TRUST MANAGEMENT FOR P2P APPLICATION IN DELAY TOLERANT MOBILE AD-HOC NETWORKS

Basit I Qureshi

PhD

UNIVERSITY OF BRADFORD

2011

TRUST MANAGEMENT FOR P2P APPLICATION IN DELAY TOLERANT MOBILE AD-HOC NETWORKS

An Investigation into the development of a Trust Management Framework for Peer to Peer File Sharing Applications in Delay Tolerant Disconnected Mobile Ad-hoc Networks

Basit I Qureshi

Submitted for the degree of Doctor of Philosophy

School of Computing, Informatics and Media
University of Bradford

2011

TRUST MANAGEMENT FOR P2P APPLICATION IN DELAY TOLERANT MOBILE AD-HOC NETWORKS

Basit I Qureshi

Abstract

Security is essential to communication between entities in the internet. Delay tolerant and disconnected Mobile Ad Hoc Networks (MANET) are a class of networks characterized by high end-to-end path latency and frequent end-to-end disconnections and are often termed as challenged networks. In these networks nodes are sparsely populated and without the existence of a central server, acquiring global information is difficult and impractical if not impossible and therefore traditional security schemes proposed for MANETs cannot be applied.

This thesis reports trust management schemes for peer to peer (P2P) application in delay tolerant disconnected MANETs. Properties of a profile based file sharing application are analyzed and a framework for structured P2P overlay over delay tolerant disconnected MANETs is proposed. The framework is implemented and tested on J2ME based smart phones using Bluetooth communication protocol. A light weight Content Driven Data Propagation Protocol (CDDPP) for content based data delivery in MANETs is presented. The CDDPP implements a user profile based content driven P2P file sharing application in disconnected MANETs. The CDDPP protocol is further enhanced by proposing an adaptive opportunistic multihop content based routing protocol (ORP). ORP protocol considers the store-carry-forward paradigm for multi-hop packet delivery in delay tolerant MANETs and allows multi-casting to selected number of nodes. Performance of ORP is compared with a similar autonomous gossiping (A/G) protocol using simulations.

This work also presents a framework for trust management based on dynamicity aware graph re-labelling system (DA-GRS) for trust management in mobile P2P applications. The DA-GRS uses a distributed algorithm to identify trustworthy nodes and generate trustable groups while isolating misleading or untrustworthy nodes. Several simulations in various environment settings show the effectiveness of the proposed framework in creating trust based communities. This work also extends the FIRE distributed trust model for MANET applications by incorporating witness based interactions for acquiring trust ratings. A witness graph building mechanism in FIRE+ is provided with several trust building policies to identify malicious nodes and detect collusive behaviour in nodes. This technique not only allows trust computation based on witness trust ratings but also provides protection against a collusion attack. Finally, M-trust, a light weight trust management scheme based on FIRE+ trust model is presented.

Keywords: Distributed trust management; Delay tolerant disconnected MANETs; Peer to peer; Network simulation; Profile based file sharing.

Declaration of originality

I Basit Qureshi, declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given in the bibliography.

Acknowledgements

I am highly indebted to pass my heartfelt thanks to the many people who helped me in one way or another. First, I would like to acknowledge my sincerest, special, deep appreciation, and true thanks to my supervisors, Dr. Geyong Min and Prof. Demetres Kouvatsos. Without their support, guidance and most importantly their demand for high quality work, this thesis would have not been possible. Through many discussions they provided me with insightful technical suggestions and help me clarify many concepts.

Many thanks to my PhD supervision team Dr. Saad Bakry and Dr. Mohamed Tounsi who provided valuable suggestions and guidance on my research work. I would like to show my deep respect and gratitude to Dr. Mohammad I. Qureshi for his ideas in improving the presentation of this work. Many thanks to my PhD internal and external examiners; Dr. Pauline Chan and Dr. Kun Yang, for their comments on improving the presentation of this thesis.

I would also like to express by regards to Prof. Mohammad Ilyas for his inspiring and motivating talks and his valuable comments. I consider myself lucky to have the honour to discuss my work with Prof. Hamid Arabnia and Prof. Munther Dahleh and I thank them profoundly for their valuable guidance.

Finally I would like to thank my parents, especially my mom for her dedication and encouragement. She always stood by me and supported me. I would like to thank my wife for always being there when it mattered and my angels Myfrah and Khawla.

Basit Qureshi
Bradford, 2011

List of papers

All work presented here is the original work of the author unless otherwise indicated. Some parts of this thesis include revised versions of the following published papers by the author:

[1]. D.D. Kouvatsos, G. Min and B. Qureshi, "Performance issues in a secure health monitoring Wireless Sensor Network System", Proceedings of the 4th Euro NGi Heterogeneous Networks Conference HET-NET06, Ilkley, UK, pp. WP01 1-6, 11-13 Sept 2006.

[2]. M. Tounsi, B Qureshi, "A Bluetooth intelligent e-healthcare system: analysis and design issues", International Journal of Mobile Communications, Inderscience publishers, Vol. 6, No. 6, pp. 683-695, Nov/Dec, 2008.

[3]. B. Qureshi, G. Min, D. D. Kouvatsos, "A Content Driven Data Propagation Protocol for Mobile Social Network in disconnected MANETs" Proceedings of the International Conference on Wireless Information Networks and Systems (WINSYS 09), Milan, Italy, pp. 57-62, July 7-10, 2009.

[4]. B. Qureshi, M. Ilyas, G. Min, D. D. Kouvatsos, "Opportunistic Routing Protocol for content sharing in P2P Mobile Social Networks", Proceedings of the Sixth Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS 09), Toronto, Canada, pp. WP1-2, July 13 - 16, 2009.

[5]. B. Qureshi, M. Ilyas, G. Min, D. Kouvatsos, "An Adaptive Content Sharing protocol for P2P Mobile Social Networks" in the proceedings of the 6th International Symposium on Web and Mobile Information Services **(**WAMIS2010), in conjunction with 24th IEEE AINA, Perth, Australia, pp. 413-418, 20-23 April 2010.

[6]. B. Qureshi, G. Min, D. Kouvatsos, "A Framework for Trust Management in P2P Mobile Social Networks", in the proceedings of 10th IEEE International Conference on Computer and Information Technology (CIT2010), Bradford UK, pp. 567-574, 29 June – 1 July 2010.

[7]. B. Qureshi, G. Min, D. Kouvatsos, "Collusion prevention in FIRE+ Trust & Reputation model" in the proceedings of the 10th IEEE International Conference on Scalable Computing and Communications (Scalcom2010), Bradford UK, pp.2548-2555, 29 June – 1 July 2010.

[8]. B. Qureshi, G. Min, D. Kouvatsos, “Opportunistic Trust based P2P Services Framework for disconnected MANETs” in the proceedings of the 7th International Conference on Autonomic and Trusted Computing (ATC2010), Xian China, pp.151-165, October 26-29, 2010.

[9]. B. Qureshi, G. Min, D. Kouvatsos. “M-Trust: A Trust Management scheme for Mobile P2P Networks”, in the proceedings of the 8th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2010), Hong Kong, China, pp. 476-483. December 11-13, 2010.

[10]. B. Qureshi, G. Min, D. Kouvatsos, “Countering the Collusion Attack with a Multidimensional Decentralized Trust and Reputation Model”, to appear in the Journal of Multimedia Tools and Applications. Available online on 2 April 2011. DOI: 10.1007/s11042-011-0780-7.

Papers submitted:

[11]. B. Qureshi, G. Min, D. Kouvatsos, “Trusted Information Exchange in P2P Mobile Social Networks”, submitted to Wiley Concurrency and Computation: Practice and Experience.

[12]. B. Qureshi, G. Min, D. Kouvatsos, “Robust Trust Ratings aggregation in Mobile Peer-to-Peer Networks”, submitted to Elsevier Journal of Computer Communications.

Table of Contents

Chapter 1 Introduction 1

1.1 Overview 1

1.2 Motivation and Problem Statement 4

1.3 Aims and Objectives 6

1.4 Contributions 7

1.5 Thesis Organization 10

Chapter 2 Review of Basic Concepts 12

2.1 Mobile Networks 13
2.1.1 Mobile Ad Hoc Networks 13
2.1.2 Disconnected Delay Tolerant MANETs 16

2.2 Routing protocols 18
2.2.1 Routing protocols for MANETs 19
2.2.2 Routing protocols for DTN 22

2.3 Mobile P2P Networks 26
2.3.1 Challenges in deployment of P2P application on MANET 27
2.3.2 Existing P2P Overlays in MANETs 28
2.3.3 Mobile P2P Applications 30

2.4 Security in Mobile Networks 32
2.4.1 Types of Attacks on MANETs 33
2.4.2 Attacks prevention with Cryptography 34

2.5 Trust and Reputation Management in Mobile P2P Networks 37
2.5.1 Centralized Trust and Reputation Models 38
2.5.2 Decentralized Trust and Reputation Models 39

2.6 Analysis of Related Work 41
2.6.1 Framework design for P2P application in MANETs 41
2.6.2 Routing issues in DTNs 42
2.6.3 Vulnerabilities in TRM for Mobile P2P Networks 43

2.7 Summary 46

Chapter 3 Opportunistic Trust based P2P Framework in disconnected MANETs.......................47

3.1 Introduction 47

3.2 Framework Design 48
3.2.1 Application Layer 50
3.2.2 Communications Layer 52
3.2.3 Trust Management Layer 53

3.3 Framework implementation details 56
3.3.1 Framework Classes structure 56
3.3.2 OSCF protocol 58
3.3.3 P2P MSN File sharing application 60

3.4 Experimental setup and results 64
3.4.1 Opportunistic contacts 64
3.4.2 Trust based opportunistic contacts 66

3.5 Related Work *69*

3.6 Summary and Discussion *70*

Chapter 4 Content Driven Data Propagation Protocol (CDDPP)..73

4.1 Introduction *73*

4.2 CDDPP Protocol Design *74*

4.3 Simulation & Results *79*
4.3.1 Simulation Parameters 79
4.3.2 Communication Scenario 80
4.3.3 Message delivery rate vs document size 81

4.4 Summary *83*

Chapter 5 Opportunistic Routing Protocol (ORP) 86

5.1 Introduction *86*

5.2 Store-Carry-Forward Model in disconnected MANETs *86*

5.3 The Opportunistic Routing Protocol *88*
5.3.1 Application Component 88
5.3.2 Content Dissemination Component 89
5.3.3 Store-Carry-Forward Component 91

5.4 Simulation Environment *94*
5.4.1 Mobility Model 94
5.4.2 Simulation Parameters 95
5.4.3 Compared Protocols 96

5.5 Evaluation of ORP protocol *97*
5.5.1 Message propagation threshold 99
5.5.2 Document Size and message payload 100
5.5.3 Updating repository with Least Recently Used Algorithm 102
5.5.4 Forwarded documents 102
5.5.5 Mobility Parameters 104
5.5.6 Communication overhead 104

5.6 Summary *105*

Chapter 6 Dynamicity Aware Graph Re-Labelling Approach to Trust Management 108

6.1 Introduction *108*

6.2 Dynamicity Aware Graph Relabeling System *109*
6.2.1 Defining the Network 110
6.2.2 Labelling Vertices 110
6.2.3 DA-GRS Algorithm 110

6.3 Membership criterion in MSN *112*
6.3.1 Trust Requirements 112
6.3.2 Trust Computation 114

6.4 Algorithms for Trust management in MSN *116*
6.4.1 Modified Dynamicity Aware-Graph Relabeling System (DA-GRS) 117
6.4.2 Greedy Labelling 117
6.4.3 High Group Trust Labelling 118
6.4.4 Optimal Group Trust Labelling 118

6.5 Simulation & Results *119*

6.6 Discussion and summary *124*

Chapter 7 FIRE+ Model for Collusion-free Trust Management in disconnected MANETs.........125

7.1 Introduction *125*

7.2 FIRE trust and reputation model *126*
7.2.1 Trust Formula 128
7.2.2 Direct & Witness Interactions 128
7.2.3 Modes of Trust and Reputation 130

7.3 The Collusion Attack in disconnected MANETs *131*
7.3.1 Collusive behaviour in Target-Witness Interaction 132
7.3.2 Collusive behaviour in Witness-Witness Interaction 132

7.4 FIRE+ Trust Model *133*
7.4.1 Trust Variables 133
7.4.2 Defining History and Reputation Variables 135
7.4.3 Trusted agents network 136
7.4.4 Detecting Collusion Attack 137
7.4.5 Interaction Policies 139

7.5 Experiments and Results *141*
7.5.1 Dropped Connections for TA1 agents 143
7.5.2 Dropped Connections for TA2 agents 144
7.5.3 Number of Connections with Malicious Agents 146
7.5.4 Direct and Witness Trust Variables 146

7.6 Summary *149*

Chapter 8 M-Trust: A Trust Management Scheme for Mobile P2P Networks.........................150

8.1 Introduction *150*

8.2 Comparison of Existing Trust Ratings Aggregation Schemes *152*
8.2.1 Received Ratings 153
8.2.2 Weighted average 154
8.2.3 Bellman-Ford algorithm based scheme 154
8.2.4 Ultimate and total trust 155

8.3 M-trust Trust Ratings Aggregation Scheme *155*

8.4 Experiments and Performance Evaluation *157*
8.4.1 Initialization of simulation experiments 158
8.4.2 Congregation time and t_list size 158
8.4.3 Query hit rate and accuracy 160
8.4.4 Malicious peer detection rate with Mobility 162
8.4.5 Trust threshold confidence 162

8.5 Summary *163*

Chapter 9 Conclusions and Future Work..165

9.1 Conclusions *165*

9.2 Contributions *167*

9.3 Future Work *169*

References...171

Appendix A **186**
Random Waypoint Mobility Model (RWP) and Madhoc Simulator 186
Chapter 4 Results for CDDPP Protocol 189
Chapter 5 Results for Opportunistic Routing Protocol 191

Appendix B **196**
Chapter 6 Results for DAGRS Trust management Technique. 196
Chapter 7 Results for FIRE+ and FIRE trust models. 198

List of Figures

FIGURE 1-1: ORGANIZATION OF CHAPTERS 11

FIGURE 2-1: EXAMPLE OF A MANET 14

FIGURE 2-2: EXAMPLE OF A DISCONNECTED MANET. [HAIL08] 17

FIGURE 3-1: PROPOSED SYSTEM OVERVIEW 50

FIGURE 3-2: FRAMEWORK ARCHITECTURE 50

FIGURE 3-3: COMMUNICATION API 52

FIGURE 3-4: (A) A NETWORK FRIENDSHIP GRAPH WITH TRUST RATINGS (B) *T_LIST* FOR NODE A 55

FIGURE 3-5: PROTOTYPE CLASSES DIAGRAM 58

FIGURE 3-6: OSCF_PROTOCOL HEADER USED FOR COMMUNICATION IN THE FRAMEWORK 59

FIGURE 3-7: FRAMEWORK API INTERACTION WORKFLOW WITH P2P MSN APPLICATION 60

FIGURE 3-8: P2P MSN APPLICATION WORKFLOW DIAGRAM 61

FIGURE 3-9: INITIALIZE AND MAIN SCREENS 61

FIGURE 3-10: SEARCH AND VIEW SCREENS 63

FIGURE 3-11: SENDING AND REQUESTING DOCUMENTS 63

FIGURE 3-12: THE SUM AND RATIO OF SUCCESSFUL CONNECTIONS, MESSAGES AND FILES RECEIVED OVER TIME 67

FIGURE 3-13: INITIAL (LEFT) AND FINAL (RIGHT) FRIENDSHIP GRAPHS WITH AND WITHOUT TRUST COMPONENT. 68

FIGURE 3-14: TRUST MATRIX FOR TRUST RATINGS ($T_{I,J}$) IN (*T_LIST*) FOR ALL USERS 69

FIGURE 4-1: ILLUSTRATION OF A DISCONNECTED MANET. 74

FIGURE 4-2: CDDPP PACKET HEADER FORMAT 75

FIGURE 4-3: DOCUMENT LIST *DOC_LIST* FORMAT 76

FIGURE 4-4: TRANSMISSIONS FOR HOST N_I 77

FIGURE 4-5: NUMBER OF DOCUMENTS RECEIVED AGAINST SENT FOR CDDPP AND GREEDY PROTOCOLS 82

FIGURE 4-6: COMPARISON OF AVERAGE NUMBER OF REPOSITORY UPDATES 82

FIGURE 4-7: COMPARISON OF PACKETS DELIVERED OVER TIME BETWEEN CDDPP 84

FIGURE 4-8: COMPARISON OF DELIVERY RATE FOR CDDPP AND GREEDY PROTOCOLS 85

FIGURE 5-1: AN EXAMPLE OF STORE-CARRY-FORWARD IN DISCONNECTED NETWORKS 87

FIGURE 5-2: THE ORP PROTOCOL COMPONENTS 89

FIGURE 5-3: DATA FORWARDING IN THE ORP PROTOCOL 93

FIGURE 5-4: INTEREST PROFILES AND DOCUMENT TYPES 96

FIGURE 5-5: COMPARISON OF A/G AND ORP PROTOCOLS 100

FIGURE 5-6: DELIVERY RATE OF MESSAGES OVER MULTIPLE HOPS (ϕ) WITH 1024KB SIZE 101

FIGURE 5-7: DELIVERY RATE OF DOCUMENTS OVER MULTIPLE HOPS (ϕ) WITH VARIOUS DOCUMENT PAYLOADS..........................102

FIGURE 5-8: RATE OF REPOSITORY UPDATE OVER MULTIPLE HOPS (ϕ) WITH VARIOUS DOCUMENT PAYLOADS..........................103

FIGURE 5-9: COMPARISON OF AVERAGE NUMBER OF FORWARDED DOCUMENTS..........................105

FIGURE 6-1: GRAPH LABELING [CAST05]..........................111

FIGURE 6-2: FOUR RULES FOR THE DA-GRS ALGORITHM [CAST06]..........................112

FIGURE 6-3: NODES A IN GROUP X AND B IN GROUP Y MERGE IN TO GROUP Z 114

FIGURE 6-4: EXAMPLES OF A GROUP OF USERS IN A MSN 117

FIGURE 6-5: MERGING OF GROUPS. 118

FIGURE 6-6: PROPOSED ALGORITHMS FOR TRUST MANAGEMENT 119

FIGURE 6-7: SCREEN-SHOTS OF NETWORKS USED FROM THE MADHOC SIMULATOR..........................123

FIGURE 7-1: WITNESS REFERRAL PROCESS [HUYN06]..........................129

FIGURE 7-2: EXAMPLE OF DIRECT TRUST AND WITNESS BASED TRUST 130

FIGURE 7-3: COLLABORATIVE BEHAVIOR IN PROVIDING FALSE RECOMMENDATION VALUES 132

FIGURE 7-4: EXAMPLE OF TRUST GRAPHS FOR NODES A, B AND C IN THE NETWORK...........................138

FIGURE 7-5: COLLUSIVE BEHAVIOUR IN DIRECT INTERACTION WITH WITNESS RECOMMENDATIONS 138

FIGURE 7-6: COMPARISON OF AVERAGE NUMBER OF CONNECTIONS DROPPED FOR FIRE AND FIRE+..........................144

FIGURE 7-7: FINAL NETWORK DIAGRAM AFTER 200 TIME STEPS. FIRE, FIRE+ WITH TA1 AGENTS 145

FIGURE 7-8: COMPARISON OF AVERAGE NUMBER OF CONNECTIONS DROPPED FOR FIRE AND FIRE+..........................146

FIGURE 7-9: FINAL NETWORK DIAGRAM AFTER 200 TIME STEPS. FIRE, FIRE+ WITH TA2 AGENTS. 147

FIGURE 8-1: TRUST OVERLAY NETWORK..........................153

FIGURE 8-2: TRUST RATINGS AGGREGATION FOR WITNESS INTERACTIONS..........................154

FIGURE 8-3: COMPARISON OF TRUST MANAGEMENT TECHNIQUES WITH OUT-DEGREE D = 3, 6 AND 9. ...160

FIGURE 8-4: COMPARISON OF TRUST MANAGEMENT TECHNIQUES WITH D = 3, 6 AND 9...........................161

FIGURE 8-5: MALICIOUS PEER DETECTION RATE WITH MOBILITY..........................163

FIGURE 8-6: TRUST THRESHOLD VALUES FOR M-TRUST..........................164

List of Tables

TABLE 2-1: COMPARISON OF SOME PROACTIVE ROUTING PROTOCOLS FOR MANETS 20

TABLE 2-2: A COMPARISON OF SOME REACTIVE MANET PROTOCOLS 22

TABLE 2-3: SECURITY ATTACKS ON PROTOCOL STACKS [MERW07]....34

TABLE 3-1: CHARACTERISTICS OF COLLECTED DATA SETS 65

TABLE 4-1: PARAMETERS USED FOR SIMULATION....81

TABLE 5-1: PARAMETERS USED FOR A/G AND ORP COMPARISON....97

TABLE 5-2: ORP SIMULATION PARAMETERS 99

TABLE 5-3: AVERAGE DELIVERY RATES OF VARIOUS PAYLOADS OVER MULTIPLE HOPS 103

TABLE 5-4: EFFECT OF NUMBER OF TRAVELER NODES 104

TABLE 6-1: DEFINITION OF TRUST LEVELS IN NODES OF THE NETWORK....115

TABLE 6-2: PROPERTIES OF THREE SETS OF EACH CATEGORY OF NETWORKS 121

TABLE 6-3: AVERAGES OF GROUP AND ISOLATION COST FUNCTIONS FOR CAMPUS NETWORKS 122

TABLE 6-4: AVERAGES OF GROUP AND ISOLATION COST FUNCTIONS FOR SHOPPING MALL NETWORKS 122

TABLE 6-5: AVERAGES OF GROUP AND ISOLATION COST FUNCTIONS FOR CITY STREET NETWORKS....122

TABLE 7-1: PARAMETERS CONSIDERED FOR FIRE, FIRE+ COMPARISON 142

TABLE 7-2: AGENT TYPES AND SPECIFICATIONS 143

TABLE 7-3: COMPARISON OF FIRE AND FIRE+ IN TERMS OF AVERAGE NUMBER OF CONNECTIONS....148

TABLE 7-4: COMPARISON OF AVERAGE DIRECT INTERACTION TRUST (DIT) VALUES FOR FIRE+ 148

TABLE 7-5: COMPARISON OF AVERAGE WITNESS INTERACTION TRUST (WIT) VALUES FOR FIRE+....148

TABLE 8-1: SIMULATION PARAMETERS....159

Abbreviations in Alphabetic Order

AODV	Ad Hoc On-Demand Distance Vector
ARAN	Authenticated Routing for Ad-Hoc Networks
ATM	Asynchronous Transfer Mode
BAN	Body Area Network
BF	Bellman Ford algorithm
CA	Certification Authority
CAP	Cell Access Provider
CBR	Constant bit rate
CDDPP	Content Driven Data Propagation Protocol
CRL	Certificate Revocation List
DA-GRS	Dynamicity Aware Graph Re-labelling System
DHT	Distribute Hash Table
DoS	Denial of Service
DSDV	Destination-Sequenced Distance Vector
DSR	Dynamic Source Routing
DTN	Delay Tolerant Networks
ETSI	European Telecommunication Standard Institute
FAS	Foreign Authentication Server
FIFO	First Come First Serve
GPRS	General Packet Radio Service
GPS	Global Positioning System
GRS	Graph Re-Labelling System
GSM	Global System for Mobile communications
HAS	Home Authentication Server
HGT	High Group Trust
HiPeRLAN	High Performance Radio Local Area Network
HSDPA	High Speed Downlink Packet Access
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSec	Internet Protocol Security
IRTF	Internet Research Task Force

J2ME	Java 2 Platform
LAN	Local Area Network
MAC	Media Access Control
MAN	Metropolitan Area Network
MANET	Mobile Ad Hoc Network
MIT	Massachusetts Institute of Technology
MPR	Multi Point Relaying
MSN	Mobile Social Network
MT	M- Trust
ODR	Opportunistic DHT-based Routing
OGT	Optimal Group Trust
OLSR	Optimized Link State Routing
ORION	Optimized Routing Independent Overlay Network
ORP	Opportunistic Routing Protocol
OSCF	Opportunistic Store Carry Forward
P2P	Peer-to-peer
PAN	Personal Area Network
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PRNet	Packet Radio Network
PRoPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
QoS	Quality of Service
RFCOMM	Radio Frequency Communication
RR	Received Ratings
RREP	Route Reply
RREQ	Route Request
RWP	Random Way Point
SAODV	Secure Ad-hoc On-demand Distance Vector
SD	Sink Device
SEAD	Secure Efficient Ad hoc Distance vector
SPKI	Simple Public Key Infrastructure
SRP	Source Routing Protocol

SSL	Secure Sockets Layer
TA	Trusted Agent
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TRM	Trust and Reputation Management
TTL	Time to Live
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunications System
UT	Ultimate Trust
VRR	Virtual Ring Routing
WA	Weighted Average
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WEP	Wired Equivalent Privacy
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless LAN
ZRP	Zone Routing Protocol

Chapter 1

Introduction

1.1 Overview

Recently with the increase in mobile internet users, access to various mobile applications and services on the Internet has been growing at an enormous rate. Popular mobile web browsers such as Opera mini [OPER] running on mobile devices, show an exponential growth in terms of number of downloads. Internet based applications such as file sharing; social networking; health monitoring and security monitoring are finding ways of opera-ability in mobile environments. On the other hand Peer-to-peer (P2P) systems consisting of a dynamically changing set of nodes connected via the Internet, at the same time, have gained tremendous popularity. While initially conceived and popularized for the purpose of file sharing, P2P has emerged as a general paradigm for the construction of resilient, large-scale, distributed services and applications in the Internet [OLIV04]. P2P computing is a networking and distributed computing paradigm, which allows the sharing of computing resources and services by direct, symmetric interaction between computers. With the advancement in mobile wireless communication technology and the increasing number of mobile users, P2P computing, in both academic research and industrial development, has recently begun to extend its scope to address problems relevant to mobile devices and wireless networks. Popular online services such as facebook, myspace, you-tube etc have extended their services to the ubiquitous computing domain. A user may access the service from the provider over the mobile internet and can connect to friends, share content such as files, photos and videos while on the go. Other service providers such as gnutella, allows connection among users having mobile devices in a P2P environment without the need to connect to a central server. This extension of services into the mobile P2P domain provides a greater freedom to users of P2P services without the need of centralized servers. Mobile P2P applications allow a team or group to create new levels of ad hoc co-operation and collaboration around a specific, real-time goal. However, due to the difficulties caused by system mobility, wireless communications, limitations of pervasive devices and the ever-changing network topology, developing compelling and secure applications in mobile P2P environment is a challenge.

