routing (OLSR), Associativity Based Routing (ABR) and Temporally Ordered Routing Algorithm (TORA) etc. can be studied for network layer discovery.

The FP-Growth algorithm has been exploited for estimating correlation among data items. However, it is not viable to use this algorithm in MANET settings with severely low resources. One way to address this issue is the development of algorithms that can calculate the associations approximately with minimal resource consumption.

The cache replacement algorithm discussed in this manuscript extends LRU algorithm by using correlation among data items. However, the same can be done with other cache replacement algorithms like MFU, MRU and LRU etc. The log database has been exploited by discovery, caching and security management components. However, further use of this information can be investigated i.e. for query optimization and transaction management etc. Finally, this thesis discussed query processing, replication and security management briefly. A thorough analysis has been left for future work.

7.5 Summary
Data Management has become an important issue of research with the developments in mobile computing. In this dissertation, various issues involved in data management have been investigated and a data management framework is thus proposed. This chapter concludes the thesis by highlighting the main contributions of the thesis. Several directions for future research have also been provided.


Telecommunications Conference (GLOBECOM 2009), Honolulu, Hawaii, USA, November 30- December 4, 2009.


Han, Jiawei, Jian Pei, Yiwen Yin, and Runying Mao. 2004. "Mining frequent patterns without candidate generation: a frequent-pattern tree approach." Data Mining and Knowledge Discovery. 8: 53-87.


presented at International Conference on Computers and Emerging Technologies (ICCET 2011), Shah Abdul Latif University, Khairpur, Pakistan, April 22-23, 2011.


Vashavsky, Alex, Bradley Reid, and Eyal de Lara. 2005. The need for cross-layer service discovery in MANETs. Department of Computer Science, University of Toronto.


Appendix A: Assumptions and Validation

A.1 Data Requests in a session are correlated to each other

The proposed work is based on the assumption that data requests made by consumers in a session are correlated to each other. To validate this notion, one can analyze the data requests’ session of users in a real networking environment. If these requests sessions have some correlation patterns, it can be said that data requests floated by users have inherent relationships. To achieve this objective, FP-Growth association rules mining algorithm has been applied on a YouTube requests dataset (Inokuchi, Washio, and Motoda 2000). This dataset is obtained by gathering http://www.youtube.com video data requests made by users of a university campus as reported in (Zink et al. 2008). The dataset is also available at http://sites.google.com/a/nu.edu.pk/noman-islam/aladin/datasets.zip. The results of the dataset are compared against a Random dataset generated via the random number generator of Java Math’s library. The dataset $D$ is first converted into a standard format before the application of the mining algorithm as shows in Figure A.1.

Let’s analyze the frequent itemsets generated when FP-Growth algorithm has been applied on this dataset. Figure A.2 shows the total number of frequent itemsets generated by the two datasets under various experiments. It can be seen that in a purely random dataset, the number of itemsets generated is very small (12 on average) as compared to the itemsets generated with YouTube dataset (145.5 on average). Figure A.2b and A.2c illustrates the distribution of frequent itemsets generated with YouTube datasets. The statistics illustrates various types of itemsets (1,2,3 or more) produced by the mining algorithm. It is obvious from these statistics that the requests by users of a session are not casual, but they are inspired by inter-relationships among the requested items. The results substantiate that there are correlation patterns present among data, and various types of data requests made by consumers in a session are dependent upon these patterns.
Data Items | YouTube Dataset | Random Dataset
--- | --- | ---
50 | 119 | 0
100 | 156 | 10
500 | 177 | 35
1000 | 154 | 20
1500 | 144 | 15
2000 | 144 | 10
2500 | 135 | 4
3000 | 135 | 2

Maximum | 177 | 35
Minimum | 119 | 0
Average | 145.5 | 12

a) Number of frequent itemsets with various experiments

<table>
<thead>
<tr>
<th>Data Items = 1000</th>
<th>Data Items = 1500</th>
<th>Data Items = 2000</th>
<th>Data Items = 2500</th>
<th>Data Items = 3000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Itemset</td>
<td>YouTube Data</td>
<td>Random Data</td>
<td>YouTube Data</td>
<td>Random Data</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>20</td>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>2-Itemset</td>
<td>34</td>
<td>0</td>
<td>34</td>
<td>0</td>
</tr>
<tr>
<td>3-Itemset</td>
<td>41</td>
<td>0</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>4 or above</td>
<td>36</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
</tbody>
</table>

b) Details of various types of frequent itemsets produced

Figure A.2: Number of Frequent itemsets generated by Association Rules Mining algorithm