Mobile ad hoc networks (MANETs) are a network of mobile nodes connected together over a wireless medium. These nodes can freely and dynamically self-organize into arbitrary and temporary ad hoc network topologies, allowing people and devices to seamlessly internetwork in areas with no pre-existing communication infrastructure such as access points, wireless switches etc. MANETs have been deployed in disaster recovery and battlefield environments. Each node is able to communicate directly with any other node that resides within its transmission range and can use its neighbour nodes as relays to communicate beyond its transmission range without relying on a fixed infrastructure. Asynchronous communication is central and essential to support MANET operation [MASU09]. MANETs and P2P systems share a lot of key characteristics: self-organization and decentralization, and both need to solve the same fundamental problem: connectivity. Although it seems natural and attractive to deploy P2P systems over MANET due to this common nature, the special characteristics of mobile environments and the diversity in wireless networks bring new challenges for research in P2P computing.

It is possible for large scale MANETs to become disconnected when, for example, the mobile hosts that compose the network are very sparsely or irregularly distributed. The whole network then appears as a collection of distinct “islands”. Communication between hosts that belong to the same island is possible, but no temporaneous communication is possible between hosts that reside on distinct islands [HAIL08]. Disconnected MANETs have been called as challenged networks [DALY10] and Delay Tolerant Networks (DTNs). A DTN provides interoperable communications with and among challenged environments [JAIN04]. A challenged network is defined as a network that has one or more of the following characteristics: high end-to-end path latency; end-to-end disconnection meaning a path between a node pair may never exist; limited resources or limited life expectancy either due to lack of battery power, such as in sensor networks, or node damage as may occur in battlefield deployments. Such networks may never have an end-to-end path from source to destination at a given time.

Security is an essential service for wired and wireless network communications. The success of MANET strongly depends on whether its security can be trusted. However, the characteristics of a MANET pose both challenges and opportunities in achieving the security goals, such as confidentiality, authentication, integrity, availability, access control, and non-repudiation. The mobile hosts forming a MANET are normally mobile devices with limited physical protection and resources. Security modules, such as tokens and smart cards, can be used to protect against physical attacks. Cryptographic tools are widely used to provide powerful security services, such as confidentiality, authentication, integrity, and non-repudiation. However, cryptography requires a central authority to share and distribute public / private keys, but in case of MANET a node cannot be guaranteed to be available at all times [YUNF07]. Also, strong cryptography often demands a heavy computation overhead and requires the auxiliary complicated key distribution and trust management services, which mostly are restricted by the capabilities of physical devices (e.g. CPU or battery). The characteristics and nature of MANET require the strict cooperation of participating mobile hosts. A number of security techniques have been invented and a list of security protocols has been proposed to enforce cooperation and prevent misbehaviour, such as 802.11 Wired Equivalent Privacy (WEP), Internet Protocol Security (IPSec), Secure Efficient Ad hoc Distance vector (SEAD), Secure Ad-hoc On-demand Distance Vector Protocol (SAODV), Secure Routing Protocol (SRP), Authenticated Routing for Ad-Hoc Networks (ARAN), Secure Sockets Layer (SSL), and so on. However, none of those preventive approaches is perfect or capable to defend against all attacks.

Recently trust management approaches have gained the attention of researchers for MANET's security. As an important concept in network security, trust is interpreted as a set of relations among nodes participating in the network activities [RAMC04] [LIMC08]. Trusted relationships among nodes in a network are based on different sources of information such as direct interactions, witness information and previous behaviours of nodes. Trust management in distributed and resource-constraint networks, such as disconnected mobile ad-hoc networks and sensor networks, is much more difficult but more crucial than in traditional hierarchical architectures, such as the Internet and access point centred wireless LANs. Generally, this type of distributed network has neither pre-established infrastructure, nor centralized control servers or trusted third parties. The dynamically changing topology and intermittent connectivity of disconnected MANETs establish trust management more as a dynamic systems problem [BARA05]. In early stages of trust and security on MANETs several researchers relied on authentication, cryptographic encryption and decryption techniques. These schemes were shown to be effective in providing security; however these are based on centralized certification authorities. Significant communication overheads from both pre-processing and during processing periods, as well as energy consumption were major challenges thus rendering these approaches to be poor for DTNs. It has been shown recently that reputation based techniques are more effective in de-centralized mobile networks [MERW07] [PIYA08] [LUOA09] [BALA07] [SRIV06] [SALE09].

1.2 Motivation and Problem Statement

This thesis provides an investigation into the development of a trust management framework for a P2P file sharing application in delay tolerant disconnected MANET. P2P file sharing applications such as bit torrent and gnutella, when allowed to run on ubiquitous devices in pervasive environments, give unacceptable performance results. This is primarily due to the fact that these P2P applications are designed to operate on fixed networks, and therefore do not take into account the issues of mobile computing and wireless communications. To this end a new P2P file sharing application is developed that primarily runs in ad hoc mode and allows users to establish connection based on profile matching. The application running on connected devices transfer / update the profile and exchange files. These files are stored on the device within the limits of storage space and forwarded to other devices as contact opportunities arise. These opportunistic exchanges combined with human mobility create a temporal communications network as in Pocket Switched Network (PSN) [SUJ07] where messages travel from device to device over multiple hops without any infrastructure connectivity reminiscent of a delay tolerant MANET. The main advantage of using this design is that, application developers can rely on the application framework for security, trusted user discovery, interaction among users and file sharing.

Routing in delay tolerant MANETs is challenging because these networks may never have an end-to-end path from source to destination at a given time. Due to the existence of long delay paths, frequent disconnections and network partitions, information may be carried by a mobile node and forwarded opportunistically across partitions, therefore allowing communication between areas of the network that are never connected by an end-to-end path. The Bundle [FALL03] and PRoPHET [LIND03] protocol, enable indirect data exchange among disconnected portions of the overall network, using a store-and-forward approach. [XUE09] improved the PRoPHET protocol by using average delivery predictabilities. SimBet Routing presented in [DALY07] exploits the exchange data based on bridge nodes. These protocols allow data transfer between nodes in a DTN, but do not address the issue of content based opportunistic forwarding. In a mobile P2P file sharing application, users typically share content among users with similar interests defined in user profiles. This social information is of importance when considering development of an opportunistic delay tolerant routing protocol. The social information can be used to discover optimal paths in routing that can reduce the overhead of routing therefore improving the overall performance [HUIP08]. A new light weight opportunistic protocol is presented for content based store carry forwarding in DTNs. The protocol considers social information when routing packet in the network. This protocol is further extended to allow data communication over multiple hops. The proposed protocols are light weight and use multi-casting techniques based on social information to reduce overall traffic in the network.

Mobile nodes enable indirect data exchange among disconnected portions of the overall DTN. To assume trustworthy interaction in this kind of networks is unrealistic due to the fact that most entities in the network are unknown. Trust management in a de-centralized P2P network is a challenging task in the absence of a lack of global knowledge for all users; any trust / reputation parameters for a user have to be computed locally [HUYN06] [SERE07]. Dynamicity Aware Graph Relabeling System (DA-GRS) presented in [CAST06] is used to develop a framework for trust management in P2P mobile file sharing application. DA-GRS allows users to be labelled with trust ratings that can increase or decrease based on number of completed transactions and ratings received from other users. The goal is to create communities/groups of users with high trust ratings while identifying untrustworthy users and isolating them from the community of users. The developed framework is effective in creating trusted communities of users by determining trust ratings for users. However for the system to work; it is assumed that all participating users are trustworthy in their interactions. In real-time systems this assumption is invalid because users may choose to be trustworthy in some interactions and untrustworthy in others. Moreover, it is also possible that an untrustworthy user can collaborate with trustworthy users to develop a positive reputation and in later interactions, provide false information. A popular distributed and de-centralized trust management system, FIRE [HUYN06], fail to address the issue of colluding malicious nodes in the network. In this study, FIRE is extended to FIRE+ by incorporating a graph for trusted agents, direct / witness reputation and various policies to counter the collusion attack. FIRE+ also defines a confidence variable to identify malicious nodes. Results prove that FIRE+ is successful in detecting colluding nodes and therefore the collusion attack.

In an open and decentralized P2P environment, peers do not have any centralized authority to maintain and distribute reputation information. A full-aggregation reputation system calculates the reputation score of a peer by considering the opinions from all other peers who have interacted or non-directly interacted with this peer. Usually a full aggregation reputation system is of high accuracy. However, the aggregation approach involves a trade-off between the accuracy and overload. To ensure trustworthiness in Mobile P2P trust management systems, the popular trust management schemes including the received ratings aggregation [LIMC08], weighted average of ratings [HUNY06], Bellman ford based algorithm [ZHAO09], total trust and ultimate trust schemes [BAHT10] are thoroughly investigated and compared. Based on the analytical results, an efficient, accurate, robust and scalable light weight trust ratings aggregation scheme, referred to as M-trust, for mobile P2P networks is presented. The extensive comparison with other schemes shows that M-trust possesses the excellent overall performance in terms of accuracy, reliability, convergence speed, rate of detecting malicious peers under various constraints of mobility, trust threshold and network out-degree.

1.3 Aims and Objectives

Traditional MANET protocols fail to deliver due to the sparse population and intermittently connected nodes in disconnected MANETs. DTN routing protocols considering the store-carry-forward paradigm for data transmission have to be developed for the P2P applications. The aim of this research work is to develop a trust based P2P framework for a file sharing application in a delay tolerant disconnected MANET. The P2P application developed in this work possesses characteristics that are similar to a mobile social networking application. The similarities include, content based routing rather than destination oriented routing of packets, file sharing using the store-carry-forward paradigm and both can be implemented as an overlay on a MANET. Nonetheless, the focus of this work is towards P2P application development and attention will be given to P2P issues in mobile networks.

Security and trust management being a critical issue, one other aim of this work is to investigate a distributed de-centralized trust and reputation model that is not computational intensive, considering the many limitations of mobile devices in disconnected delay tolerant MANETs. The trust model must be multi-dimensional, built on trust ratings from reputed neighbouring nodes and must consider direct and witness interaction. Since trust ratings from various nodes can be modified and tampered with, a set of policies need to be developed to counter collaborative behaviour among malicious nodes and to avoid impending collusion attacks. Moreover, due to the decentralized nature of DTN, the scheme for acquiring reputation information from direct and witness interactions and aggregating the received trust ratings to compute trust values must be robust, accurate and reliable. To achieve these aims, the following objectives are considered:

- To develop a trust based framework for a P2P content based file sharing application. The framework has to be tested in an environment with user interactions to gain insight into the routing and trust management issues in data transfer over a DTN.
- To develop light weight content driven data propagation protocol for data transfer using store carry forward in a DTN.
- To extend CDDPP into a multi-hop opportunistic content driven routing protocol for data transfer using store carry forward in a DTN.
- To develop a framework for building trust based communities in a mobile P2P network utilizing DA-GRS. The framework must be distributed, de-centralized and must use a trust model rely on trust ratings from neighbouring nodes in the network.
- To extend FIRE, a popular trust and reputation management system for de-centralized distributed networks in to FIRE+. FIRE+ addresses the weakness of FIRE by providing solution for detecting false ratings, collaborating nodes and collusion attack.
- To develop a robust and efficient trust ratings aggregation scheme for use in a DTN.

1.4 Contributions

This research work addresses the problem of trust management in P2P applications over an underlying delay tolerant disconnected MANET. A generic framework for P2P applications based on trust management and opportunistic routing mechanism in a disconnected MANETs is presented. Users can share content and transfer files in an opportunistic manner utilizing store-carry-forward paradigm. The framework was implemented in J2ME Personal Profile and tested on mobile Personal Digital Assistant (PDA) devices using Windows Mobile 6.0. In experimental setup for testing with user trials the successful construction of communities between nodes that contact each other opportunistically in close proximity and ad hoc manner was demonstrated. The framework also implements a light weight trust model to identify trustable and untrustworthy users based on social contacts.

Based on results obtained the underlying opportunistic protocol and trust management modules are modified and improved. A light weight CDDPP is developed for opportunistic content based data delivery in disconnected MANETs. The protocol is further improved to address multi-hop data dissemination and routing in the adaptive Opportunistic Routing Protocol (ORP). ORP considers a disconnected MANET where nodes can communicate based on user interests (content based data delivery) to distant nodes in a multi-hop communication model. Due to the frequent disconnection in these kinds of MANETs, the opportunistic approach to data delivery is considered. The nodes simply do not just forward the messages and data to adjacent nodes but also store them. The stored messages or data can be transmitted to intended destinations once such a chance occurs. ORP is defined with three components including application component, content dissemination component and content store & forward component. Simulations with various parameters such as mobility model, repository sizes, mobility of nodes, data delivery over multiple hops, document sizes and payloads etc. have been done to study the effects of performance of ORP. Performance of ORP is compared to a similar opportunistic content driven routing protocol, Autonomous Gossiping (A/G) algorithm, presented in [DATT04].

A trust based approach to membership management in a disconnected MANET utilizing the DA-GRS presented in [CAST06] is an adaptation of the Graph Relabeling Systems (GRS) to the paradigm of dynamic and self-organizing networks. DA-GRS is a model invented for the conception and the analysis of decentralized applications and algorithms targeting dynamically distributed environments like disconnected MANETs. In the context of this study, DA-GRS approach allows a way of designing a decentralized algorithm for constructing and maintaining a graph of trusted nodes in disconnected MANETs, relying on a careful rule-based token management. The goal of the DA-GRS algorithm is to create groups of nodes with strong trust values and isolate nodes with poor trust values. Two cost functions are proposed to compute and update trust and isolation values. These two cost functions are utilized in the three greedy approach based algorithms presented to create groups with high trust values. Simulations are carried out to prove the effectiveness of the proposed algorithms compared to the original DA-GRS algorithm in different context environments.

FIRE [HUYN06] is a completely de-centralized trust model as it integrates up to four types of trust and reputation from different information sources, according to availability: interaction trust, role-based trust, witness reputation, and certified reputation. In this research work FIRE+ is proposed as an extended version of FIRE trust and reputation model [HUYN06], for decentralized distributed networks such as disconnected MANETs. This work addresses the vulnerability of FIRE model to collusion attack from a group of malicious nodes. A multidimensional model based on direct and witness trust interaction for detecting collusion attack is proposed. FIRE+ defines a mechanism for periodically detecting the confidence in direct and witness information received from recommending nodes and storing it in a rating history database for identifying collaborative behaviour in recommendations. Based on this information trust aware nodes can use policies to reduce the level of encountered risk of an attack.

To ensure trustworthiness in Mobile P2P trust management systems, this work presents the effectiveness of various distributed and decentralized trust ratings aggregation schemes on DTN. To this end, the popular trust schemes including the received ratings aggregation [LIMC08], weighted average of ratings [HUNY06], Bellman ford based algorithm [ZHAO09], total trust and ultimate trust schemes [BAHT10] are thoroughly investigated and compared. Based on the analytical results, an efficient, accurate, robust and scalable light weight trust ratings aggregation scheme, referred to as M-trust, for P2P mobile networks is presented. A trust ratings aggregation algorithm is proposed that acquires trust ratings not only from direct recommendations but also from recommendations from distant nodes. Results obtained from extensive simulations show that this proposed method can decrease the time required to compute the list of trust ratings and reduce the required storage space. The extensive comparison with other schemes shows that M-trust possesses the excellent overall performance in terms of accuracy, reliability, convergence speed, rate of detecting malicious peers under various constraints of mobility, trust threshold and network out-degree.

1.5 Thesis Organization

In this chapter the background to the proposed research has been described along with the motivation and need for this work. The aims and objectives are mentioned and a list of contributions is presented. The rest of the thesis is organized as follows.

Chapter 2 reviews the basic concepts in delay tolerant disconnected MANETs. Various classes of routing protocols developed have been discussed. Several kinds of security attacks on MANETs are presented with various approaches recently proposed for defence against the attacks. Work done in the area of trust management for delay tolerant MANETs is also presented.

Chapter 3 proposes a trust based framework for P2P applications in disconnected MANETs. The design and experimental test-bed for user trials is discussed in detail. The results are carefully analyzed and the shortcomings in the design are identified. Further work into improvement of the framework takes two directions. The first direction involves development of an opportunistic content driven routing protocol that is discussed in chapters 4 and 5. The second direction leads research into trust management into mobile P2P systems which is discussed in chapters 6, 7 and 8. Figure 1.1 shows the flow of information and the link between chapters.

Chapter 4 provides a light weight Content Driven Data Propagation Protocol that utilizes the store-carry-forward mechanism for data delivery in disconnected MANETs. The CDDPP protocol is further improved by incorporating multi-hop relays for data transfer in an adaptive Opportunistic Routing Protocol (ORP). The ORP routing protocol discussed in chapter 5 considers the store-carry-forward paradigm for multi-hop packet delivery in delay tolerant MANETs and allows multi-casting to selected number of nodes.

In chapter 6, a framework based on Dynamicity Aware Graph Re-labelling System (DA-GRS) for trust management mobile P2P file sharing application is presented. The framework utilizes a set of greedy distributed algorithms to identify trustworthy nodes and generate trustable communities while isolating misleading or untrustworthy nodes.

Chapter 7 provides FIRE+, a multi-dimensional model for trust management. FIRE+ is an extension of FIRE trust model and incorporates direct and witness based users' ratings, a witness graph building mechanism to detect collusive behaviour and a set of policies to prevent collusion attack.

Chapter 8 presents M-trust: a trust ratings aggregations scheme based on FIRE+ trust model. M-trust includes a trust ratings aggregation algorithm that acquires trust ratings not only from direct recommendations but also from recommendations from distant nodes. Results obtained from extensive simulations show that M-trust can decrease the time required to compute the list of trust ratings and reduce the required storage space.

Chapter 9 concludes the research work with a list of contributions and future research directions.

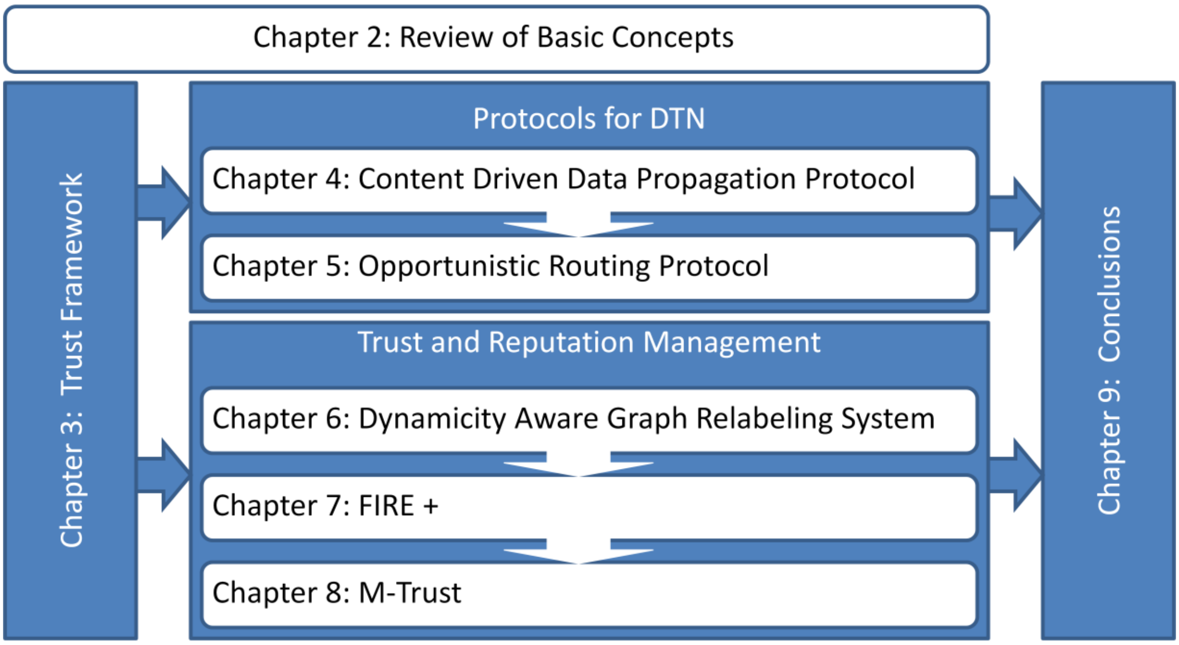

Chapter 2: Review of Basic Concepts
Chapter 3: Trust Framework
Protocols for DTN
Chapter 4: Content Driven Data Propagation Protocol
Chapter 5: Opportunistic Routing Protocol
Trust and Reputation Management
Chapter 6: Dynamicity Aware Graph Relabeling System
Chapter 7: FIRE +
Chapter 8: M-Trust
Chapter 9: Conclusions

Figure 1-1: Organization of Chapters

Chapter 2
Review of Basic Concepts

Peer-to-Peer (P2P) computing is a networking and distributed computing paradigm, which allows the sharing of computing resources and services by direct, symmetric interaction between computers. With the advance in mobile wireless communication technology and the increasing number of mobile users, P2P computing, in both academic research and industrial development, has recently begun to extend its scope to address problems relevant to mobile devices and wireless networks. MANETs and P2P systems share a lot of key characteristics including self-organization and decentralization; both need to solve the same fundamental problem: connectivity. Although it seems natural and attractive to deploy P2P systems over MANET due to this common nature, the special characteristics of mobile environments and the diversity in wireless networks bring new challenges for research in P2P computing.

Ad hoc networks represent complex distributed systems comprised by wireless nodes that can freely and dynamically self-organize into arbitrary and temporary (ad hoc) network topologies, allowing communications in areas with no pre-existing infrastructure. The ad hoc network paradigm is not a new concept, since it was proposed many years ago mainly for tactical networks [DALY10]. Recently, the introduction of enabling technologies, such as Bluetooth [BLUE] and Wi-Fi, has allowed the deployment of commercial ad hoc networks outside the military domain, generating a renewed and growing interest in the research and development of such networks.

This chapter provides an overview of the delay tolerant ad hoc networking paradigm, protocols and design constraints. Issues in P2P systems deployment on ad hoc networks are presented. Security attacks on MANETs are discussed followed by existing trust and reputation management techniques for P2P applications deployed on MANETs. Furthermore, a summary at the end of the chapter provides discussion on the challenges of trust management in P2P applications for MANET environments.

2.1 Mobile Networks

In recent years, wireless communication technologies have developed rapidly. Many different kinds of technologies exist for various applications and many are coming in near future. From cellular networks of 70s, satellite communication, end user wireless networks and ad hoc networks have come to age leading way to wireless sensor networks and personal and body area networking.

The increased popularity of mobile computing and communication devices, such as cell phones, laptops and handheld digital devices such as Personal Digital Assistants (PDAs), means that wireless networks are increasingly the most convenient solution for interconnection in many usage scenarios. Since the early 2000s mobile devices have been getting smaller, cheaper and more convenient to carry, with the ability to run applications and connect to network services [LEHR02]. Currently, most of the connections among wireless devices are achieved through fixed infrastructure service providers or private networks. For example, since the 1980s mobile phones have been connected by cellular networks, and the connection of laptops to the Internet via wireless access points has grown rapidly in popularity in the early 2000s [IBNK04]. Current developments, such as 3G and 4G phones, show little signs of change in this trend. While infrastructure-based networks provide an effective mechanism for mobile devices to get network connectivity, setting up the necessary infrastructure can be time consuming and incurs potentially high costs. There are situations where networking connections are not available in a given geographic area, and providing connectivity and network services in these situations becomes a real challenge. Examples range from wildlife tracking and habitat monitoring sensor networks, military networks, inter-vehicle communication, disaster response networks, and inter-planetary networks to nomadic community networks. For this reason, alternative ways to deliver services in disconnected environments have been emerging. Two such areas include MANETs which arose in the 1990s, and more recently Delay-Tolerant intermittently connected MANETs also known as DTNs which were first introduced in 2001.

2.1.1 Mobile Ad Hoc Networks

MANETs are collections of mobile nodes connected together over a wireless medium. These nodes can freely and dynamically self-organize into arbitrary and temporary ad hoc network topologies, allowing people and devices to seamlessly internetwork in areas with no pre-existing communication infrastructure (e.g., disaster recovery and battlefield environments). Each node is able to communicate directly with any other node that resides within its transmission range and can use its neighbour nodes as relays to communicate beyond its transmission range without relying on a fixed infrastructure. Asynchronous communication is central and essential to support MANET operation [MASU09].

MANETs do not depend on centralized administration, rather each node acts as an independent router and typically also as an application node, generating and receiving application data. As such, network management is distributed across the nodes. Fig. 2.1 shows an example of multi-hop routing [ALCH08]. In the scenario, node *a* is out of direct communication range with node *c*, but can communicate with node *c* by using node *b* as an intermediary. Node *b* received messages from node *a* and forwards the messages to node *c*.

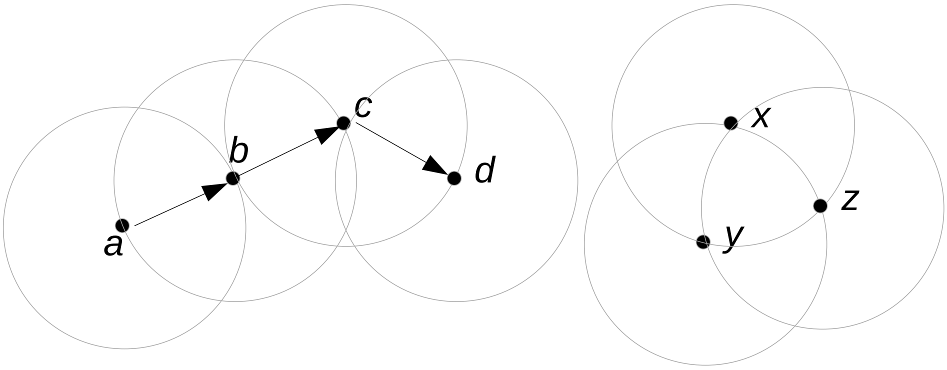


Figure 2-1: Example of a MANET

Despite having many interesting features, ad hoc networks inherit all the traditional problems of wireless communications and wireless networking:

- The wireless medium has neither absolute nor readily observable boundaries outside of which nodes are always unable to communicate;
- The wireless medium is unprotected from outside signals;
- The wireless medium has time-varying and asymmetric propagation properties;
- Hidden-terminal and exposed-terminal phenomena may occur.

Beside these issues, the ad hoc networking adds a number of specific characteristics and design constraints [CORS99]:

- Multi-hop routing. Every node acts as a relay and forwards neighbours' packets to enable communications beyond the coverage area.

- Self-organization and infrastructure-less. Each node operates in distributed P2P mode, acts as an independent router and generates independently data. All the network services have to be distributed across different nodes.
- Heterogeneity. Each node may be equipped with one or more wireless interfaces with different communication capabilities, resulting in possible asymmetric links. In addition, each node might have a different software/hardware configuration, resulting in variability in processing capabilities.
- Network scalability. Ad hoc network applications can involve large networks, as it happens in sensor and tactical networks [FREE01]. Although scalability is critical to the successful deployment of these networks, many challenges have still to be solved [HONG02].
- Transient network topology. Since nodes can move arbitrarily, the network topology may change frequently and unpredictably, resulting in route failures and frequent network partitions.
- Energy constrained operation. Because batteries carried by each mobile node have limited power supply, processing power is limited, which in turn limits services and applications that can be supported by each node.

Various technologies can be used as building blocks for constructing multihop ad hoc networks. Based on the coverage area, Body Area Networks (BANs); Personal Area Networks (PANs); Local Area Networks (LANs); Metropolitan Area Networks (MANs) and Wide Area Networks (WANs) have been used in the literature to construct MANETs [CONT03].

A body area network has to provide the connectivity among wearable devices, i.e. computing devices placed on the user body, therefore the typical communicating range of a BAN corresponds to the human body range, i.e. 1-2 meters. Personal area networks connect mobile devices carried by users to other mobile and static devices. While a BAN must assure the interconnection of one-person wearable devices, a PAN is a network composed by devices of several persons along with some environmental devices. Therefore, the communicating range is typically up to 10 meters. Wireless LANs (WLANs) have a communication range typical of a single building, or a cluster of buildings, that is 100-500 meters. A WLAN should satisfy the same requirements typical of any LAN, including high capacity, full connectivity among attached stations, and broadcast capability.

Currently, two main technologies have emerged for ad hoc wireless networks: the Bluetooth specifications (Bluetooth, Internet) for BANs/PANs and the IEEE 802.11 standard for WLANs [IEEE802.11]. In addition to these standards, the European Telecommunication Standard Institute (ETSI) has promoted the High Performance Radio Local Area Network (HiperLAN) [ETSI] family of standards for WLANs. Among these, the most interesting standard for WLAN is HiperLAN/2, which achieves data rates ranging from 6 to 54 Megabits/s and supports both infrastructure-based and ad hoc configurations. Along with HiperLAN, different standards have been proposed in the last years, i.e. ZigBee [IEEE802.15] and WiMAX [IEEE802.16].

Routing in a MANET is a challenging task, however many routing protocols for MANETs have been proposed. Section 2.3 details various protocols developed over the past few years.

2.1.2 Disconnected Delay Tolerant MANETs

Disconnected MANETs have been called as challenged networks and Delay-Tolerant Network. A DTN provides ‘interoperable communications with and among challenged environments’ [JAIN04]. A challenged network is defined as a network that has one or more of the following characteristics: high end-to-end path latency; end-to-end disconnection meaning a path between a node pair my never exist; limited resources or limited life expectancy either due to lack of battery power, such as in sensor networks, or node damage as may occur in battlefield deployments. Such networks may never have an end-to-end path from source to destination at a given time.

A MANET can become disconnected when, for example, the mobile hosts that compose the network are very sparsely or irregularly distributed. The whole network then appears as a collection of distinct “islands”. Communication between hosts that belong to the same island is possible, but no temporaneous communication is possible between hosts that reside on distinct islands. Figure 2.2 shows a disconnected MANET. This MANET is composed of a number of laptops carried by users, which can move in and between buildings (for example, the buildings of a campus). In this example, some laptops are temporarily isolated (either because there is no other laptop within their transmission range, or more simply because they have been put in suspended mode for a while), while other laptops have a number of neighbours, with which they can try to communicate using either single-hop or multi-hop transmissions.

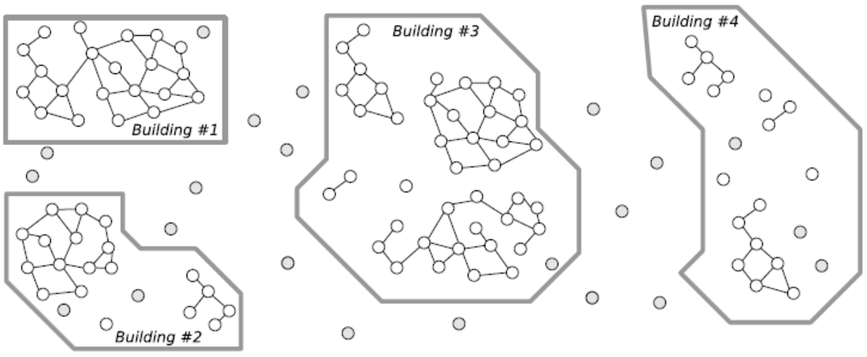


Figure 2-2: Example of a disconnected MANET. [HAIL08]

The routing problem in DTNs can be described as 'where messages are to be moved end-to-end across a connectivity graph that is time-varying but whose dynamics may be known in advance' [JAIN04]. The Delay-Tolerant Network Research Group [DTNRG] has proposed architecture to support messaging in delay-tolerant applications. The architecture presented in [FALL03] consists of an overlay, called the bundle layer. A bundle is defined as a number of messages to be delivered together. DTN nodes implement the bundle layer which forms an overlay that employs persistent storage to overcome network interruptions. The bundle layer stores and forwards bundles between DTN nodes. The bundle layer is situated below the application layer and above the transport layer, thus allowing environment-specific underlying protocols.

The challenge for routing protocols in DTN is to achieve the best delivery ratio with the available information about the network. Messages are buffered using a store-and-forward mechanism, where the data is physically carried through the time-varying network graph. These challenged environments are characterized by their disconnected nature where continuous end-to-end connectivity cannot be assumed. As a result, they suffer from long or variable delay times, asymmetric data rates and high error rates. The disconnected nature and the lack of end-to-end connectivity between nodes, means that the communication must be delay-tolerant. Daly E. et.al [DALY10], detail the various categories of challenges faced by delay tolerant disconnected MANETs.

Recently solutions to routing problem in DTNs have been presented. One of the basic solutions is epidemic routing [VAHD00], where messages are blindly stored and forwarded to all neighbouring nodes generating a flood of messages. The drawback of epidemic dissemination lies in the very high number of messages which are needed to obtain successful delivery to the right recipient. Other solutions have been proposed to tackle the problem of routing in (possibly mobile) DTNs, based on the previous knowledge of the routes of the potential carriers [JAIN04] [ZHAO04] [SARA06] or on probabilistic approach [SPYR05]. More recently, researchers [CALE08] [HAIL08] [CHAI09] and [MUSO08] have tried to take advantage of opportunistic routing paradigm. The opportunistic and collaborative routing protocols exploit the time-variant nature of the network topology to provide connectivity for sparse topologies usually by resorting to the store-carry-forward paradigm.

The store-carry-forward paradigm requires broadcast of messages to neighbouring nodes in order to opportunistically deliver the messages. One of the objectives of the work presented in this thesis is to develop a content based opportunistic routing protocol for DTNs. Among many benefits of using content based store-carry-forward approach is to reduce the flooding in the network. This is achievable when select nodes are chosen from neighbouring nodes that share similar interests. The social networking theory implies that the users having similarity of interests share similar types of content [CHAI08] [DALY07]. Authors of [MUSO09] have shown in their work that routing in a DTN can be done by taking advantage of social contacts of nodes. Intermediate nodes with similar social interests, work as multi-point relays to forward packets opportunistically to other nodes. Section 2.3.6 discusses routing protocols for disconnected delay tolerant MANETs.

2.2 Routing protocols

Many protocols have been designed in the last few years in order to support destination-driven routing in MANETs. The major measures that are used to evaluate routing protocols are: network size and structure, routing accuracy, and routing overhead. A good routing protocol can find a good trade-off between routing accuracy and routing overhead. This section presents protocols for MANETs and DTN.

2.2.1 Routing protocols for MANETs

Routing protocols for MANET can be classified into various categories including proactive routing protocols, reactive routing protocols and hybrid routing protocols.

Proactive Routing Protocols: These routing protocols calculate the routes to all the destinations before a transmission actually happens. These protocols calculate the routing table even when there is no packet to send. The benefit of calculating routes beforehand is the short latency in finding a route. The drawback is that to maintain routes for each destination, the nodes have to keep exchanging routing messages even when there is no traffic at all. One of the oldest routing protocols for MANET is Destination Sequences Distance Vector (DSDV) protocol [PERK96]. DSDV is a modified version of the classical Bellman Ford Routing protocol. For a destination, DSDV's routing table keeps the next hop, the metric (the hop count), and a sequence number, which is generated by the destination to mark the freshness of the route. A node periodically broadcasts its whole routing table or modifications to its routing table to its immediate neighbours. For each route, the routing update carries a new sequence number which is originally given by the destination node. Upon receipt of a new route for a destination, if there is no route for the destination yet, a node adds the route together with the sequence number to its routing table. If a route already exists, the node picks the route with a greater sequence number. If a route with the same sequence number already exists, the node picks the route with a better metric.

Jacquet et.al. [JACQ01] proposed a link state routing algorithm that can eliminate many unnecessary link state message broadcasts using a method called Multi Point Relaying (MPR). In addition, the amount of link state transmitted can also be reduced by only advertising the MPR selectors of a node. Every wireless node maintains a list of its immediate neighbours through periodic beacon messages. Neighbouring wireless nodes exchange their neighbour lists through HELLO messages. These HELLO messages work like link state routing messages. Every node thus knows the two hop topology around itself. Every node picks a set of one hop neighbours to cover all of its two hop neighbours. These sets of immediate neighbours are called MPR nodes. Every node tells its immediate neighbours whether they are chosen as MPR nodes for it. This is also implemented using HELLO messages. Upon receipt of a link state routing message, a node checks if it has been chosen by the sender as its MPR node. If true, the node re-broadcasts the link state message. Only the nodes that are chosen by some nodes as their MPR nodes generate link state messages. The link state messages only contain the nodes that choose them as MPR nodes. These sets of nodes are called MPR selectors. Using the Dijkstra algorithm, the route to every single destination can be calculated. The OLSR routing protocol is very popular and has become IETF RFC 3626 [CLAU03]. The greatest strength of the protocol is that the flooding overhead can be greatly reduced.

Xu K. et al. designed a hierarchical routing architecture (H-LANMAR) [XUK03] for large MANETs (on the order of a few thousand nodes) based on the structure of the Internet. The routing for the hierarchical network uses a modified version of LANMAR [PEIG00], a routing protocol for flat MANETs. LANMAR is a routing protocol used in situations where groups of wireless nodes move together.

Redi et.al. proposed a complete architecture, JAVeLEN, for low power consumption MANET [REDI06]. The architecture mainly targets two problems, power management in the link layer and efficient power-aware routing. It is especially suitable for large scale sensor networks. Table 2.1 summarizes some proactive routing protocols.

Table 2-1: Comparison of some Proactive Routing Protocols for MANETs

Name	Network Size	Network Structure	Route Update	Routing Overhead	Power Awareness
DSDV	Small	Flat	1 - hop	Medium	No
OLSR	Large	Flat	Multi-point Relay	Low	No
JAVeLEN	Large	Flat	Multi-point Relay	Low	Yes
LANMAR	Large	Hierarchical	Subnet Routing	Low	Only 2 radios

Reactive routing protocols: These routing protocols calculate the route to a destination only when it’s necessary for a transmission. The basic idea of reactive routing protocols is to find the route to a destination only when necessary. By eliminating the periodic routing updates, these routing protocols are aiming at reducing the routing overhead. These routing protocols assume that the network is not very big and the nodes’ rate of motion is moderate. Johnson et.al. proposed Dynamic Source Routing (DSR), a reactive source routing protocol for MANET [JOHN96]. DSR is a source routing protocol. When a node tries to send a packet to a destination, it checks to see if there is a source route available in its route cache. If so, it attaches the route to the packet and sends it out. The packet is forwarded by the nodes specified in the route; otherwise, a route discovery process starts.

The benefits of DSR are its simplicity and its support on directed networks. The problems of it are, flooding is costly, a whole route has to be rebuilt even when a single link is broken and the use of route cache can put a limit on the size of the network supported by DSR. Ad hoc On-demand Distance Vector (AODV), proposed by [PERK99] is very similar to DSR. The most important difference is that, instead of storing the complete routes, only the node that sent the last message is stored. The last hop node is used as the next hop toward the originating node. AODV also uses a “Ring Search Algorithm” to reduce the flooding overhead.

Authors in [SINH01], proposed a solution solving the high cost of flooding query messages in reactive ad hoc routing protocols. DSR and AODV are two of such protocols. The paper uses the result of an earlier paper [SIVA99] on constructing a minimal set of nodes who can communicate with all other nodes in a MANET, a minimal dominating set. The nodes elected to the dominating set are called “core nodes”. The core nodes are at most three hops away from each other. A communication tree can be constructed among the core nodes by exchanging beacon messages in the network. A beacon message is like a link state routing message carrying the list of core nodes connected to the source node. A beacon message travels at most two hops. Using the beacon messages, a core node can find a route to any other core node in its 3 hop neighbourhood. Now, the QUERY messages in DSR and AODV are not broadcast any more. Instead, they are sent to their neighbouring core nodes using unicast. By using unicast, IEEE 802.11 ACK and RTS-CTS mechanisms can be used to alleviate conflicts. Only the core node that is directly connected to the QUERY destination needs to forward the QUERY message to it. Therefore, the message overhead can be reduced greatly. This paper presents a better solution than MPR in OLSR for reducing the broadcast cost of the QUERY based routing protocols. The key to this solution is CEDAR, the distributed dominating set election mechanism designed in [SIVA99]. This mechanism would actually be useful for any protocol that requires flooding messages throughout the whole network. Table 2.2 shows a comparison of reactive MANET protocols discussed in this section.

Hybrid routing protocols: These routing protocols combine proactive routing and reactive routing. Proactive protocols response is quicker but they have a higher routing overhead. They are more suitable for fast changing, larger sized MANETs. Reactive protocols are more suitable for small sized, less dynamic MANETs. Hybrid routing protocols try to combine the benefits of both of them. ZRP [HAAS02] divides a network into zones from the point of view of each single node. In ZRP, a node propagates its proactive routing message (distance vector) to nearby nodes within a fixed number of hops (a routing zone). The limit on the hop count is called zone radius, a critical parameter of ZRP. Hence, each node has complete routing information about every single node within its routing zone. When the “zone radius” is 1, the protocol becomes a pure reactive routing protocol. When the zone radius is the radius of the network, the protocol becomes a pure proactive routing protocol. An optimal radius needs to be found to get the best trade-off.

Table 2-2: A comparison of some reactive MANET protocols

Name	Network Size	Network Structure	Route Discovery	Route Stored
DSDV	Small	Flat	Flooding	Route Cache
AODV	Small	Flat	Flooding	Next hop
CEDAR	Large	Hierarchical	Core Forwarding	N/A

2.2.2 Routing protocols for DTN

Disconnected MANETs are a class of Ad hoc networks where node density is low, and contacts between the nodes in the network do not occur very frequently. As a result, the network graph is rarely, if ever, connected and message delivery must be delay-tolerant. Traditional MANET routing protocols such as AODV [PERK99], DSR [JOHN96], DSDV [PERK96] and ZRP [HAAS02] makes the assumption that the network graph is fully connected and fails to route messages if there is not a complete route from source to destination at the time of sending. For this reason traditional MANET routing protocols cannot be used in disconnected MANETs. To overcome this issue, node mobility is exploited to physically carry messages between disconnected parts of the network. These schemes are sometimes referred to as *mobility assisted routing* that employs the *store-carry-and-forward* model. Mobility-assisted routing consists of each node independently making forwarding decisions that take place when two nodes meet. A message gets forwarded to encountered nodes until it reaches its destination.

The earliest form of replication-based routing is epidemic, discussed in [VAHD00], where transmitted data is continuously replicated until all nodes receive a copy. In particular, when a node receives a new packet, it first checks whether it is the final destination of the packet, and if not, it multicasts the received packet to every other node it shares a link with.

In this context, all messages generated by a source node are delivered to all nodes in the network and eventually, the receiver. If a path towards the receiver exists, then epidemic routing guarantees that all messages will be successfully delivered, without spending any time for communication purposes prior to each transmission. However, epidemic routing has the main drawback of wasting valuable network resources, especially in space communications where resources are scarce. The constant flow of data packets in the network will inevitably lead to buffer overflow and loss of data. The A/G algorithm presented in [DATT04], utilizes the epidemic algorithm to spread data items selectively based on vulnerability of other nodes (multicasting), instead of treating all nodes homogeneously and flooding the network.

Probabilistic Routing Protocol using a History of Encounters and Transitivity (PRoPHET) for disconnected DTNs is presented in reference [LIND03]. PRoPHET is used for intermittently connected networks, where there is no guarantee that a fully connected path between source and destination exists at any time, rendering traditional routing protocols unable to deliver messages between hosts. Based on the history of encounters, the PRoPHET protocol predicts the delivery of messages for each node. If a node has been reached recently its delivery predictability is increased, on the contrary if two nodes have not encountered each other for a long time period, an aging factor is used to lower the delivery probability. Moreover PRoPHET also seeks nodes that can function as relays for other nodes by computing the frequency of encounters. Another important parameter that affects PRoPHET performance is HelloTimer, which defines the frequency that a node informs its neighbours of its existence. The lower the value of HelloTimer is, the faster a node is discovered after a link outage.

PRoPHET is a completely autonomous routing protocol since no management is required; available links between nodes are dynamically discovered and previous knowledge is used for planning future transmissions. Moreover, opportunistic contacts are utilized as well. An important drawback of PRoPHET routing, however, is its inability to support priorities and, as a result, to provide any form of Quality of Service. In this context, all data packets are handled equally and no special treatment can be applied to urgent data. Most important, PRoPHET routing consumes considerable amount of both energy and time for message exchange prior to each transmission.

Very recently [XUE09] improved the PRoPHET protocol by using average delivery predictabilities. SimBet Routing presented in [DALY07] exploits the exchange data based on bridge nodes. Some bridge nodes are identified based on their centrality characteristics, i.e., on their capability to broker information exchange among otherwise disconnected nodes. Nodes are not required to exchange information about the entire network topology, but only locally available information is considered.

Content Driven routing protocols for DTN: Content-based routing protocols are intrinsically data-centric. Data-centric networking protocols use content addressing instead of host (e.g., IP) addressing. Data-centric routing is intrinsically different from host based routing in that data is routed based on users' specified interests [MOTT08]. A number of protocols have been designed in the last few years in order to support destination-driven routing in disconnected MANETs [PELU06]. In contrast content-based communication in such networks has not justified much research so far. Many papers about content-based communication have already been published, but these papers consider either stable, wired networks, or fully connected MANETs [COST06]. They usually propose to construct and maintain content-based *routing structures* in order to forward messages efficiently between publishers and subscribers. [COST06] describes an approach whereby a content-driven multi-hop routing structure (limited to a given horizon) is built around each host. A utility-based function is used to select the best carriers and/or forwarders for each kind of message, and mobile carriers are meant to transport messages between non-connected parts of the network. [PELU06] present an opportunistic approach to data forwarding in DTNs. Messages are forwarded to a number of potential carrier nodes that physically move to connect to previously disconnected nodes and deliver the messages.

[YONE04] proposed a content-based publish/subscribe system for MANETs, which integrates an extended ODMRP (On-Demand Multicast Routing Protocol) [CHIA99] and content based subscriptions. ODMRP supports optimized data dissemination mechanisms with context awareness including location, network topology, network ability and mobility. To optimize construction of an event dissemination structure, the proposed system defines an interface to apply the context from a publish/subscribe system to ODMRP. The context is summarized subscriptions and notifications. The interface is generic to supply data to be attached to the ODMRP packet and indicate how to process them. Content-based subscriptions at a broker node are aggregated and summarized and the event source broker node defines the multicast group by examining the propagated subscriptions. Context-awareness allows both middleware and network layer components to exploit information to provide an efficient and dynamic event routing mechanism for better performance.

[MUSO08] proposed Context Aware Routing (CAR) protocol for intermittently connected MANETs. CAR presents an approach to delay tolerant MANET routing which uses prediction to allow the efficient routing of messages to the recipient. A host willing to send a message to a recipient, or any host in the multi hop path to it, uses a Kalman Filter prediction and multi-criteria decision theory to choose the best next hop (or carrier) for the message. The decision is based on the mobility of the host (a highly mobile host is a good carrier as it meets many hosts) and its past collocation with the recipient. CAR does not assume any previous knowledge of the routes of the hosts and is based on a single copy of the message in the system, instead of having multiple replicas.

Opportunistic Store-Carry-Forward Routing in DTN: Recently very large MANETs that are intermittently connected and are delay tolerant have received a great attention from researchers. In this kind of MANETs it is possible to have multiple regions of clusters of nodes that are intermittently connected. Two kinds of routing is required, inter-regions and intra-regions. The collaborative and opportunistic routing class exploits both the temporal diversity and the broadcast nature of the wireless propagation, usually by resorting to broadcast communications instead of traditional unicast ones, to provide connectivity in presence of hostile wireless propagation conditions [PELU06]. Delay tolerant disconnected MANETs are a typical application domain for opportunistic routing, since they try to provide connectivity to sparse topologies usually by resorting to a so-called store-carry-forward paradigm [ZHU07].

In their pioneer work [BISW05], the authors suggest to broadcast the packets and to select the next forwarder at the receiver side to take advantage by all the opportunities provided by the wireless propagation. In other words, they exploit spatial diversity, which can assure more resilience to lossy links. Since such a routing, referred to as opportunistic routing, allows several nodes to receive the same packet, the authors single out a sub-set of neighbour nodes, namely a candidate set, allowed to forward the packet to limit the network flooding.

NOMAD presented in [MUSO08] addresses the multi-region routing problem using store-carry-forward nodes in the network. NOMAD utilizes the PRoPHET [LIND03] to dissipate data packets in the intra-region. Mobile nodes can carry and forward packets of data from one region to another while having physically moved to the new region. This approach is similar to the data-mules project presented in [JEAD05] where nodes can carry data and move to another location, dissipating the stored information. NOMAD also addresses deterministic and probabilistic approaches to message delivery using various kinds of multicast messages.

The work presented in this thesis extends the concept of opportunistic content driven routing in DTNs. The Content Driven Data Propagation Protocol (CDDPP) presented in chapter 4 is a light weight protocol that exploits the store-carry-forward mechanism when possible to forward data packets to one-hop nodes having similar content types. The proposed protocol considers identifying nodes in a network based on identities. User defined identities with personal profiles that if matched would lead to communication in a social context. Socially aware users can participate in storing and carrying messages and files from one location to another and forwarding the message should an opportunity arise. An extension of the CDDPP protocol, referred to as Opportunistic Routing Protocol (ORP) is presented in chapter 5. The ORP protocol extends CDDPP to multi-hop packet transmission over a DTN. In comparison to a popular A/G algorithm ORP performs better.

2.3 Mobile P2P Networks

Recently, P2P systems consisting of a dynamically changing set of nodes connected via the Internet have gained tremendous popularity. While initially conceived and popularized for the purpose of file sharing. P2P has emerged as a general paradigm for the construction of resilient, large-scale, distributed services and applications in the Internet [OLIV04].

P2P systems are defined as self-organizing, decentralized distributed systems that consist of potentially untrusted, unreliable nodes with symmetric roles [TANG04]. The self-organization, decentralization, diversity, and redundancy inherent in the approach are relevant to a large class of applications beyond file sharing, anonymity, and anti-censorship. The P2P paradigm has largely adopted a layered approach. A P2P overlay network built on top of the Internet provides a general-purpose layer that provides many common properties desired by distributed applications, such as self-organization, decentralization, diversity, and redundancy. Such an overlay shields distributed application designers from the complexities of organizing and maintaining a secure overlay, tolerating node failures, balancing load, and locating application objects.

P2P overlay networks in the Internet and mobile wireless ad hoc networks share many key characteristics such as self-organization and decentralization due to the common nature of their distributed components [WAN04]. Due to the P2P nature of MANETs, all protocols designed for MANETs are inherently P2P. Examples include multi-hop routing protocols (e.g., DSR and AODV). Existing studies have effectively proposed a Mobile P2P overlay abstraction [PUCH06] [WUJ05], i.e. borrowing the topologies and objection location techniques of Internet P2P overlays developed in the Internet and supporting them in MANETs. However in MANETs, due to the dynamic nature and fast-changing topology of physical network may be a significant problem from P2P point of view. As the underlying physical network keeps changing, it is hard for an overlay P2P network to maintain an optimal or reasonable topology.

2.3.1 Challenges in deployment of P2P application on MANET

Many fundamental differences between the Internet and a MANET pose challenges to implementing P2P applications in MANETs, including:

1. Bandwidth limitation. Unlike the wired Internet, MANETs have lower network capacity due to the use of wireless channels. This limits the usability of P2P protocols that have high message overhead.
2. Node mobility. In the Internet, the topology of a P2P overlay changes at a large time scale. On the other hand, in a MANET, limited transmission range and node mobility results in frequent topology changes. This places pressure on P2P applications constructed in MANETs to update the overlay topology much more frequently to maintain the matching between the overlay topology and the underlying network topology.
3. Lack of infrastructure. Certain P2P protocols make use of some infrastructure components in their designs. For example, a P2P routing protocol may assign node identifiers based on locations determined from static landmarks to improve routing performance. These techniques may not be usable in MANETs due to the lack of any static infrastructure.

4. Limited energy. Most P2P applications in the Internet are not designed to operate with minimum message transmissions. In an energy-limited environment such as a MANET, it may be very important for nodes to reduce the number of message transmissions while keeping the performance acceptable.
5. Addressing. Nodes in a MANET are likely to disconnect and reconnect to the network many times. Although no specific addressing architecture has been standardized for MANETs, it is plausible to assume that nodes will have changing IP addresses over time. This could challenge structured P2P protocols that store logical to physical address (nodeID-to-IP) mappings in their routing tables.
6. In P2P applications deployment over MANETs, it is impossible to rely on a central authority for security due to the dynamic nature of the network. Traditional trust and reputation management systems require global knowledge of the network which is impossible to acquire in MANETs due to the ever changing topology. Any trust and reputation management system for P2P applications deployed in MANET environments must be de-centralized and should acquire trust information from immediate neighbours.
7. De-centralized trust and reputation management systems require trust information from peers. It is possible that peers would provide false information. Peers can also collude with malicious peers to promote or demote trustworthy peers. Moreover it is challenging to identify a peer as trustworthy based trust information provided by others; reputation based on positive or negative interactions can also provide a reliable account of trust history.

2.3.2 Existing P2P Overlays in MANETs

Klem A. et.al. proposed integrating a Gnutella-like P2P application into the network layer [KLEM03] and compared it to a layered design similar to that of [OLIV03]. Optimized Routing Independent Overlay Network (ORION) is a P2P file sharing application that allows the setup of on-demand overlay connections that closely match the physical topology of the underlying MANET. When a query for a data item arrives, ORION employs one-hop broadcast to contact all its physical neighbours in one transmission. ORION combines the P2P operation with routing techniques from AODV. The results of the study indicate that the integrated overlay abstraction design has significantly lower overhead compared to the layered design while achieving better performance according to application-specific metrics.

DPSR [PUCH06] integrate a pastry-like [ROWS01] structured P2P protocol with the DSR routing algorithm, while CROSS-Road [DELM05] integrates a Pastry-like DHT over the OLSR routing algorithm, and VRR [CAES06] proposes a routing algorithm which provides indirect routing by resorting to a Pastry-like structure too. All these techniques associate an identifier, namely a key, to each peer by means of a hash function and organize the keys in a ring structure. Since the identifiers are randomly assigned to peers, the P2P overlay topology is usually built independently from the physical one, and thus no relationship exists between overlay and physical proximity. As shown in [RIPE02], this implies that overlay hops can give rise to physical routes which are unnecessary long. MAD-Pastry [TAKE08] integrates the Pastry protocol with the AODV routing algorithm and tries to overcome this issue by resorting to clustering. However, the overlay and physical proximity are in some way related only for inter-cluster communications.

[REPA05] utilized distributed hash tables (DHTs) and proposed adaptive content-driven routing and data dissemination mechanisms in mobile P2P networks. DHTs are a class of decentralized distributed mechanisms that provide a lookup service similar to a hash table; (key, value) pairs are stored in the DHT, and any participating node can efficiently retrieve the value associated with a given key. Under this mechanism nodes build and maintain content summaries of their data and adaptively disseminate them to the most appropriate peers. A peer can then use these summaries to determine if one of its peers can provide the requested data or services. Hence, peers choose to maintain summaries of other peers' content, in order to be able to efficiently locate needed information. Therefore, this protocol always propagates the queries to those peers that have a high probability of providing the desired results. This content-driven routing mechanism can efficiently find objects in large-scale, unstructured P2P network.

[CALE08] proposed a DHT-based routing protocol, Indirect Tree-based Routing (ITR). The ITR integrates both traditional direct routing and indirect key-based routing at the network layer. For both direct and indirect routing, each node maintains a unique routing table which stores only physical 1-hop neighbours, i.e. only peers with which the node can communicate at the link layer. As a result, each overlay hop consists of only one physical hop, limiting the message overhead and avoiding the redundancy.

2.3.3 Mobile P2P Applications

Many popular applications running on the internet have recently been migrated to Mobile networks. File Sharing and Social Networking are a few P2P applications that have been recently considered.

Mobile P2P File sharing. P2P file sharing systems account for a high percentage of the traffic volume in the fixed Internet, having exceeded http (www) or email traffic. The increasing availability of mobile data networks such as GPRS and UMTS in conjunction with attractive pricing schemes makes P2P file sharing an interesting application in the mobile context. But the operation of P2P systems in mobile environments encounters several problems, such as a relatively narrow and expensive air interface, highly varying online-states (presence) of the subscribers, a hierarchical network structure, and limited device capabilities.

Klem A. et.al in [KLEM03], present a mobile P2P file sharing application, Optimized Routing Independent Overlay Network (ORION). ORION comprises of an algorithm for construction and maintenance of an application-layer overlay network that enables routing of all types of messages required to operate a P2P file sharing system, i.e., queries, responses, and file transmissions. ORION is built to include the routing tables and route updating and forwarding mechanisms defined in reactive MANET protocols such as DSR and AODV. Additionally [KLEM03] use their own file transfer protocol. A file is split into equal-sized blocks prior to transfer. A file is fetched block by block by the querying node. This allows for parts of files to be fetched from different nodes based on the current network conditions. Because TCP is not used, ORION incorporates its own packet scheduling and loss-recovery mechanisms. File blocks can arrive out of order as long as one copy for each block is received.

[ANDR04] proposed architecture for P2P file sharing application. An example of earlier work on Mobile P2P file sharing applications is a mobile client for gnutella and can be found in [CONT05].

Mobile Social Networking. Online social networks have exploded in popularity very recently [ZIVN06]. Social networks provide a variety of mechanisms for users to share rich sets of contextual data with other users, including searching for other users with similar interests, as well as a means to establish and maintain communication with other users. *Mobile social networking is social networking where one or more individuals of similar interests or commonalities, conversing and connecting with one another using the mobile phone* [BEAC08]. Much like web based social networking, mobile social networking occurs in virtual communities. Recent implementations of mobile social networks from popular social network sites such as Facebook [FACE] and Myspace rely on Internet, Email and short messaging service on the client's device. To search for a friend in the social network a user needs to subscribe to the service and query the database, residing on service provider's servers, for possible friends with common interests. A subscriber's mobile device when connected to the Internet, searches and downloads the requested content thus requiring the subscriber to stay connected to the Internet at all times while communication is in progress.

In a social network, users subscribe to the service by making a public profile. A profile is designed to introduce a person to other members of the network announcing personal information, interests, location and a list of documents to share. If a user makes a search, his personal interests are matched in a database and query results are returned. The user may choose to select from a number of interested users and send an invite. The invited user receives the invitation message, if interested the user responds and the two users become friends. Friends can show their documents publicly and may even share them. A user announces his documents to a friend, if the friend is interested he can request a document. Researchers in [EAGL06], [LUGA07] and [RAEN05] discuss implementation of various forms of social networks. Typically three factors are essential to successful data sharing in a social network, Interest Profiles, Document Lists and Document Repository.

<u>Interest Profiles:</u> Each user maintains a list of keywords describing his interests. These keywords are used for searching and indexing purposes. An interest profile can be detailed and may even contain both texts as well as graphics data and therefore it can take increasing amount of storage allocation. However for the proposed protocol it is assumed that an interest profile would be a collection of keywords only and therefore would take minimal amount of storage.

<u>Document List</u>: A document list is a list of documents stored at a host. A document list consists of certain attributes of documents stored in the repository. These attributes include but are not limited to a Unique Identifier for the document, Document size, Document type, ownership and a Timestamp. Each document stored in the document repository has

this information. Document size is mentioned in bytes. Document type could be categories of documents such as image, video, text or object etc. Ownership is the MAC address of a device. A timestamp is the date and time for the document creation and indicates when the document was last updated. A list of documents is announced whenever two users with similar interests decide to share.

Document Repository: Each node maintains a document repository for documents to be shared. Since there are limits to the number of documents stored in a host depending on the availability of sufficient storage area, therefore limits are set on the size of the repository.

Mobile social networking provides various challenges at two levels. At the network communications level, many limitations of providing social networking service to users connected to a mobile network exist. Frequent disconnections due to power exhaustion, poor signal quality and mobility hinders the quality of service for any mobile application. Knowing the network features such as throughput and delay can help mobile social networks select a user to which the network route has the best performance. This leads to the so-called wireless-aware social networks. Much work has been done in providing quality of service and performance evaluation of routing protocols for MANETs.

At the second level, there are also social-aware or social inspired wireless networks where the knowledge of social network users is exploited for the benefit of wireless network design. Researchers in [DALY07] present a social network analysis for routing in disconnected delay tolerant MANETs. References [HUIP08] and [RAEN05] presented methods for detecting community behaviour in DTNs, exploiting the benefit of store and forwarding data in socially interactive users. Authors in [HUIP08] present a novel technique determining the impact of human mobility on the design of opportunistic forwarding algorithms in DTNs.

2.4 Security in Mobile Networks

Security is an essential service for wired and wireless network communications. The success of a mobile network strongly depends on whether its security can be trusted. However, the characteristics of a mobile network pose both challenges and opportunities in achieving the security goals, such as confidentiality, authentication, integrity, availability, access control, de-centralization and non-repudiation. Typically mobile hosts form a MANET with mobile devices having limited physical protection and resources.

There are a wide variety of attacks that target the weakness of MANET. For example, routing messages are an essential component of mobile network communications, as each packet needs to be passed quickly through intermediate nodes, which the packet must traverse from a source to the destination. Malicious routing attacks can target the routing discovery or maintenance phase by not following the specifications of the routing protocols. There are also attacks that target some particular routing protocols, such as DSR, or AODV. More sophisticated and subtle routing attacks have been identified in recent published papers, such as the black-hole (or sinkhole) [HUY04], Byzantine [AWER02], and wormhole [HUY02] [SAZI02] attacks. In the terminology of information system security, a risk exists if there is vulnerability and a threat. Vulnerability is the opportunity to cause damage. A vulnerability of an information system may be caused by a logical design flaw (e.g., a badly designed protocol), an implementation flaw (e.g., a buffer overflow), or a fundamental weakness (e.g., passwords and cryptographic keys that can be guessed). A threat arises from an attacker trying to find and exploit the vulnerability in order to inflict damage. Damage may also be caused by an incidental, non-intentional exploitation of vulnerability [STAL02]. A number of security techniques have been invented and a list of security protocols has been proposed to enforce cooperation and prevent misbehaviour, such as 802.11 WEP, IPSec, SEAD, SAODV, SRP, ARAN, SSL, and so on. However, none of those preventive approaches is perfect or capable to defend against all attacks [ZOUR06] [KERR09].

2.4.1 Types of Attacks on MANETs

The attacks in MANET can roughly be classified into two major categories, namely passive attacks and active attacks, according to the attack means [YIS04]. A passive attack obtains data exchanged in the network without disrupting the operation of the communications, while an active attack involves information interruption, modification, or fabrication, thereby disrupting the normal functionality of a MANET. Examples of passive attacks are eavesdropping, traffic analysis, and traffic monitoring. Examples of active attacks include jamming, impersonating, modification, denial of service (DoS), and message replay. Attacks can also be classified according to network protocol stacks. Table 3.1 shows an example of a classification of security attacks based on protocol stack; some attacks could be launched at multiple layers. Following paragraphs discuss some of the attacks on MANETs that have been identified and heavily studied in recent research work.

Table 2-3: Security Attacks on Protocol Stacks [MERW07]

Layer	Attacks
Application layer	Repudiation, data corruption
Transport layer	Session hijacking, SYN flooding
Network layer	Wormhole, blackhole, Byzantine, flooding, resource consumption, location disclosure attacks
Data link layer	Traffic analysis, monitoring, disruption MAC (802.11),WEP weakness
Physical layer	Jamming, interceptions, eavesdropping
Multi-layer attacks	DoS, impersonation, replay, man-in-the-middle

IEEE 802.11 incorporates wired equivalent privacy (WEP) to provide WLAN systems a modest level of privacy by encrypting radio signals. It is well known that WEP has a number of weaknesses and is subject to attacks [STAL02] [BORI01] [KARY02].

Attacks targeting the route discovery process have been discussed in references [LOU03] and [HUT04]. Some attacks also target data packet forwarding functionality in the network layer. Researchers in [PAPA03] study the vulnerability of attacks on packet forwarding mechanism in MANET protocols. Wormhole attack has been extensive studied in [ILYA03] [SAZI02]. In a wormhole attack an attacker records packets at one location in the network and tunnels them to another location. Routing can be disrupted when routing control messages are tunnelled. Denial of service (DoS) attacks could be launched from several layers. An attacker can employ signal jamming at the physical layer, which disrupts normal communications. At the link layer, malicious nodes can occupy channels through the capture effect, which takes advantage of the binary exponential scheme in MAC protocols and prevents other nodes from channel access. At the network layer, the routing process can be interrupted through routing control packet modification, selective dropping, table overflow, or poisoning. At the transport and application layers, SYN flooding, session hijacking, and malicious programs can cause DoS attacks.

2.4.2 Attacks prevention with Cryptography

Confidentiality, integrity, authentication, and non-repudiability are achieved by cryptographic methods. Cryptographic algorithms are employed for secure data storage and for secure transmission. For secure data transmission involving more than one party, the algorithms must be embedded in cryptographic protocols which define the sequence of steps to be undertaken by the participating parties. Most access control systems rely on public key management systems to certify an association between an identity and a key in form of a *digital certificate*. These certificates contain the public key and the identity along with other details cryptographically signed by a trusted third party.

The two main public-key [PUBK] management solutions are Pretty Good Privacy (PGP) [ZIMM95] and the X.509 public key infrastructure [Public key infrastructure, Internet]. PGP has an anarchic organization in contrast to a rigid hierarchy of X.509. In PGP though there are some central certificate repositories these are not much used. In X.509 there is a hierarchy of Certification Authorities (CA) which is responsible for the issuing of certificate and their verification. A node verifies the authenticity of a certificate by using the public key of the CA. The CA may revoke a certificate and periodically release a Certificate Revocation List (CRL) containing references to the revoked certificates. Delays in the release of a CRL may lead to the acceptance of some revoked certificates by nodes in the network. In ad hoc networks this approach is difficult to operate as access to a CA cannot be guaranteed at all times to obtain the latest CRL. In PGP a certificate's trustworthiness is assigned by the user using it. This process is made difficult in PGP as most of the certificates are self-signed and their trustworthiness needs to be verified by the user. The process to estimate the trustworthiness of a certificate may be prolonged and difficult in an ad hoc network. The key management approaches for ad hoc networks try to eliminate the need for a centralized CA (Public key infrastructure, Internet). The first approach described below emulates a conventional CA by distributing it on several nodes. In the second approach each node authenticates the other using some prefixed criteria, while in the last approach a self-organized public-key infrastructure is used.

Distributed Certification Authority: Researchers in [ZHOU99] have proposed a key management scheme for ad hoc networks using threshold cryptography and the public key paradigm. The scheme provides for distribution of parts of the secret key among some special ad hoc nodes designated as *servers*. An attacker has to break into a *threshold number* of servers in order to get access to the secret key of the service. To prevent progressive compromise of servers *share refreshing* is done periodically. This scheme requires prior communications and coordination between the nodes for setting up the service. Also, in this scheme some nodes (namely the *servers*) will have to work more than other nodes. Furthermore the requirement for each server to know the public key of all nodes is difficult if the number of nodes in the ad hoc network is large.

Pre-arranged Shared Secret: This approach is based on the existence of a shared secret among the nodes in the ad hoc network. Individual nodes in the network use the shared secret to generate their respective keys. One such scheme proposed in [DECL01] has a hierarchical framework. Each area in the hierarchy has a controller. These area controllers re-key a node when it moves between different “areas”. Another scheme proposed by [KONG02], uses the emulation of certification authority and shared secret model along with a Public Key Infrastructure (PKI) based centralized model. Initially the scheme has an aerial node acting as the centralized node for key distribution. If this aerial node is destroyed the scheme uses threshold cryptography based on secret sharing to emulate a distributed certification authority.

Self-Organized Public-Key Infrastructure: [HUBA01] proposed a public-key distribution based trust building scheme for ad hoc networks which is similar to the PGP *web of trust* concept. The scheme differs from PGP as there are no central certificate directories for distribution of certificates. Instead a user selects a subset of certificates from its repository to disclose to the other user. Both the users then merge the received certificates with their own certificates. In order to find the public-key of a remote user the local user makes use of the Hunter Algorithm [HUBA01] on the merged certificate repository to build certificate chain(s). A certificate trust chain should lead from the local user certificate to the remote user’s certificate. The local user uses the public-key contained in the remote user’s certificate. The probability of finding such a certificate chain in this scheme is high but is not guaranteed. This decentralized scheme leads to disclosure of too much information about the originating node as it releases several unnecessary certificates, which may not be needed in chain formation. There are two other certificate types, capability and property certificates. An identity certificate merely binds names to keys, while a capability certificate has embedded authorizations in it allowing the owner (client) to perform certain authorized actions on resources of the issuing server. The third and most generalized type of certificate is the property-based certificate. A property based certificate has the ability to embed arbitrary property name/value pairs into the certificate. Property based certificates are relatively new compared to the other two and can be used to express both the identity and capability certificates. The best example of identity certificate based systems is version 1 of X.509 (Public Key Infrastructure, Internet). Version 3 of X.509 which supports arbitrary attribute name / value pair is property certificate based but is primarily used as an identity certificate on the Internet. Capability certificate based systems like the IETF Simple Public Key Infrastructure [SPKI] and Keynote scheme in [BLAZ98] restrict the context in which a certificate can be used in authentication and authorization. The client's certificates in SPKI and Keynote systems contain embedded access permissions for services on the issuing server. Therefore the certificate is only valid on the issuing server.

2.5 Trust and Reputation Management in Mobile P2P Networks

Trust is one of the most crucial concepts driving decision making and establishing relationships. Trust is indispensible when considering interactions among individuals in artificial societies such as electronic commerce [YUB03]. As an important concept in network security, trust is interpreted as a set of relations among nodes participating in the network activities [RAMC04] [LIMC08]. Trusted relationships among nodes in a network are based on different sources of information such as direct interactions, witness information and previous behaviours of nodes.

Trust management in distributed and resource-constraint networks, such as DTNs and sensor networks, is much more difficult but more crucial than in traditional hierarchical architectures, such as the Internet and access point centred wireless LANs. Generally, this type of distributed network has neither pre-established infrastructure, nor centralized control servers or trusted third parties. The dynamically changing topology and intermittent connectivity of disconnected MANETs establish trust management more as a dynamic systems problem [BARA05]. Furthermore, resources (power, bandwidth, computation etc.) are limited because of the wireless and ad hoc environment, so the trust evaluation procedure should only rely on local information. In early stages of trust and security on MANETs several researchers relied on authentication, cryptographic encryption and decryption techniques. These schemes for security were shown to be effective; however these are based on centralized certification authorities. Significant communication overheads from both pre-processing and during processing periods, as well as energy consumption were major challenges thus rendering these approaches to be poor for DTNs. It has been shown recently that trust and reputation based techniques are more effective in de-centralized mobile networks [SRIV06] [MERW07] [BALA07] [PIYA08] [LUOA09] [SALE09].

As reputation and trust have recently received considerable attention in many diverse domains several definitions exist.

Mui et.al in [MUIL02], define trust as "*a subjective expectation a node has about another's future behaviour based on the history of their encounters*".

Also in reference [BALA07] trust is defined as "*a firm belief in the competence of an entity to act as expected such that the belief is not a fixed value associated with the entity, rather it is subject to the behaviour of the entity and applies only to the given context within a defined time".*

While trust definitions focus more on the history of user's encounters, reputation is based on the aggregated information from other individuals. For instance, Sabater and Sierra [SABA05] declared that "*reputation is the opinion or view of someone about something*".

Trust and reputation models have been developed to improve the success of interactions by minimizing uncertainty. Many of the models are based on Marsh's trust formalism [MARS94], in using trust to assess the likelihood that a user honours its promises. Trust and reputation models can be classified into centralized and decentralized models.

2.5.1 Centralized Trust and Reputation Models

Reputation mechanisms have been widely used in online electronic commerce systems e.g. eBay [EBAY], Amazon which typically manage the reputation of all its users in a centralized manner. The main building block of these models is information about a node's past behaviours. This information is used to deduce the trustworthiness of that node in terms of its competency and reliability. Online reputation mechanisms e.g. those on eBay [RESN02] and Amazon Auctions [AMAZ] are probably the most widely used such models. They are implemented as a centralized rating system so that their users can report about the behaviour of one another in past transactions via rating and leaving textual comments. In so doing, users in their communities can learn about the past behaviour of a given user to decide whether it is trustworthy.

Disconnected MANETs are essentially distributed in nature, therefore centralized trust and reputation models may not be suitable. Recently some decentralized models for trust management for distributed systems have been proposed, some of these are presented here.

2.5.2 Decentralized Trust and Reputation Models

As more and more computational systems of all kinds move toward large-scale, open and dynamic architectures, more and more trust models are designed such that each node can carry out trust evaluation itself without the need for a central trust authority.

Jurca and Falting introduce a reputation mechanism where nodes are incentivized to report truthfully about their interactions results [JURC03]. They define a set of broker nodes called R-nodes whose tasks are buying and aggregating reports from other nodes and selling back reputation information to them when they need it. All reports about a node are simply aggregated using the averaging method to produce the reputation value for that node. In order to incentivize nodes to share their reports truthfully, [JURC03] propose a payment scheme for reputation reports. This scheme guarantees that nodes who report incorrectly will gradually lose money (during the process of selling reports and buying reputation information), while honest nodes will not. Therefore, this mechanism makes it rational for a node to report its observations honestly and this is the main contribution of their work.

ReGreT [SABA01] is a completely de-centralized model of trust and reputation with three dimensions of information: individual, social and ontological. The social dimension includes information on the experiences of other members of the evaluator’s group, or neighbourhood, which is assumed to be a group of nodes with some common knowledge. Employing Regret, each node is able to evaluate the reputation of others by itself. In order to do so, each node rates its partner’s performance after every interaction and records its ratings in a local database. The relevant ratings will be queried from this database when trust evaluation is needed. The trust value derived from those ratings is termed direct trust and is calculated as the weighed means of all ratings. Each rating is weighed according to its recency. Intuitively, a more recent rating is deemed to be more current and is weighted more than those that are less recent. Besides direct trust and witness reputation, Regret also introduces the concepts of neighbourhood reputation and system reputation. The former is calculated from the reputation of the target’s neighbour nodes based on fuzzy rules.

Reference [YUB08] developed an approach for social reputation management, in which they represented a node's ratings regarding another node as a scalar and combined them with testimonies using combination schemes similar to certainty factors. In this system, nodes cooperate by giving, pursuing, and evaluating referrals (a recommendation to contact another node). Each node in the system maintains a list of acquaintances (other nodes that it knows) and their expertise. Thus, when looking for a certain piece of information, a node can send the query to a number of its acquaintances who will try to answer the query if possible or, if they cannot, they will send back referrals pointing to other nodes that they believe are likely to have the desired information (based on that node's expertise).

Reference [HANG08] proposed an adaptive probabilistic trust model that combines probability and certainty and offers a trust update mechanism to estimate the trustworthiness of referrers. Some other trust-based network models include Trust-Net [SCHI00] and Histos [ZACH00]. [PAPA03], present an encryption based technique for secure message transmission in networks. A Robust reputation based approach to trust management in MANETs is presented in [BUCH04]. Authors in [ZOUR05] and a later paper [ZOUR06] define trust metrics and evaluate performance of proposed reputation based techniques with an emphasis on secure data delivery rates. An adaptive trust management scheme is proposed in distributed applications for MANETs in [LIH07], and [YUNF07].

A popular decentralized TRM is FIRE presented in [HUYN04] and [HUYN06]. FIRE presents a modular approach that integrates up to four types of trust and reputation from different information sources, according to availability: interaction trust, role-based trust, witness reputation, and certified reputation. FIRE model classifies users in a network as *Agents*, a set of users participating in trust interaction; *Targets*, users whose trust and reputation is being sought in an interaction and *Evaluators*, users requesting trust information about a target. Each time agent *i* gives a rating, it will be stored in the agent's local rating database. Ratings in this database will be retrieved when needed for trust evaluation or for sharing with other agents. However, an agent does not need to store all ratings it makes. Old ratings become out of date due to changes in the environment and may not be stored in limited amount of memory. In FIRE, trust rating is calculated based on recommendations from direct interaction, witness interaction or rule based interactions.

The evaluator node uses its previous experiences in interacting with the target agent to determine its trustworthiness. This type of trust is most frequently used [WANG08] [SRIV06] and is called Direct Interaction Trust (DIT). Assuming that nodes are willing to share their direct experiences, the evaluator node can collect experiences of other nodes that interacted with the target node. Such information will be used to derive the trustworthiness of the target node based on the views of its witnesses. Hence this type of trust is called Witness Interaction Trust (WIT).

2.6 Analysis of Related Work

This section presents analysis of related work in comparison to the work presented in this thesis. Section 2.6.1 presents issues and analysis of framework design for MSN application. Section 2.6.2 discusses the issues in opportunistic routing protocols for P2P applications in DTN environment. In section 2.6.3, vulnerabilities of existing TRM and comparison of techniques presented in this thesis are discussed.

2.6.1 Framework design for P2P application in MANETs

MANETs and P2P paradigm are decentralized and distributed in nature and share many similarities. In mobile P2P applications, users interact by means of handheld mobile devices while on the move. Point-to-point connections are made typically using Bluetooth or Wi-Fi networks and data is transmitted over these channels. Some efforts in deployment of P2P applications over MANETs have been made in [KLEM03], ORION presented in [OLIV03] and a Gnutella style application in [CONT05]. These P2P applications are implemented as overlays over MANET and employ broadcasts for data transmission over single hop using reactive MANET protocols such as AODV and DSDV. A drawback of using these protocols is that they compute the destination path for routing which may not be guaranteed in a delay tolerant environment.

Apart from communication issues, security in data transmission in mobile P2P applications is a challenging issue. Recent advances in semi de-centralized P2P application proposed in [SERE07] and [MERW07] rely heavily on encryption protocols in client to server communication but provide no security in P2P interactions. Without the existence of central authority in P2P applications secure transmission is difficult. Furthermore, methods using encryption require heavy computation whereas mobile devices have limited computation resources. A popular approach to providing security in P2P applications is reliance on trust and reputation management models and techniques. Most trust and reputation models require peers in a network to generate trust ratings based on interactions with other peers. Trust ratings from peers are used to compute and update the local and global trust ratings of peers in the network. Trust and reputation models such as FIRE [HUYN06], GossipTrust [ZHOK07], Power-trust [ZHOU07] and H-trust [ZHOU08] all rely on trust recommendations from peers to compute trust ratings.

Chapter 3 presents a framework for a file sharing P2P application considering content driven data transmission in delay tolerant environment. The proposed framework utilizes opportunistic store-carry-forward approach to data transmission based on eMule [EMULE]. Bluetooth connections are used for point-to-point data transmission. A drawback of the proposed framework is, unlike ORION, it is single-hop and does not provide data delivery over multi-hop. Furthermore, it implements a light weight trust and reputation model based on the popular weighted average model [HUYN06]. Only the direct interaction trust ratings are considered when computing trust. Chapter 6 extends the trust management model presented in chapter 3. A trust based framework for a P2P mobile social networking application is presented using Dynamicity Aware Graph Relabeling System (DA-GRS) [CAST06]. The proposed framework is tested in various simulation environments.

2.6.2 Routing issues in DTNs

A number of protocols have been designed in the last few years in order to support destination-driven routing in MANETs. DTN, being a relatively new type of network, has been receiving enormous interest from researchers in recent years. Due to the frequent disconnections and topology changes, nodes in the DTN can scatter and form clusters, therefore efficient routing mechanisms for MANETs are not applicable. The earliest approach to routing in DTN was in epidemic routing presented in [VAHD00], where data is continuously replicated until all nodes receive a copy; this approach causes flooding and therefore is not efficient, in particular to DTNs.

The A/G algorithm, presented in [DATT04], utilizes the epidemic algorithm to spread data items selectively based on vulnerability of other nodes (multicasting), instead of treating all nodes homogeneously (broadcast) and flooding the network. Another benefit of using multicasting instead of broadcasting messages is the improvement of efficiency in transmission. Researchers in [WIES00], [BANE03] and [LIUB08] have addressed the issue of energy efficiency in transmission of broadcast and multicast protocols for mobile wireless networks. Results and discussion presented in these works show that multicasting is more effective in reducing the cost of transmission over a period of time. The Opportunistic Routing Protocol (ORP) presented in chapter 5 implements multicasting approach to data transmission in a DTN environment.

PRoPHET [LIND03] protocol implements the store-carry-forward approach for packet delivery in DTN. More recently the CAR protocol presented in [MUSO09], utilize a similar methodology, although it uses a statistical approach to calculation of delivery probabilities. The CDDPP protocol presented in chapter 4 facilitates implementation of a P2P Mobile Social Networking application. Its takes the opportunistic content driven approach to data propagation; i.e. data packets are forwarded to nodes with similar content. It also utilizes the opportunistic store-carry-forward approach to routing data packets. The CDDPP protocol is extended into ORP including the multicasting of data packets and transmission over multiple hops. Work presented in chapter 5 compares the performance of ORP with A/G algorithm [DATT04].

2.6.3 Vulnerabilities in TRM for Mobile P2P Networks

Decentralized TRMs presented in section 2.5.2, might use different sources of information such as direct experiences, witness information, sociological information and prejudice [LIMC08]. Researchers in [LIJ08] [LIAN07], have identified the existence of cheaters (exploitation) in artificial societies employing trust and reputation models and the existence of inaccurate witnesses. This inaccurate information can challenge the integrity of the reputation system based on witness information leading to misleading trust information. A new type of attack presented in [SALE09], is referred to as con-man attack. In this type of attack an attacker sits between the sender and the receiver and sniffs any information being sent between two ends. In some cases the attacker may impersonate the sender to communicate with the receiver, or impersonate the receiver to reply to the sender. Con-man attack is similar to another attack of its type referred to as the collusion attack. Collusion attacks occur when one or more nodes conspire together to take advantage of breaches in trust models to defraud one or more nodes [QURE10]. It can be the case that nodes in the colluding group adopt a sacrificial stance in collusion attacks in order to maximize the utility of the colluding group. Collusion attacks often work based on the basic idea that one or more nodes show themselves as trustworthy nodes in one type of interaction (usually direct interaction). Afterward, they will be untrustworthy in other type of interaction (e.g., witness interaction) by providing false information in favour of other members of the colluding group. This false information usually encourages a victim to interact with members of the colluding group and rely on false information provided to compute trust information.

The reputation management system presented in [YUB08] is based on acquiring trust ratings from social contacts. Similarly the works presented in [ZOUR06] utilizes the effectiveness of the proposed scheme to maximize secure data delivery rate. More recently [LIH07] and [YUNF07] proposed adaptive trust and reputation system for an application in MANET. FIRE [HUYN06] trust and reputation model is a well known and vastly utilized trust and reputation model. All of these approaches rely on trust ratings inquiry from neighbouring nodes in the network, in direct or witness interactions. Regardless of the effectiveness of these techniques, they are susceptible to collusion and the con-man attacks. Work presented in chapter 7 is inspired by evaluating the impact of collusion attack on FIRE. The proposed FIRE+ [QURE10] trust and reputation model, defines a mechanism for keeping a history of trust ratings and measure of confidence in ratings received from direct and witness interactions. The trust network graph determines the reliable ratings provided by direct and witness agents utilizing the experience of interactions while synthesizing unreliable ratings from colluding / malicious agents with dubious recommendations. The determination of the value of confidence in trust values is crucial to the success of FIRE+. In this context, new policies were defined to determine collusive behaviour and show experimentally that FIRE+ nodes using a multidimensional trust and reputation model while utilizing the trust policies can counter the risk of a direct interaction and witness interaction collusion attack by malicious agents in FIRE.

Most of the trust and reputation models presented in section 2.5.2 utilize a full aggregation trust ratings mechanism. Usually a full aggregation reputation system is of high accuracy; however, the aggregation approach involves a trade-off between the accuracy and overload. The overload of the full aggregation is quite heavy when the network expands very large. In addition, the reputation convergence is not fast. In mobile P2P networks, peers join or leave the network frequently, which leads to the dynamic network topology changes. Due to frequent changes, a trust management system needs to repeatedly revise and update trust ratings, which in turn can increase the communication overhead. Moreover, pervasive devices that are resource-constrained need to avoid unnecessary trust ratings computations and storing redundant or obsolete trust ratings. Furthermore, accuracy of direct and witness trust rating from reliable and trustworthy peers, is necessary for the reliability and robustness of the trust ratings aggregation scheme. In order to attain a highly accurate, robust and efficient trust and reputation management system a trade-off between the computational complexity and accuracy is vital. To the author's knowledge, no trust and reputation model exists that has been specifically designed for P2P mobile networks, considering the limitations of computations power, limited storage and wireless communication issues. In designing M-trust, five key characteristics to address the essential trade-off in ratings aggregation.

- Reliability; in detecting malicious activity from a peer and categorizing it as a malicious peer.
- Accuracy; in computing trust ratings for local interactions and maintaining global trust ratings.
- Adaptability; in considering frequent topology changes due to mobility.
- Robustness; in avoiding trust ratings from untrustworthy and unreliable sources.
- Light-weighted-ness; in avoiding heavy computation and frequent communications with peers for updates. Furthermore, reducing the size of trust list by removing redundant and obsolete ratings.

To this end, work described in Chapter 8 presents trust ratings aggregation mechanism referred to as M-Trust. M-trust relies on confidence in reputation for computing direct trust ratings and witness recommendations from reliable peers to determine trust ratings for a peer using the proposed trust ratings aggregation algorithms. Simulation results demonstrate that the overall performance of M-trust is accurate, reliable and robust for detecting malicious peers in P2P mobile networks. Four trust management techniques, Bellman-Ford [ZHAO09], Received Ratings [LIMC08], Weighted Average [HUYN06] and Ultimate Trust [BAHT10], were compared with M-trust to analyze the performance of the proposed scheme. Simulation results show that M-trust is comparable or better than the rest of the presented techniques in the five performance categories mentioned above.

2.7 Summary

This chapter presented a review of important concepts related to this thesis. Section 2.1 described mobile networks with emphasis on properties and applications of MANETs and DTNs. Routing protocols for MANETs and DTNs are presented in section 2.2. Section 2.3 presented features of mobile P2P networks, challenges in deployment of P2P overlays on MANETs and the existing P2P overlays for MANETs. Popular mobile P2P applications such as file sharing and MSNs are also discussed. Security issues and challenges in MANETs are discussed in section 2.4 along with types of attacks on MANETs and proposed solutions to known attacks using cryptography techniques. Section 2.5 presented trust and reputation management models. Features for the popular centralized and de-centralized models are presented. Section 2.6 presents a discussion on analysis of related work.

Chapter 3

Opportunistic Trust based P2P Framework in disconnected MANETs

3.1 Introduction

Due to advances in micro-electronic wireless technologies, mobile devices with better and better processing, storage and communications capabilities are being made available. Devices such as multi-function mobile phones, personal digital assistants, wearable devices and handheld sensor devices are considered as pervasive devices. These pervasive devices when used in urban computing scenarios bring a lot of unknown people together allowing discovery of other people and possible communication and sharing of information. Personal handheld devices carried by people can communicate with embedded servers to obtain relevant information thus forming an open and a dynamic network. The networks formed in these open and dynamic environments are delay tolerant ad-hoc P2P networks [HUIP08] [ALMA08]. These networks are categorized by not having a pre-deployed fixed infrastructure nor centrally administered space controlled users. Rather, these pervasive devices are resource constrained, self-organized and dynamically self-configured to set up in the network by both consuming and providing services as peers.

In a disconnected MANET, information may be carried by a mobile node and forwarded opportunistically across partitions, therefore allowing communication between areas of the network that are never connected by an end-to-end path. Recently, this kind of opportunistic forwarding scenarios became popular in the research area investigating DTNs. Mobile nodes enable indirect data exchange among disconnected portions of the overall network, typically using a store-and-forward approach and some form of opportunistic forwarding [ZHAN07] [DALY07] [CHAI08]. To assume trustworthy interaction in this kind of networks is unrealistic due to the fact that most entities in the network are unknown. Consequently, trust has recovered a big interest as a basis to secure and manage P2P relationships [PIET09].

Trust can be used to establish new connections between unknown entities, or to measure certain parameters such as cooperation ability, QoS, individual behaviour and social environment. Recent studies have already demonstrated the feasibility of using distributed trust techniques in self-organized, distributed and resource-constrained networks. TRAVOS [TEAC05] is a trust model that is built upon probability theory and based on observations of past interaction between nodes. Yu and Singh developed an approach for social reputation management, in which they represented a node's ratings as a scalar and combined them with testimonies using combination schemes [YUB00]. [HANG08] proposed an adaptive probabilistic trust model that combines probability and certainty and offers a trust update mechanism to estimate the trustworthiness of users in a de-centralized distributed network. Authors in [PATW05] proposed a reputation-based decentralized trust management middleware. The reputation information of every peer is stored in its neighbours and piggy-backed on its replies to requests for data or services. eBiquity Group proposes a trust based data management framework, in order to enable mobile devices access to the available distributed computation, storage, and sensory resources [REPA06]. This also includes a reputation system from the history of prior encounters.

In this chapter a light weight trust based framework for secure digital content sharing in pervasive devices is proposed. The main contribution of this work is to allow providing non-existing security services to the applications in a dynamic way by making pervasive devices act as secure peers. The proposed framework allows peers to store, carry and forward shared content in an opportunistic manner. A file exchange protocol for opportunistic host discovery and file sharing in ad hoc environment is also proposed. The framework is implemented using J2ME Personal Profile and tested in PDA devices. User trials conducted, test the performance of the framework in presence and absence of the trust module. Finally, conclusions based on results discuss the strengths and shortcomings of the proposed framework followed by recommendations for further improvement.

3.2 Framework Design

The main motivation of the proposed trust-based framework is to provide flexible security services to the P2P applications in a disconnected MANET. The proposed architecture is based on the concept of distributed decentralized trust models that eliminates the complexity of establishing new relationships, the dependence on a central server, the need for frequent manual setting and always-on global connectivity. Any device that can participate in a P2P communication model can establish connections in a secure way using an opportunistic communication protocol. Figure 3.1 shows an overview of the proposed framework in use by mobile and stationary users.

Users mainly operate in mobile ad hoc mode, as the devices come within each other's range, files are shared among users. It is assumed that a mobile user may physically change location to an area with Internet accessibility providing an opportunity to synchronize data or upload / download latest versions of files. A user can also move to a new location and establish connection with a cluster of mobile ad hoc users while sharing the latest version of the downloaded files thus utilizing the notion of exchanging files with an opportunistic store-carry and forward mechanism. The neighbourhood discovery method depends on the radio technology being used: commonly available options with today's mobile device hardware include Bluetooth device discovery or broadcast beacons on a well known Wi-Fi SSID. Figure 3.1, illustrates the ad hoc mode as the circle in the centre where two devices move in the vicinity of each other and engage in interaction. Of course the neighbourhood can, and usually will, contain more than two devices; the system must therefore manage multiple simultaneous connections. In a delay tolerant MANET, nodes discover each other as long as they can communicate in a limited range depending on the device capability and radio technology used. Nodes can frequently appear or disappear depending on various environmental factors or device limitations.

The application running on connected devices transfer / update the profile and exchange files. These files are stored on the devices within the limits of storage space and forwarded to other devices as contact opportunities arise. These opportunistic exchanges combined with human mobility create a temporal communications network as in Pocket Switched Network (PSN) [SUJ07] where messages travel from device to device over multiple hops without any infrastructure connectivity. Figure 3.2 depicts the architecture of the proposed framework. The main advantage of using this design is that, application developers can rely on the framework for security, trusted user discovery and interaction and file sharing. The three layers of the framework namely; application layer, communication layer and trust management layer are discussed in detail.

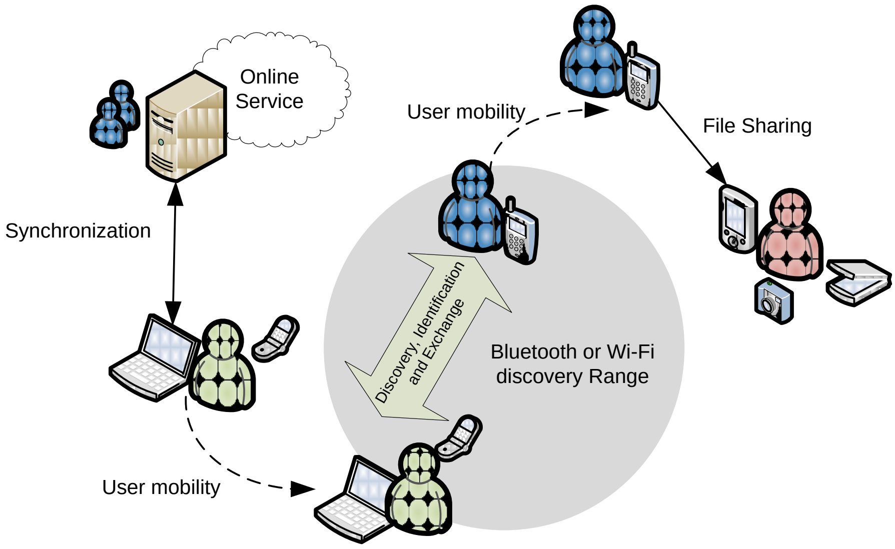

Online Service
User mobility
File Sharing
Synchronization
Discovery, Identification and Exchange
Bluetooth or Wi-Fi discovery Range
User mobility

Figure 3-1: Proposed System Overview

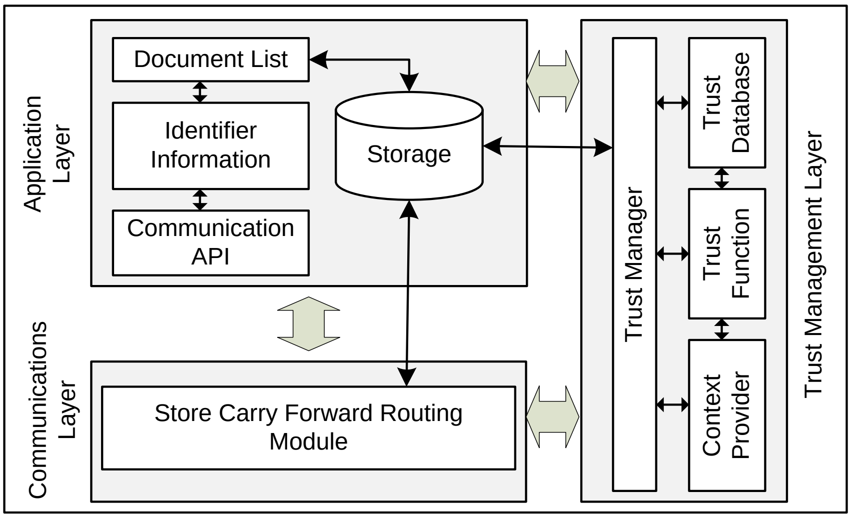


Figure 3-2: Framework Architecture

3.2.1 Application Layer

Recent implementations of mobile social networks from popular social network sites such as facebook and myspace rely on Email and short messaging service on the client's device. To search for a friend in the social network a user needs to subscribe to the service and query the database for users with common interest. A P2P implementation of this service would be effective in congestion control and would provide additional functionality of mobility to the users where the users would be able to communicate while on the go. This would provide users to directly communicate instead of subscribing to the service provider or paying for short text messages and hence may be able to share rich media content. A very effective network topology would be to use delay tolerant MANETs, where the nodes have the freedom of mobility. A node may receive data intended for a target node, store and forward it when an opportunity arises, thus forming disconnected clusters of participating nodes.

In a social network, users subscribe to the service by making a public profile. A profile is designed to introduce a person to other members of the network announcing personal information, interests, location and a list of documents to share. If a user makes a search, his personal interests are matched in a database and query results are returned. The user may choose to select from a number of interested users and send an “invite”. The invited user receives the invitation message, if interested the user responds and the two users become friends. Friends can show their documents publicly and may even share them. A user announces his documents to a friend, if the friend is interested he can request a document. Researchers in [EAGL06], [LUGA07] and [RAEN05] discuss implementation of various forms of social networks. Typically three factors are essential to successful data sharing in a social network, Interest Profiles, Document Lists and Document Repository.

A light weight and simple P2P application to exchange messages and files between pervasive devices has been developed. A communication API used to provide interaction between the application and the framework is shown in Figure 3.3. This application shows the user the current set of neighbouring devices with related information such as user profile. A user may search for new neighbours; remove users from this list of neighbours and send messages. User can tag a neighbouring user as a trusted friend. The user can also enable distinct alerts to be notified when a friend is in the neighbourhood. When two neighbours communicate, they can share a list of files stored in the user’s device. If willing, users can share these files in an opportunistic manner, i.e. a user can store, carry and forward files and share with other users.

Identifier Information: Each user maintains a unique identifier. This identifier is used to search for neighbours along with maintaining a list of neighbours. Furthermore this identifier is also used to compute trust values based on recommendations from neighbouring nodes.

Interest Profiles: Each user maintains a list of keywords describing his interests. These keywords are used for searching and indexing purposes. An interest profile can be detailed and may even contain both text, as well as graphics data and therefore it can take increasing amount of storage allocation. However for the proposed framework it is assumed that an interest profile would be a collection of keywords only and therefore would take minimal amount of storage.

Document List: is a list of documents stored at a host. A document list consists of certain attributes of documents stored in the device storage area. These attributes include but are not limited to a Unique Identifier for the document, Document size, Document type, ownership and a Timestamp. Each document stored in the device's storage has this information. A Unique identifier uniquely identifies a document, where each document name is the standard file name format i.e. (filename.extension). Document size is mentioned in bytes. Document type relates to the particular interest and contains the description for that interest. Ownership is the unique user identifier. A Timestamp is the date and time for the document creation and indicates when the document was last updated. A list of documents is announced whenever two users with similar interests decide to share files.

Storage area: Each node maintains a document repository for documents to be shared. Since most mobile devices have limited storage for documents, a limit is set to the size of a device's storage area.

```
Reg_application(app_no)
Set_identifier(OSCF_UID)
[get|set]document_list(OSCF_dir)
[get|set]user_list(OSCF_UID)
message(dest, ttl_timestamp, message, OSCF_dir)
```

Figure 3-3: Communication API

3.2.2 Communications Layer

Communication layer is the second layer of the framework. This layer is responsible for discovery, user identification and providing document exchange. It is composed of a store-carry-forward module which is a communication protocol. This module is a modified and improved version of the eMule [EMULE]. Details of these modules are as follows.

Store Carry Forward Module: The framework implements a simple three step process for all transmissions. It is assumed in ad hoc mode, each node executes a periodic loop that consists of three steps: (1) neighbourhood discovery, (2) user identification (and authentication), and (3) data exchange. Upon discovery of a new device in the neighbourhood, the system enters the identification phase where the devices open a communication link between each other to exchange the user identity information. Upon a first encounter the devices running the application, exchange their profiles during the identification phase. The system stores the profiles persistently along with other contact statistics to avoid unnecessary profile updates and to make subsequent decisions, e.g., to forward messages between nodes. During subsequent contacts the profiles are exchanged only if the profile has changed since the last encounter (i.e., user changed his nickname or status etc.), otherwise the nodes only exchange their user identifiers. Once the identification is successfully completed, the last step of the interaction is the data exchange phase.

It is possible that many adjacent nodes would request the same document from a host. In this case a unicast message needs to be sent to all requesting nodes. This however would greatly decrease the performance due to overhead of repeatedly sending the same message. As a solution to this problem the *n-list* is used. The *n-list* is a list of adjacent neighbouring nodes that have been contacted in the past interactions. If a simple majority of hosts request the same document, a broadcast message is sent to all immediate neighbours, instead of individual unicast messages, thus flooding the corresponding group of users. This is to ensure that all members of the group would keep forwarding the content until everyone has received a copy of the document and no copy of the document is sent to a user who is not a member of the group. This provides a minimum of guarantee on privacy and also helps as an incentive mechanism.

3.2.3 Trust Management Layer

In the absence of a centralized server for trust management and security credentials verification, providing trusted interaction among users is a challenging task. De-centralized reputation based models depend on periodically updating trust values of a node based on local knowledge gained from neighbouring nodes. A light weight trust manager, based on Pervasive Trust Management (PTM) model [ALMA06] is included. PTM allows establishing trust relationships in an ad hoc manner between nodes. Each node has its own public/private key pair, a protected list of trustworthy and untrustworthy users, and behavioural information.

Trust Management Layer is responsible for providing and storing trust ratings for neighbouring nodes. Each node in a cluster of connected nodes periodically asks for trust ratings from its neighbours and updates trust values defined in a list. The Trust layer consists of four components, trust manager, trust function calculations, trust values list and context provider.

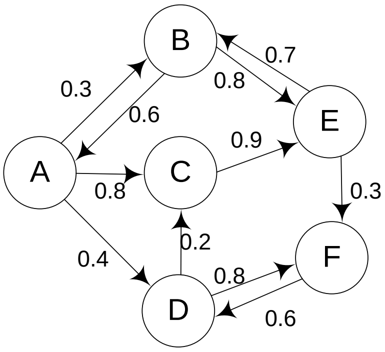
Trust Manager: A trust relationship between two nodes is established based on direct interaction trust values. It is possible to have two scenarios for direct trust interactions, (1) trust establishment determined by contextual information, (2) trust establishment determined by recommendations from direct neighbours. In the first scenario two nodes having no history of encounters can trust each other based on contextual information. Contextual information is gathered often as a consequence of a complex set of beliefs, perceptions and interpretations based on periodic monitoring of the behaviour of nodes in direct interactions. The context provider component of the trust management layer controls this information. In the second scenario, a node *i* requiring trust value for another node *j*, will request recommendation from its neighbours. Recommendation replies are sent if there already exists a trust relationship between some neighbouring nodes. Such replies are only accepted if they come from trusted peers. Recommendations are considered from a trusted peer if it has a trust value larger than a thresholdσ, for instance, $\sigma > 0.5$. At the moment, only recommendations from directly trusted nodes are acknowledged. Long recommendation chains are avoided to minimize security and scalability problems.

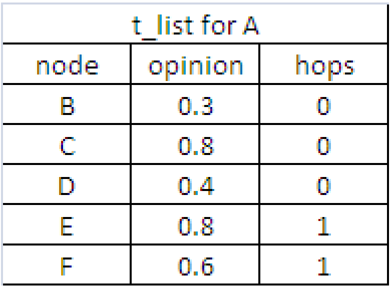
The unauthorized access to a resource is avoided via the Access Control. The Trust Manager enables to distinguish among different authenticated users. It checks whether the user is trusted or not and subsequently requests the Access Control module to grant access.

Trust Function: Trust variable $T_{i,j}(t)$ is defined to identify the level of trust a node *i,* has for a target node *j* after *t* interactions between agent *i* and agent *j*, while $T_{i,j}(t) \in [-1, +1]$ and $T_{i,j}(0) = 0$. One agent in the view of the other agent can have one of the following levels of trustworthiness: Trustworthy, Not Yet Known, or Untrustworthy. The trust value is calculated as the sum of all the available ratings weighted by the rating relevance and normalized to the range of $[-1, +1]$.Trust value for an agent is given by the function

$$T(x) = \frac{\sum_{i \in t\_list} \left(Trust\ (i) * opinion\ _i(x)\right)}{\sum_{i \in t\_list} Trust\ (i)} \tag{3.1}$$

Whereas *x* is the node whose trust is to be computed; *i* is a node in the list of trusted users (*t_list*) and the function *opinion$_i$(x)* indicates the opinion of user *i* towards user *x*. The value of *opinion$_i$(x)* is determined by the context provider component. Value for *T(x)* is always in the interval (1, -1), i.e. a Trustworthy user will obtain a positive value, whereas a negative value indicates an untrustworthy node.





t_list for A		
node	opinion	hops
B	0.3	0
C	0.8	0
D	0.4	0
E	0.8	1
F	0.6	1

Figure 3-4: (a) A network friendship graph with trust ratings (b) *t_list* for node A

To further explain the usage of (3.1) consider figure 3.4(a) that presents a network friendship graph with trust ratings. Assuming A is connected to nodes B, C and D, and A seeks trust ratings for E. Since there is no direct contact with E initially, A requests trust ratings for node E from immediate neighbours B, C and D. In the figure 3.4, the values on the edge between two vertices in the graph indicate opinion. B, C and D returns values 0.8, 0.9 and 0.5 as trust ratings for node E. Using the equation (3.1) node A computes the trust ratings for E using node A's opinions for nodes B (0.3), C (0.8) and D (0.4) as follows:

$$\mathrm{T_{AE}} = \frac{(0.8*0.3)+(0.9*0.8)+(0.5*0.4)}{0.3+0.8+0.4} = 0.773$$

The newly computed trust value T$_{\mathrm{AE}}$ is stored as node A's trust rating for node E in the *t_list* for node A. Figure 3.4 (b) shows *t_list* for node A in the friendship graph.

Trust Values List: The trust values for all users in contact are stored in the *t_list* and are updated frequently. If a trust rating is requested for a particular user, the latest value stored in *t_list* is forwarded to the requesting user. If a node *i* has requested trust value for node *j* from neighbouring nodes, trust values received are stored in the *t_list*, only if a trust value for the recommending node exists in the *t_list* provided the $T_{i,j}$ is greater than 0.5.

Context provider: The context provider updates the *t_list* according to trust ratings received from neighbouring nodes. Since each record in the history has a timestamp *ttl* value for each trust recommendation, older values can be discarded to reduce the size of the *t_list*. This interaction history gives a reflection of the relevant past transactions of a node. To determine if a service performed in an interaction was to the desired expectation, the desired value of service is compared to the actual value at the completion. The values of two variables, α (number of positive interactions) and β (number of negative interactions) accordingly. Based on the values of α and β the $opinion_i(x)$ function provides the context information as a positive or negative value thus affecting the trust value for a target node. As an example, if a user was able to successfully complete the transfer of a desired file, the value of α would be incremented. Alternatively if a user received an unexpected file instead of a desired file, value of β would be incremented since the desired service was not to the expectations.

As mentioned earlier, the value of $T_{i,j}$ determines if a node *i* is trustworthy, untrustworthy or not yet known. A node whose trust value $T_{i,j}$ falls below -0.2 ($T_{i,j}(t) < 0.2$) due to poor opinions, is considered untrustworthy. Untrustworthy nodes are removed from the *t_list* and their membership is effectively revoked. It is however possible that an untrustworthy node gains enough confidence in later transactions with other nodes to improve its trust value in neighbours and thus can be forgiven.

3.3 Framework implementation details

The prototype of the proposed framework is implemented using the J2ME Personal Profile. The size of the prototype binary is 280KB which is an acceptable size for devices with limited capabilities. In this section, a brief overview of the class structure in the framework is given. Section 3.3.2 details the protocol used in the framework followed by design for P2P MSN application in section 3.3.3.

3.3.1 Framework Classes structure

Figure 3.5 shows the Classes diagram for the implementation. The set of classes can be classified into five categories, OSCF_protocol, OSCF_Profile, OSCF_message, OSCF_files, and Trust_Manager. What follows is the summary of these five categories of classes.

- OSCF_Protocol: This class provides implementation of the Opportunistic Store Carry Forward (OSCF) module presented in the framework design in section 3.2. This class provides connection between the other categories of classes.
- OSCF_Profile: This category of classes is composed of OSCF_sniffer, OSCF_announce, OSCF_request and OSCF_invite classes.
    - The OSCF_sniffer class provides the listening module that enables the application to discover neighbouring devices using the Bluetooth discovery protocol. If a neighbouring device is discovered class OSCF_announce is called. For other types of messages received, appropriate classes are called from OSCF_Protocol.
    - OSCF_announce class implements the announce function of the framework. User profile with user identification information and interests are announced.
    - OSCF_request class generates a connection request using the similarity of interest profiles information.
    - OSCF_Invite class is invoked if a target user agrees to initiate file sharing provided that the interest profiles for both users match.
- OSCF_message: This category of classes is composed of OSCF_UID, OSCF_receive_msg, OSCF_send_msg and OSCF_user_list. The purpose of this set of classes is to manage message passing between users.
    - OSCF_UID maintains the unique identification information. User id is stored in device storage along with the MAC address of the device being used for communication.
    - OSCF_receive_msg and OSCF_send_msg are used to send and receive messages.
    - OSCF_user_list simply maintains a list of users with their interest profiles.
- OSCF_files: This category of classes is composed of OSCF_storage, OSCF_dir, OSCF_dir_file, OSCF_dir_file_perm, OSCF_dir_file_loc, OSCF_send_file, and OSCF_receive_file.
    - OSCF_storage implements storage for trust ratings, neighbour list *n_list* and interest profiles. Besides OSCF_Protocol class, this class is also used by all modules of the framework.

- OSCF_dir, OSCF_dir_file, OSCF_dir_file_perm and OSCF_dir_file_loc classes maintains list of files with their attributes, permissions and location in the actual directory on the devices storage. The attributes of files are defined in the previous section.
- OSCF_send_file and OSCF_receive_file classes are similar to OSCF_send_msg and OSCF_receive_msg, the major difference is using files instead of messages. In the current implementation of the framework the whole file is sent or received over the Bluetooth RFCOMM communication channel.

- Trust_Manager class implements the trust manager. Trust_Manager is responsible for acquiring, computing and providing trust information. Classes utilized by Trust_Manager are:
  - Opinions_list class maintains list of neighbours with their trust ratings.
  - Context_provider class context information from interactions with other peers.
  - Trust_function class implements the trust function defined in equation 3.1.

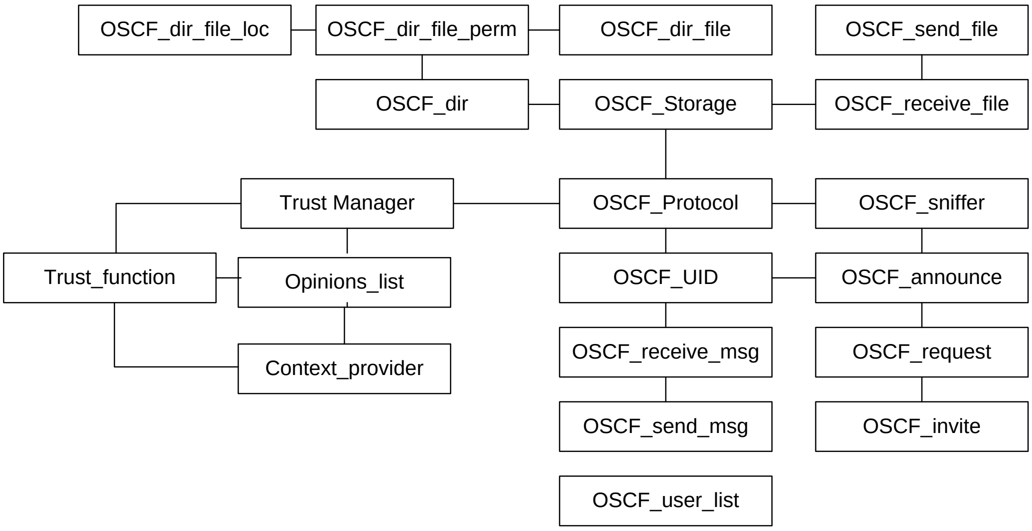


Figure 3-5: Prototype Classes Diagram

3.3.2 OSCF protocol

The framework implements a simple Opportunistic Store carry forward protocol (OSCF). Packets sent using OSCF Protocol follows the header shown in figure 3.6 with five fields and 18 bytes of header size. A device in the network is identified using MAC address as source or destination and is thereby 6 bytes. *ttl* is the time to live for the packet. Payload length is 4 bytes that contains the number of bytes that follow the header. Message type takes one byte and contains one of the following types of packets; announce, invite, doc_list req, doc_list rep, t_req, t_rep, and data.

- announce: used by the OSCF protocol. Every device emits a announce packet periodically. The announce packet contains four interest profiles. The interest profiles are used for discovery and identification of peers with similar interests. At the moment, the interest profile consist of two byte keywords (e.g. A0, B3, C3 etc)
- invite: If a peer receives a announce request and is willing to share with the requesting peer, it replies with a invite packet. The replies received from neighbouring nodes help populate the list of neighbours *n_list*. A session is established when two peers with similar interest profiles proceed with further transmission.
- doc_list req: peers in a session based on mutual interest profiles, request for document list.
- doc_list rep: a packet with doc_list rep contains a list of documents. A document list is sent as a list containing information about files stored in local host's storage. The following details about files are included in a reply. Filename.extension, file size, file type, owner and timestamp.
- t_req: Request trust ratings for a peer. The MAC id for the target peer is included in the data field of the packet.
- t_rep: A reply to trust ratings requested. The replying peer sends the trust ratings value for the selected peer appended to the data field in the packet.
- data: indicates that the packet contains data.

Source
Destination
Message type
time to live (ttl)
Payload

Figure 3-6: OSCF_Protocol header used for communication in the framework

In the current implementation data transmission is handled by RF_COMM in the Bluetooth communication protocol. The OSCF packet is compiled including the header and data items as an object and is sent to other peers. The device receiving the packet should also have Bluetooth communications enabled and should have installed the framework application to process the received request. Files received from peers are stored in the local cache (storage) of the device. A user can choose to store a file destined for another user, if an opportunity arises, the file is forwarded to the destination device.

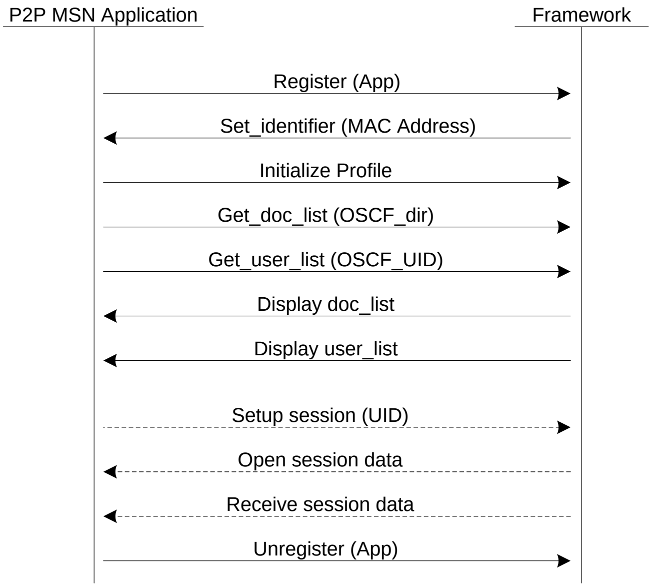


Figure 3-7: Framework API interaction workflow with P2P MSN Application

Figure 3.7 shows the usage of the framework API interacting with the P2P MSN files sharing application. Initially the MSN application registers with the framework and the device MAC address is registered as valid user id. The application requests the user to initialize the interest profile including the interest keywords and the documents to be selected for sharing. When user selects and approves the keywords and documents, the selections are displayed and the values are set. When this process is completed, the framework initializes peer search, if a peer is found with similar interest profile, a session is created from which data can be transferred to the other peer's device. Application can unregister from the framework when all tasks are completed. The details for the P2P MSN file sharing application are given in the next section.

3.3.3 P2P MSN File sharing application

The P2P MSN file sharing application is part of the framework and provides user interface to the peers using the Framework API given in Figure 3.3. The main usage of this application is to allow peers to chat and share files. Users can write messages and send them to others much like the functionality of chat rooms. Users can also select files stored in local cache and send these to other peers who in turn can also send files in a P2P manner. Figure 3.8 shows a workflow diagram for this application.

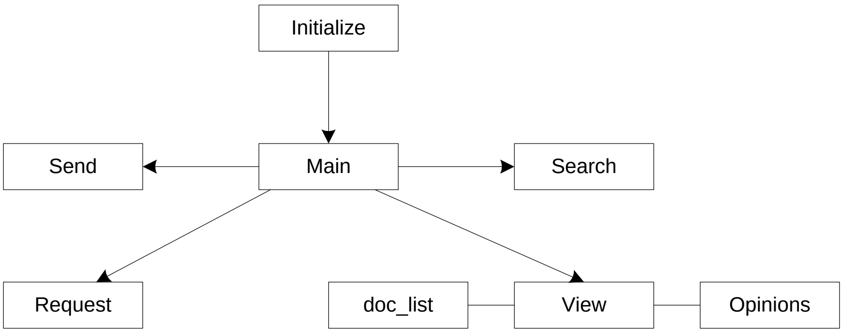
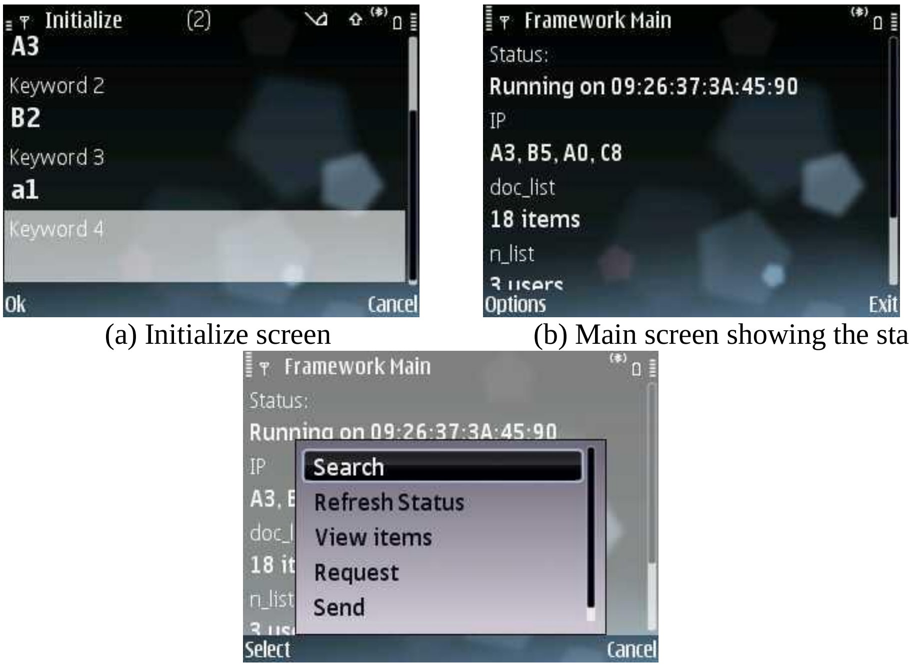


Figure 3-8: P2P MSN Application workflow diagram

At the outset the application needs to initialize by acquiring the MAC address of the device and asking user to provide four keywords for the interest profile. When this information is acquired, the application displays the main screen[1]. Figure 3.9(a) shows the initial screen; Figure 3.9(b) shows the main screen. Being the focal point of this application this screen also shows the status of the current application, the users connected recently and message alerts. The status about recent activities is updated whenever a new activity commences. The main screen allows user to search, view, send or request information as can be seen from Figure 3.9(c).



Initialize (2)

A3

Keyword 2

B2

Keyword 3

a1

Keyword 4

Framework Main

Status:

Running on 09:26:37:3A:45:90

IP

A3, B5, A0, C8

doc_list

18 items

n_list

3 users

(a) Initialize screen

(b) Main screen showing the status

(c) Main screen with menu options

Figure 3-9: Initialize and Main screens

[1] The screens were captured using Bestscreensnapper http://www.softpedia.com/ on Nokia E71 device

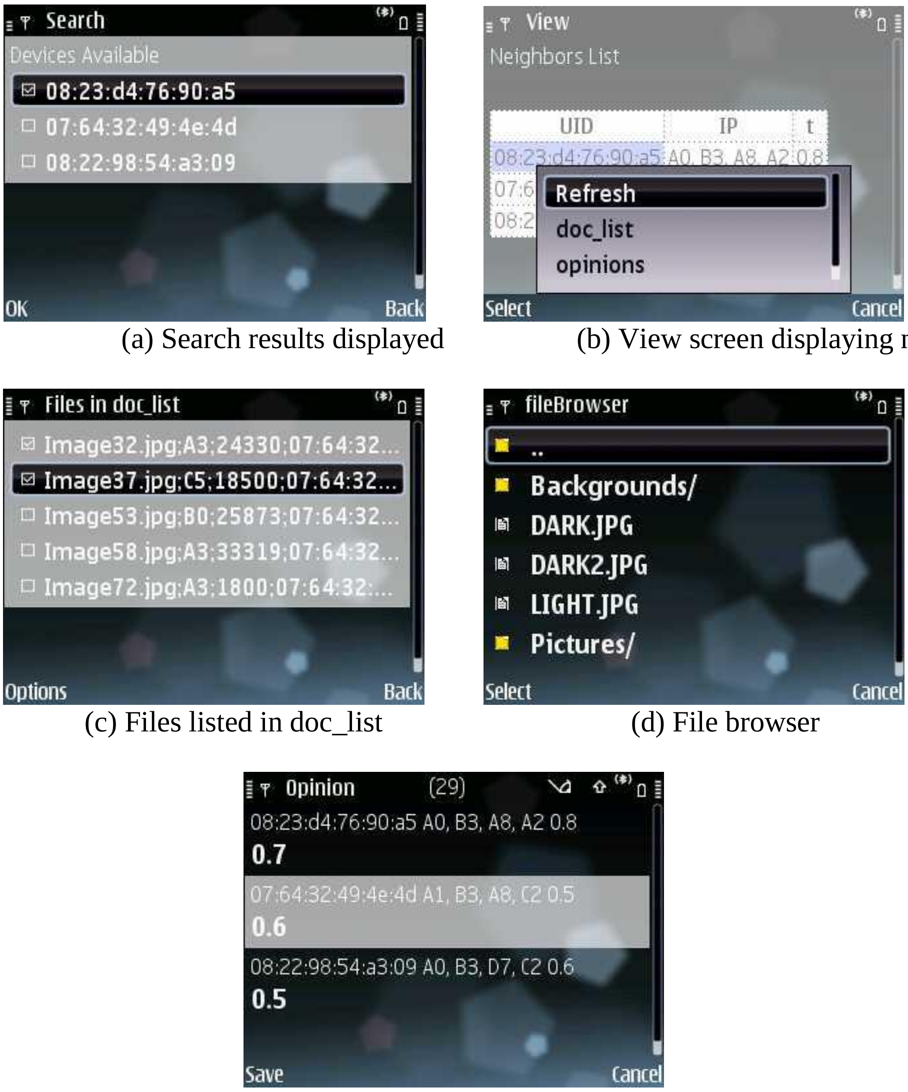
Search: One of the initial processes is to search for other peers. When a search call is made the application interacts with the framework to discover devices using the same application. The invitation is broadcast to the neighbouring peers, if willing; a receiving peer will send its interest profile and *doc-list* to the initiating peer, which will store the received information locally. The status on the main screen is updated and a list showing the result of the search is displayed as can be seen from figure 3.10(a).

View: The view screen allows users to view neighbours list, *doc_list* stored locally and opinions of other peers. The neighbour list displays the MAC address of the device, four associated interest profile keywords and the current trust ratings for the particular device. Users can refresh the screen to obtain recent trust rating values. The *doc_list* for the local device is also accessible from the view screen. Figure 3.10 (c) shows a list of files available in the *doc_list*. Each file listed is displayed as <filename.extension; file type; file size; MAC address for the owner; data and time>. The files can be selected from the local device's memory and added to the *doc_list* of the application. Figure 3.10 (d) shows files listed in the file browser for the local device. View screen also allows users to read and modify the opinion for other devices. When user selects opinions option from the menu a list is displayed showing the device details along with its associated opinion. The user can modify and save these values. It must be noted that the value of opinion is used in computing the trust values associated with each peer.

Send: A user can send a message or a document to a recipient. The message can be typed and user can be selected from the neighbour list displayed on screen. The message is sent to the selected recipients. User can also select a file from the *doc_list* and send it just like a message. Figure 3.11 (a) shows a message typed by the user to be sent to a selected recipient.

Request: The request document screen allows user to request a *doc_list* or a document. Initially the list of neighbours is displayed with no information about the documents. When a user selects a neighbour and requests document list from the selected device, the *doc_list* for the selected user is appended to the neighbour list. When more connections are available, all document lists from neighbouring users will be displayed. User can select the documents he is interested in and request these. Figure 3.11(b) shows doc-list from neighbouring nodes.

User can terminate the session by exiting the application from the main screen. The next section presents the experiments done using the framework described in this section.



Search

Devices Available

☑ 08:23:d4:76:90:a5

☐ 07:64:32:49:4e:4d

☐ 08:22:98:54:a3:09

(a) Search results displayed

View

Neighbors List

UID	IP	t
08:23:d4:76:90:a5	A0, B3, A8, A2	0.8
07:6		
08:2		

(b) View screen displaying n_list

Files in doc_list

☑ Image32.jpg;A3;24330;07:64:32...

☑ Image37.jpg;C5;18500;07:64:32...

☐ Image53.jpg;B0;25873;07:64:32...

☐ Image58.jpg;A3;33319;07:64:32...

☐ Image72.jpg;A3;1800;07:64:32:...

(c) Files listed in doc_list

fileBrowser

..

Backgrounds/

DARK.JPG

DARK2.JPG

LIGHT.JPG

Pictures/

(d) File browser

Opinion (29)

08:23:d4:76:90:a5 A0, B3, A8, A2 0.8

0.7

07:64:32:49:4e:4d A1, B3, A8, C2 0.5

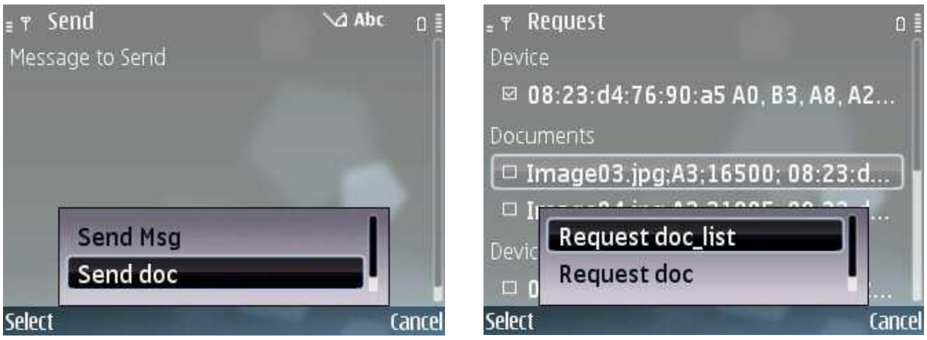
0.6

08:22:98:54:a3:09 A0, B3, D7, C2 0.6

0.5

(e) Screen displaying Opinions

Figure 3-10: Search and View screens



Send

Message to Send

Request

Device

☑ 08:23:d4:76:90:a5 A0, B3, A8, A2...

Documents

☐ Image03.jpg;A3;16500; 08:23:d...

(a) Sending a message or a document (b) requesting documents

Figure 3-11: Sending and Requesting documents

3.4 Experimental setup and results

The objective of the experiments is to validate the framework's design and to collect information on contact opportunities in P2P mobile applications. Furthermore testing of the trust module in the proposed framework would provide important information of users mutual trust ratings. The developed prototype has been successfully tested on HP iPAQ 211 PDA running Microsoft Windows Mobile 6.0.These devices are capable of Wi-Fi and Bluetooth communication. It was observed, while running Wi-Fi interface, the battery drains in a couple of hours whereas with Bluetooth interface the device can run for up to 6 hours. For the experiments, Bluetooth connectivity is preferred due to its energy efficiency. The Bluetooth device discovery is performed every 2 minutes (+/- small delays for synchronization purpose) for the duration of 10.24 seconds, which is recommended minimum duration by the Bluetooth standard. Only 3 device-to-device connections are allowed at a time with RFCOMM links although Bluetooth permits up to 7 connections.

User trials were conducted with 7 users each equipped with a Bluetooth enabled HP iPAQ 211 PDA running the prototype. Each PDA has the P2P file sharing application installed. In the experiments conducted, all users run the P2P file sharing application, where users can announce their files and share. User trials were run for an average of 3 days (approx. 35 work hours) in a campus setting where users move between classes in the same building, twice for the framework with and without the trust module. For the purpose of quantifying the number of connections made, active and inactive times are used. A*ctive time* is the time while a device is running any of the prototype applications. Each device is re-charged whenever the battery is depleted. The time when the prototype application is not used is *inactive time*. Not all the users previously know each other and a few have pre-existing social relationship, the experiments intend to exploit this with the proposed framework to view the opportunities created for interaction. Table 3.1 shows the characteristics of collected data set for both sets of user trials.

3.4.1 Opportunistic contacts

In the initial experiment trust management is not considered and users are allowed to make contacts. During the trial period the average active time for all devices was 21.2 hours (60%). The average inactive time was 13.6 hours (40%). It can be noted that the amount of inactive time is quite high, this is primarily due to the battery depletion, consequently making a user go off-line for recharging. While this can also be partly due to normal use of mobile devices, or because the prototype adds to the energy consumption due to frequent Bluetooth operations.

Table 3-1: Characteristics of collected data sets

	without trust	**with trust**
Duration	35 approx.	35 approx.
Active time	21.2	20.9
Inactive time	13.6	13.7
Bluetooth connections	529	491
Successful connections	271	247
Total messages sent	252	217
Total messages received	155	136
Total files sent	94	81
Total files received	55	42

During the trial period a total of 529 Bluetooth contacts were made between all devices of which 271 (51%) were successful connections while the rest were refused or dropped. Bluetooth discovery was used to identify other devices, prior to RFCOMM connection establishment for exchange of document lists and files. A total of 155 messages were successfully received (75% success rate). Maximum size of a file is set to be 1 MB for transfer, a total of 55 files was successfully received (58% success rate) with a size of 1MB or lesser. The results were recorded for the experiment to be utilized for the purpose of comparison with experiment described in the next section.

Despite of many limitations a set of interesting results were obtained in terms of opportunistic relationship building and communication. Figure 3.13(a) presents a friendship graph for all the users before and after the experiment. The initial friendship graph has a mixture of connected and disconnected nodes. After the completion of the experiment, the user friendship graph has a high degree of connectivity (average 4.8 connections per user) which shows that users were able to contact almost all other users and establish connections thus evolving a well defined community. Figure 3.12 plots the established successful connections, messages and files received by all devices over the period of the experiment. Ratio of successful connections, messages received and files received between the two experiments can also be seen from the figure.

3.4.2 Trust based opportunistic contacts

The previous experiment showed that peers can connect, communicate and share a number of files successfully in a given period of time. It is possible however that the service requested from a peer is not completed as expected. As an example, a peer requesting a particular file receives a different file because the peer sending the file intentionally sent the wrong file (such as a file containing a worm or virus). Furthermore a peer with malicious intentions can send false or misleading information to distract or distort information such as messages, interest profiles and files. In the second experiment the trustworthiness of users in their interactions is considered. The purpose of this study is to understand the impact of the light weight trust model defined in section 3.2.3 on user interactions both trustworthy and malicious in nature.

The trust model defined earlier relies on trust ratings and opinions of peers for other peers in the network for computing trust ratings. A false trust rating received from malicious peers could reduce the trust ratings in the network, for this reason the received ratings are averaged using equation 3.1. In this experiment, initially all peers are considered to be trustworthy; therefore the initial trust ratings and opinions for all users were set to be 0.5. Out of seven users, four would continue to be trustworthy and honest in all interactions; the rest of the peers are going to perform in an untrustworthy manner. For the sake of confidentiality, the three malicious users were not identified to the rest of the users. The malicious users perform three levels of malicious activities.

- User M1 is always deceptive and provides false information and false recommendations. If T_{ij} is the true trust value, M1 will reply with [1- T_{ij}]
- User M2 is frequently deceptive and provides false information 50% of times. A random Boolean variable was used in the framework to reply with T_{ij} (the true trust value) or [1- T_{ij}] as false trust rating.
- User M3 is rarely deceptive and provides false information rarely (10% of times).

As mentioned earlier the opinions are based on positive α and negative β interactions between peers. In this experiment due to the limited size of the dataset, the contextual information is not available, therefore opinions cannot be calculated. A framework user can decrease or increase the value of opinion about a peer, manually when a requested service from a peer is completed or otherwise. In determining the opinion, the users were advised to reduce the value of opinion by 0.1 if one of the following cases occurred for every instance of file transfer:

- File received not the same as expected
- File type not the same as expected
- Incomplete file received

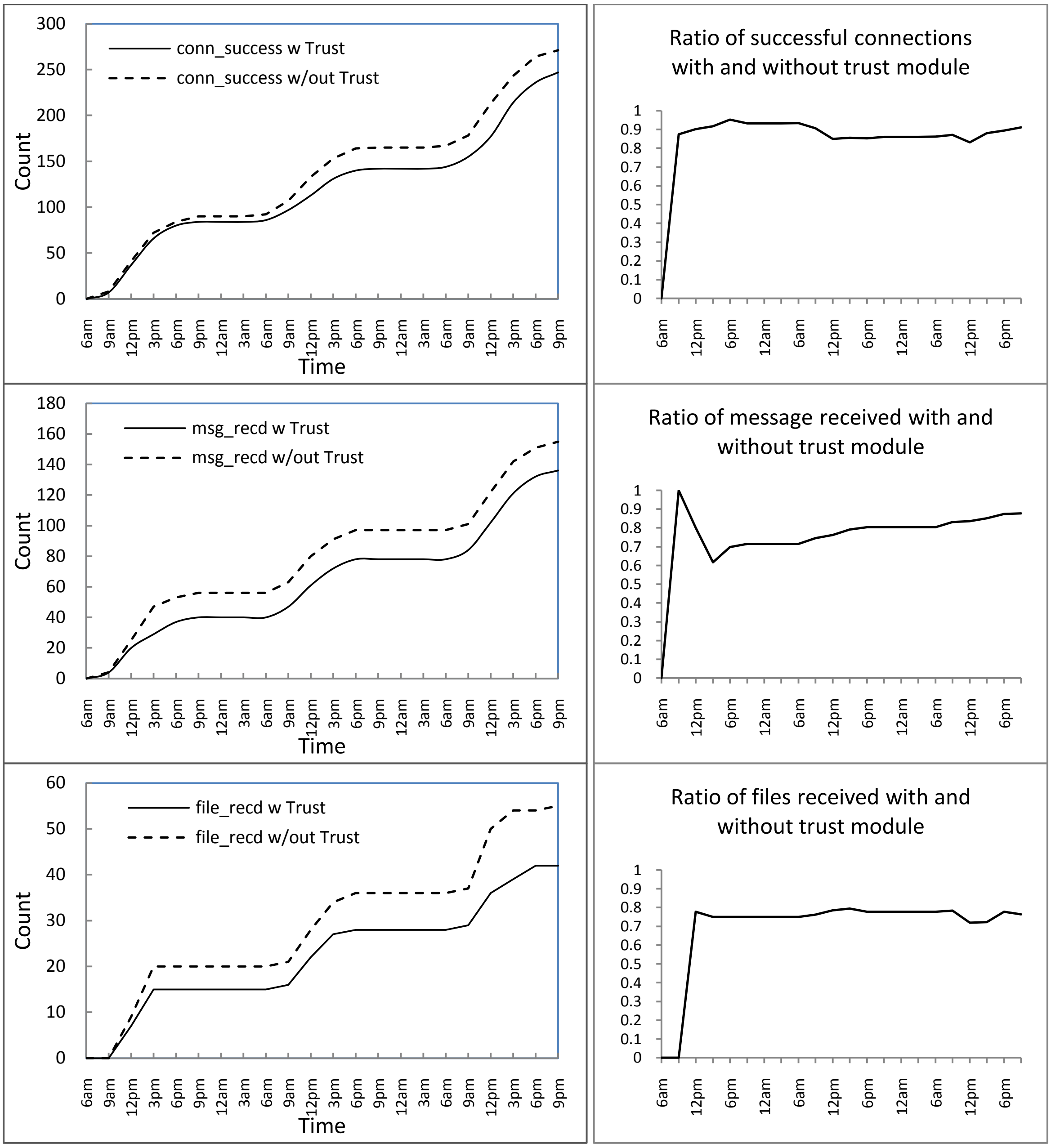
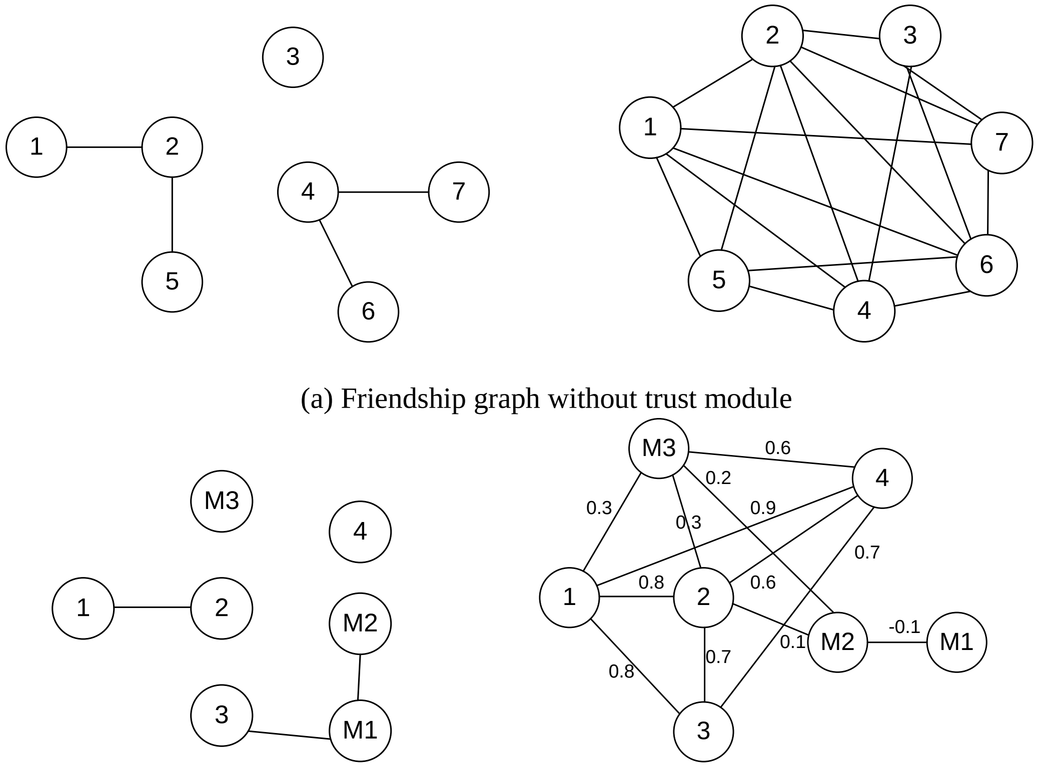


Figure 3-12: The sum and ratio of successful connections, messages and files received over time

Consequently, for a completed as expected service, the users increase the value of opinion by 0.1. The connections dropped due to Bluetooth connectivity issues were not considered for having an impact on the value of opinion. All users request an update for trust ratings after every one hour.

With similar parameters and limitations in the first experiment, the number of Bluetooth connections was observed to be 491 of which 247 were successful (50%). A total of 136 messages were successfully received (72% success rate). A total of 42 files were successfully received (52% success rate). The average success rate for messages delivered and file transfers completed is similar to the first experiment. This shows the additional burden of calculations for trust management module has a minor effect on the performance. Comparison of results from both experiments can be seen in Figure 3.12.



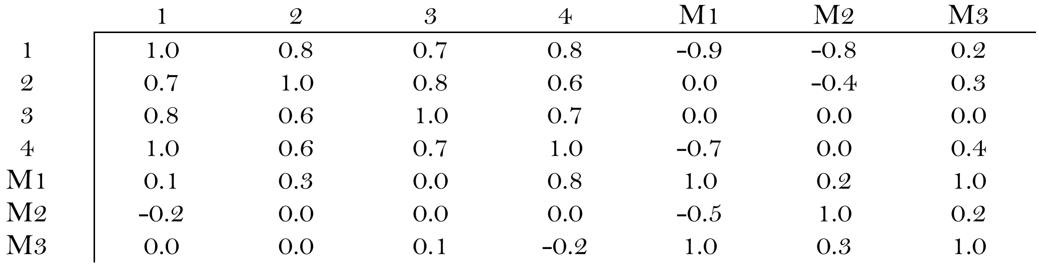
(a) Friendship graph without trust module

(b) Friendship graph with trust module. Edge values depict average Trust value of the vertices of an edge

Figure 3-13: Initial (left) and final (right) friendship graphs with and without trust component.

Figure 3.13(b) shows the friendship graph before and after using the trust based framework prototype. The normal (honest) nodes are able to create contacts with other nodes whereas malicious nodes are partially isolated due to the untrustworthy behaviour. The average value of trust between two nodes is computed and shown as edge value in the graph. The average trust value between honest nodes is higher (>0.5) where as between an honest and malicious node is significantly lower. It can be seen that node M1 initially had two contacts but lost connection with all nodes except node M2 which is also a malicious node. The average trust rating between M1 and M2 is a negative value (-0.1) which suggests that the connection between M1 and M2 would be broken if the trust value fall below -0.2. The friendship graph for the prototype utilizing the trust module proves that the experiment was successful in identifying untrustworthy nodes.

Figure 3.14 shows the trust matrix for trust values (*t_list*) in all nodes at the end of the experiment. For the purpose of acknowledging the existence of connection between an honest user and a malicious user, if the trust rating falls below -0.2, it was not removed from the *t_list* although this rating was not used in computing the trust values. It can be seen from Figure 3.8, the trust rating of malicious user M1 is -0.9 in (*t_list*) of user 1, however the rating is +0.1, yielding an average of -0.4 which identifies M1 as an untrustworthy user.



	1	2	3	4	M1	M2	M3
1	1.0	0.8	0.7	0.8	-0.9	-0.8	0.2
2	0.7	1.0	0.8	0.6	0.0	-0.4	0.3
3	0.8	0.6	1.0	0.7	0.0	0.0	0.0
4	1.0	0.6	0.7	1.0	-0.7	0.0	0.4
M1	0.1	0.3	0.0	0.8	1.0	0.2	1.0
M2	-0.2	0.0	0.0	0.0	-0.5	1.0	0.2
M3	0.0	0.0	0.1	-0.2	1.0	0.3	1.0

Figure 3-14: Trust Matrix for trust ratings ($T_{i,j}$) in (*t_list*) for all users

3.5 Related Work

Most of the work in mobile social communications has been commercial and centred around sending location and status updates from mobile devices towards centralized (and proprietary) activity aggregation services (and then possibly again back to the mobile devices as notifications). Examples include Dodge-ball [DODGE], Twitter [TWITT], and facebook [FACE]. In contrast to these the framework functions mainly in ad hoc mode. MIT's Serendipity [EAGL05] was a socially motivated project based on Bluetooth proximity detection. Similar projects based on the basic idea of Bluetooth based contact discovery were also presented in [MIKL07] [NICO06] where the proximity data is stored on a central server and can be later visualized through a Facebook application. Mobile Social network middleware architectures have been proposed in [YAOJ07] and a technique for profile based query routing presented in [TOMI06]. Both these techniques rely on centralized servers.

Recently MobiClique [ANNA09] presented a middleware for mobile social networking using opportunistic contacts and Bluetooth node discovery. It utilizes the user proximity to detect new contacts and help create new types of communities in a mobile social network. It does not however consider trust management in messages and data transfer and is thus susceptible to security flaws. As opposed to MobiClique, in addition to the opportunistic store-carry-forward protocol, the proposed work incorporates trust into the framework architecture and is accessible using a communication API.

Authors in [ELDE09] address the security issues in a DTN based on social contacts. Using trust model and trust ratings between users of a social network, they are able to identify untrustworthy users. Comparatively, the proposed trust based framework addresses the trust management issues by leveraging social trust ratings and applies opportunistic contact discovery protocol and allows content transfer between nodes. Furthermore the framework can be used to study mobility and social contact behaviour of users.

3.6 Summary and Discussion

The framework proposed allows forming of trust based communities in a disconnected and delay tolerant MANETs. Users can share content and transfer files in an opportunistic manner utilizing store-carry-forward paradigm. A framework was designed in J2ME Personal Profile and tested on devices using Windows Mobile 6.0. Using the framework this work demonstrates through two experiments, the successful construction of communities between nodes that contact each other opportunistically in close proximity and ad hoc manner. The trust management module manages trust ratings based on reputation from neighbouring users. Results prove the effectiveness of trust management module, nevertheless various factors having an impact on results need further investigation.

Due to many limitations the experiments were carried out with a rather small dataset and limited resources, the framework needs to be tested in a large scale environment to fully investigate the social contact and file transfer opportunities without pre-existing social contacts. Despite of a variety of advantages for using the prototype, the proposed framework needs further improvement. Here a list of shortcomings in the framework design is presented and points to be addressed in further research work are discussed.

- Bluetooth discovery is expensive in terms of repeatedly scanning neighbourhood every 11 seconds (approx.). When the connection is established users must identify themselves and share the documents list. Currently RFCOMM is used for file transfer for files of all sizes. The results show approximately 50% of connections drop due to Bluetooth limitations. Therefore there is a need for a light weight protocol to be used for discovery and file transfer. Chapter 4 introduces a content driven light weight data dissemination protocol that addresses user discovery and file transfer with immediate neighbours.
- The limit of a maximum of three connections introduced in the experiments reduces the ability of contacting distantly located users. In a disconnected ad hoc network that is sparsely populated, a user should be able to discover and contact distant users through referrals from intermediate users. Moreover users should be able to store content, carry it to a new location and forward it to other users should an opportunity arise. Moreover the protocol used in the framework sends content to the requesting neighbouring node. Unicasting packets to a large number of users, is an expensive process and has a detrimental effect on the performance of the devices and the network. Therefore a multicasting technique for data dissemination over a multihop ad hoc network has to be developed. Chapter 5 addresses these issues by developing an adaptive opportunistic routing protocol for disconnected MANETs.
- The current trust module considers trust ratings from immediate neighbours and assumes all ratings to be true. It is however possible that a set of users may collaborate together to provide collective false ratings to artificially boost the ratings of a malicious user. There is a need for a stronger trust module that acknowledges group based trust ratings. Since the trust function utilizes the sum of averages of trust ratings received from neighbours, a group based trust function would reduce the impact of false trust ratings. Work presented in chapter 6 addresses the group based trust management.
- In the experiments carried out the number of Bluetooth connections per device was limited to 3, although in a MANET setting a node can maintain connections not only to the immediate neighbours but also to distant nodes over a multihop network. Consequently a user may receive ratings of other from distant parts of the

network thus increasing the chances of a collusion attack. To prevent such attack the trust model needs to accommodate not only direct interaction trust ratings from immediate neighbours but also witness trust ratings from distant node in the network. Chapter 7 addresses detection and prevention of collusion attacks in multi-hop disconnected MANETs.

Chapter 4

Content Driven Data Propagation Protocol (CDDPP)

4.1 Introduction

Content-based communication is a style of communication, whereby the flow of information is interest-driven rather than destination-driven [COST06]. An interested node in a network would subscribe to the kind of information that it is interested in; the provider of this information would simply send the information in the network without addressing it to a specific destination node. Content-based communication allows a clear decoupling between senders and receivers. For this reason it is especially suited to being used in ubiquitous computing environments, in which it can serve as a communication paradigm for applications dedicated to information sharing, news distribution, service advertisement and discovery, etc. [CALE08].

In connected wired networks, content-based communication can usually rely on a logical, content-based routing infrastructure, which itself can be implemented as an overlay network over the physical point-to-point network [CARZ01]. This underlying infrastructure is then used to route each message towards interested hosts whenever needed. In a disconnected MANET such as that shown in Figure 4.1 there is no guarantee that an end-to-end path (based on a succession of one-hop links) can ever exist between senders and interested receivers in the network. In such networks the store-carry-forward approach is considered suitable and has been used in recent research work [HAIL08] [HUIP08].

Previous chapter mentions the OSCF protocol that was implemented as part of the framework. The OSCF protocol facilitates communication between hosts using Bluetooth connections for transmissions. OSCF protocol also sends complete files between peers without managing packets and relies on Bluetooth library for access control resulting in poorer delivery rates and incomplete and broken files due to disconnections. This chapter presents a simple Content Driven Data Propagation Protocol (CDDPP) for data sharing in delay tolerant MANETs. CDDPP is a light weight data propagation protocol and does not rely on costly methods for constructing and maintaining complex routes. The protocol is designed so as to minimize the global amount of data transmitted on the wireless medium, while avoiding unnecessary retransmissions. Nodes in the network maintain a list of profiles and discover other nodes with similar interests. Nodes sharing common interest profiles can share messages as well as documents as long as they can be in the transmission range of each other. Messages are broadcast to all nodes in the network reducing the overall cost of multiple transmissions. This concept of content driven data sharing is similar to social networking where users share information based on their interests. CDDPP can be implemented as part of the framework defined in chapter 3. The proposed protocol in this section does not however implement any trust management models.

The purpose of this protocol is to facilitate content driven communication using opportunistic store-carry-forward paradigm in a disconnected MANET. CDDPP protocol uses broadcast messages for transmission for immediate neighbours. The protocol is light weight and performs communication between nodes in three simple steps including announcements, invitations and sharing of interest profiles and sending, receiving and processing documents. The CDDPP protocol is tested in a simulation environment with hundreds of nodes in the network. Simulation experiments compare the CDDPP with a greedy version of itself. The results show that the proposed protocol is effective in content based data delivery when node storage size is limited.

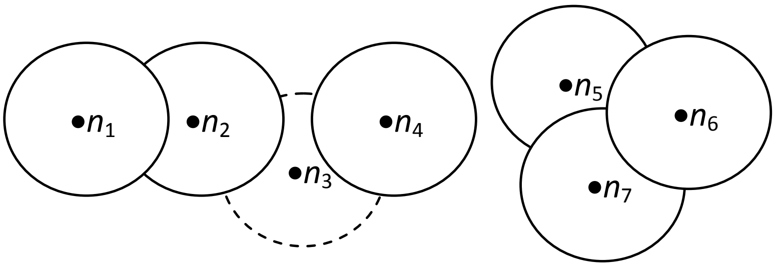


Figure 4-1: Illustration of a disconnected MANET.
Circle around a node depicts transmission range. Dotted circle indicates node is about to get disconnected.

4.2 CDDPP Protocol Design

This section presents a content driven protocol where nodes in an ad hoc network share data only if they are interested, i.e. a node would send or receive messages, store data and forward the message only if it is interested and hence routes would be established in opportunistic manner with nodes having similar interests. Routes can be established to distant nodes if they also show interest, provided that a relaying node is able to forward message in a multi-hop manner. This however requires the essential storage capability at each node for storing messages as transient messages for later transmission to the intended destination. The proposed protocol relies on broadcast transmission for announcement and point to point transmission for destination oriented messages. Broadcast transmissions are also used for single hop transmission sending messages to neighbouring nodes depending on the number of requests received for a particular message.

The packet format for CDDPP follows the header structure given in Figure 4.2. With six fields the size of the header is 22 bytes. All nodes in CDDPP are identified by their unique MAC address; therefore the source and destination fields in the message take 6 bytes each. The next field is the packet ID field, in 4 bytes it contains the number of packets sent from the source node. The source ID and packet ID can be used to uniquely identify a packet. The unique ID of individual packets is needed when broadcasting packets to ensure that a packet is not forwarded twice. The time-to-live (*ttl*) field takes 1 byte. Packets received with *ttl* less than 0 are discarded. The field payload with length of 4 bytes designates the number of data bytes expected to follow the header. The Packet Type field identifies one of the following packets in one byte.

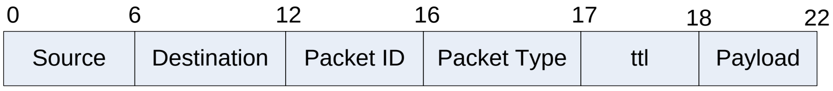


Figure 4-2: CDDPP Packet header format

- **announce**: Every device emits a announce packet periodically. The announce packet contains four interest profiles. The interest profiles are used for discovery and identification of peers with similar interest profiles. Each announce packet is appended by eight bytes of four interest profile keywords. Each keyword takes two bytes and is of the form A0, B3, C8 etc.
- **invite**: If a peer receives a announce request and is willing to share with the requesting peer, it replies with a invite packet. The replies received from neighbouring nodes help populate the list of neighbours *n_list*. A session is established when two peers with similar interest profiles proceed with further transmission. The invite message includes the document list *doc_list* for the host node. The *doc_list* is implemented as a linked list of document objects where each

object takes 28 bytes. Figure 4.3(a) shows format for *doc_list*, Figure 4.3(b) shows a *doc_list* for a host.

- **request**: The request packet is used to request two kinds of information including an updated *doc_list* and a document (document id, type, size, ownership, timestamp). In case of a fresh copy request for a *doc_list*, the request packet is sent without any data appended. The receiving node replies with the requested *doc_list*. For a document request, the requesting node appends the list of document objects needed with the request packet. It is highly likely, in mobile communications, packets can be lost in transmission. For a missing packet, a re-send request for a missing packet can be answered by any node in range having the missing packet stored in the cache, otherwise the request is forwarded.
- **send**: The send packet delivers two types of information. If an updated document list is requested, an updated *doc_list* is sent. In case of a document request, a document (document id, type, size, ownership, timestamp) is sent to the requesting node. A Boolean variable is used to discriminate between the two types of send packets.
- **data**: Indicates that the packet contains data. Depending on the type of packet, the data is formatted for quick retrieval by nodes.

0 12 14 18 24 28

Filename.ext	Type	size	Owner	Time stamp

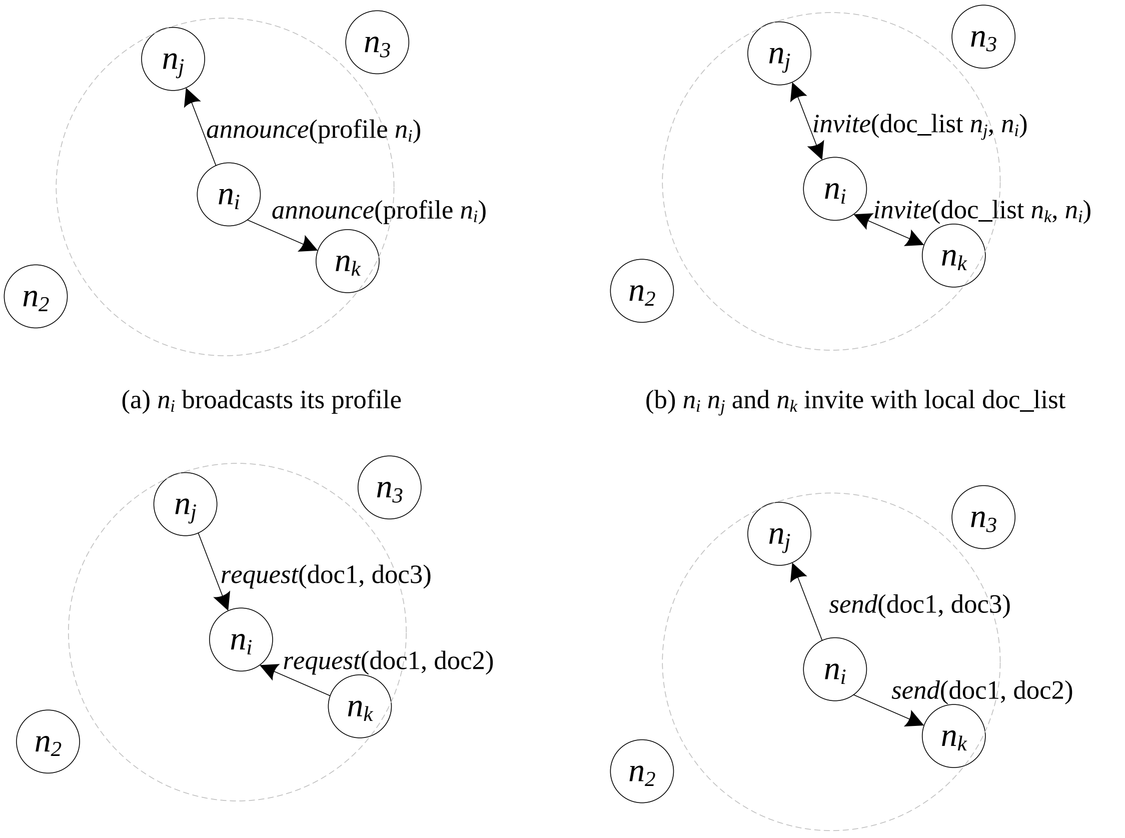
(a) doc_list format

Image31.jpg	C5	23518	09:26:37:3A:45:90	22.35 10.02.09
Fileac.txt	A3	238	07:64:32:49:4E:4D	09.19 12.12.09
Image45.jpg	C4	18845	09:26:37:3A:45:90	19.55 12.12.09
Image48.jpg	C5	9853	09:26:37:3A:45:90	04.05 16.12.09

(b) four tuples in doc_list for a node

Figure 4-3: Document list *doc_list* format

To make the model simple, a three step process is followed for all transmissions. Each node n_i periodically broadcasts a *announce*(n_i) message containing interest profile of the user. Neighbouring nodes n_j and n_k receive this announcement and process the interest profile. If willing n_j sends an *invite*(n_i) message to n_i including document list of n_j. n_i responds with its own *invite*(n_j) including list of documents for n_i. Both nodes would parse document list and may tag documents to be shared. For a document with a unique identifier to be requested by n$_i$ a *request*(n_j, *doc-id1*, …) is made upon which n_j would *send*(*doc-id1*, …) the required document as shown in Figure 4.4. These three transmissions are detailed as follows.



(a) n_i broadcasts its profile

(b) n_i n_j and n_k invite with local doc_list

(c) n_j and n_k request documents from n_i

(d) n_i sends requested documents to n_j and n_k

Figure 4-4: Transmissions for host n_i

- **Announcing Interest Profile**: In a neighbourhood of nodes announcements for personal interests are made. A host n_i periodically broadcasts *announce*(n_i) including its interest profile. By broadcasting its own interest profile, a host lets its neighbours know what kind of documents it is interested in. on the contrary, by receiving similar information from all its neighbours, each host can adjust the *doc_list* it broadcasts periodically, thus avoiding to transmit a *doc_list* pertaining to documents that cannot interest any of its current neighbours. Adjacent nodes receiving this announcement match their own interest profile keywords, if the receiving host is interested, it sends an *invite*() invitation to the announcer as a unicast transmission. Consequently, if the receiving host is not interested in the interest profile, it simply ignores the announcement. This use of unicast transmission aims at avoiding the replies from several neighbours of n_i that received the request and assume that they are expected to answer to it. For example, in the configuration shown in Figure 4.4 (b) the request sent by n_i may be received by n_3 if it was not sent explicitly to n_j. This would potentially lead to some documents being broadcast several times in answer to a single request.

- **Inviting interested host**: When an announcement from n_i reaches a node n_j, it compares the interests in the users interest profile. If any of the keywords match, the receiving host n_j may be interested in starting a conversation. It therefore creates a *invite*(n_i) message to be sent to the originating node n_i. This *invite* contains a documents list including document attributes such as a Unique Identifier for the document, Document size, Document type, ownership and a Timestamp. It is assumed that the size of the *invite*() may not exceed 300 bytes thus keeping the payload of transmission to minimal. When the originating node n_i receives the invite message from n_j, it may send its own invite to n_j describing a list of n_i's current documents. When both nodes receive each other's invite messages they can process the document lists to search for an interesting document to share. If there exists such a document, it can be tagged for sharing among these two nodes. Any tagged document may be sent if requested. Information about neighbouring nodes is stored in the local routing table referred to as *n_list*.

- **Requesting, Sending and Storing Documents**: Nodes that had a chance to look at the document lists of each other can request or send documents. As described earlier a document-list contains attributes for each document stored in a node's repository. These attributes include a Unique Identifier for the document, document size, document type, ownership and a Timestamp. If the node n_j requires a document *doc-1* that is available in repository of node n_i it would send a *request*(n_i, *doc-1*) message to n_i. To process the request n_i would proceed by forwarding the document *doc-1* to the requesting node by embedding the document in the *send*(*doc-1*) message. This send message is forwarded and is intended only for the requesting node n_i. When a document is received, it has to be stored in the node's repository and the documents list is updated. It is possible that many adjacent nodes would request same documents, in which case the requests are processed sequentially. As with the case of ad hoc networks a new or returning node can enter the range of n_i and start communication; if a node n_k enters the moment n_i sent the broadcast, n_k would receive a copy of the document, which can be saved in the repository of n_k. In this case a message needs to be sent to all requesters. This consequently would decrease the performance due to overhead of repeatedly sending the same message. As a solution to this problem maintaining a list of adjacent nodes at all times is suggested. If a simple majority of hosts request same documents a broadcast message is sent to all nodes instead of individual messages.

An important feature of CDDPP protocol is the ability to store packets destined for other nodes, referred to as data caching. These packets are delivered at a later time if an opportunity arises for data transmission provided the intended node is available. This phenomenon is referred to as opportunistic store carry forward. This approach is fairly useful in delay tolerant MANETs where nodes can go out of network coverage for a period of time and then return later. As an example from Figure 4.4 (c), due to mobility, node n_j physically relocates to another position and is disconnected from node n_i. Since the transmission session is aborted, n_j will repeat the announce and invite process and try to discover neighbours. Assuming that a node n_1 is collocated in the transmission range of n_j, both nodes establish connection and share files of mutual interest. Node n_j can also share files received from node n_i. If at a later time n_1 moves in the range of n_i, it can forward the files received from n_j to intended destination n_i thus serving as a store-carry-forward node.

4.3 Simulation & Results

The Content Driven Data Propagation Protocol (CDDPP) has been implemented in Java and interfaced with Madhoc [HOGI] simulation tool. Madhoc is a MANET simulator that allows the simulation of large wireless mobile networks in metropolitan environment. Details about the simulation tools are provided in appendix A.

4.3.1 Simulation Parameters

A number of 15,000 iteration / seconds, simulations are run to study the various conditions of the protocol based on many parameters. These parameters are discussed as follows. In the experiments it is assumed that each user is equipped with a laptop device or a Wi-Fi enabled PDA device. Each device has an Omni directional transmission range of 100m.

There are 100 users in a 1000m x 1000m environment. This environment consists of various spots with a random size no larger than 100m x 100m. These spots can be considered as shops or other buildings. The transmission range is reduced to 40 m when inside a spot due to various factors. The users move between spots using a variant of Random Way-Point mobility model (RWP) [BROC98] [BETT02] implemented as part of Madhoc. In the RWP mobility model, each node randomly chooses a destination location (in terms of its x, y coordinates) in the simulation area following uniform distribution and moves towards this destination with a determined velocity. When the destination is reached, the station remains at the same place for a while. Once this time expires, the node chooses another random destination (following uniform distribution) in the simulation area. The node then travels toward the newly chosen destination at the selected speed. This process is repeated by each node until the end of the simulation. Further details about the RWP can be found in appendix A. For the mobility model, it is assumed that the user moves with a speed of 3 m/s when not in a spot and 2 m/s when inside the spot area; amount of mobility within the spot is set to 60% and outside is 40%. User may pause for up to 100 seconds to look for a destination. 32 different interest profiles are defined in the experiment. Each user in the network is randomly assigned four distinct interests at the start of the simulation. These interest profiles are matched to create pairs of users willing to share documents.

4.3.2 Communication Scenario

Users create documents with varying sizes (32KB - 512KB) and store in the host repository with an average global document creation rate of 1 document every 5 seconds. It is assumed that the user’s repository is limited therefore a bound is placed on the size of the repository set at 10 mega bytes. Hosts broadcast an announce message every 15 seconds, this delay is introduced because at pedestrian speeds 15 seconds is generally considered as an adequate time for MANETs [HAIL08]. A host willing to share, announces four interests in its profile, any neighbour with at least one of the similar interests, sends invite to share documents. At a certain time if the repository is filled and no further documents can be stored, the node in question would remove the least recently used document to make space for a newer document.

Table 4-1: Parameters used for simulation

Mobility model	Random Waypoint Mobility model
Number of nodes	100
Number of interest profile (keywords)	32
Repository size	10 MB
File size used	32KB, 64KB, 128KB, 256KB, 512KB
Simulation time	15000 s
Area	1000x1000 m
Spot area	100x100m
Spot velocity	2m/s
Normal velocity	3m/s
Pause time interval	0.1 - 100s

To evaluate the proposed protocol its performance is compared with a modified version of the same protocol. In the modified version of the protocol, every host requests for every possible document from a neighbour with no limits to numbers of documents being shared, thus being a greedy host. The consequence of the greedy host protocol would be that each host requests and stores documents it may not be interested in, but these documents can be forwarded later to other interested hosts.

4.3.3 Message delivery rate vs document size

Figure 4.5 shows a comparison between the numbers of documents received by both protocols with documents of size 64KB. The proposed CDDPP protocol proves to be more efficient in document delivery. As the number of documents created is more than the documents received by either protocol, it can be seen that CDDPP protocol received 91% of the documents. The greedy protocol however, is less efficient in this regard as it receives only 48%. In the beginning of the simulation the rate for documents received by either protocol is much lower, the reason is that it takes time for documents to disseminate in the network. The number of documents received by the greedy protocol (98%) is higher than CDDPP protocol (52%) in the beginning of the simulation, i.e. up to 3000 sec; since greedy protocol enthusiastically searches and stores more documents regardless of relativity to the interests, for that reason it is able to obtain more documents. Another point to be noted is the limited space available in each host's repository creates frequent updates as the space quickly fills up in the beginning of the simulation. When there is no space to store a newer document, the node looks for the least recently used document and removes it from the repository. This technique for making more space obviously has a disadvantage of removing some documents before these are even shared on the network.

As time progresses in the simulation, the document delivery rate of the greedy protocol decreases due to the frequent updates of the repository as can be seen in Figure 4.6. Due to these updates many documents in the repository need to be removed to make way for newer documents thus decreasing the availability of a shared document. At the end of the simulation (15000 simulation seconds) the documents delivered for CDDPP surpasses the greedy protocol. The document delivery ratio for the CDDPP protocol is 90.7% compared to only 47.8% in the greedy protocol. Graphs showing comparison of greedy protocol and CDDPP protocol with various document sizes can be found in appendix A.

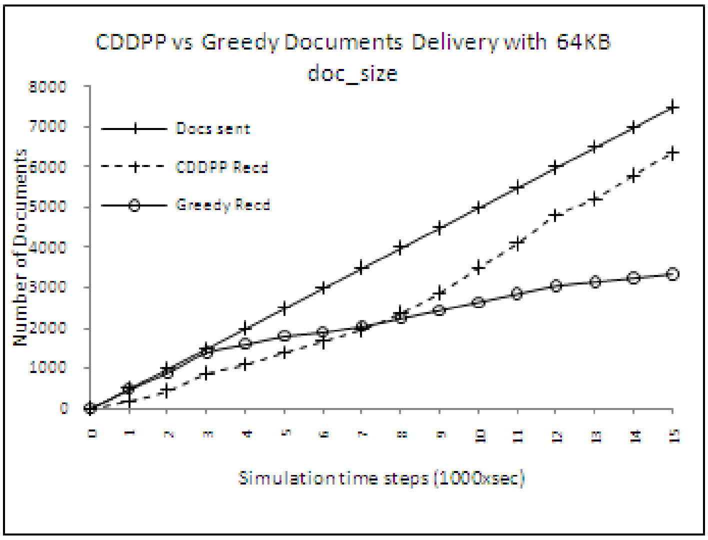


Figure 4-5: Number of documents received against sent for CDDPP and greedy protocols

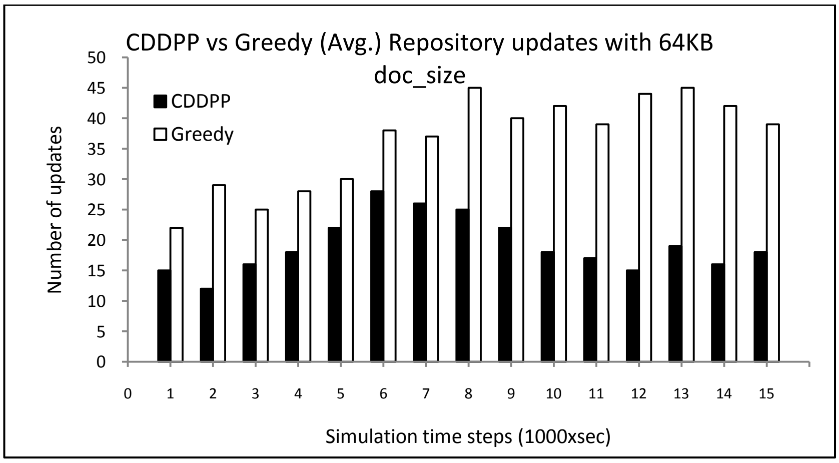


Figure 4-6: Comparison of Average number of repository updates for Greedy protocol and CDDPP protocol with 64KB document size

Figure 4.7 shows the comparison of delivery rates for both protocols based on document size. The size of document is set to 64, 128, 256 and 512 Kilo bytes. It can be seen, with all document sizes the delivery rate is much higher, i.e. more than 80% for the CDDPP protocol, but is lower for the greedy protocol. The size of documents affects the delivery rate for documents using the greedy version of the protocol. Figure 4.8 shows number of documents delivered over time with 128 KB document size. The graph shows the difference in packets sent versus packets received using CDDPP and Greedy protocol. It can be seen that the number of repository updates continuously increase with greedy protocol compared with CDDPP, which shows that CDDPP protocol reduces the overall amount of number of updates required. Further results with different document sizes can be found in appendix A.

4.4 Summary

Users of Mobile Social Networks share data only if they are interested, therefore there is a need to create a content driven communication protocol for disconnected MANETs. This chapter presented a simple CDDPP protocol for data sharing in disconnected MANETs. The protocol is light weight and does not rely on costly methods for constructing and maintaining complex routes. The ability of a node in MANET to store, carry and forward documents has been exploited to allow users to announce their interest profiles, documents and share them. A node therefore successfully announces its documents stored in repository and shares them with other users. Documents thus stored are carried to other locations and are shared with other users having similar interest profiles. Simulation results shown in the previous section prove that the CDDPP protocol is effective in propagating documents between senders and interested receivers thus successfully disseminating and forwarding messages in single-hop connections in the network.

CDDPP protocol does not however consider data propagation over a multi-hop topology. It can be seen from simulation results that greedy protocol is more efficient in delivering documents to a larger set of users in the start of the simulation and is handicapped due to the limited repository size and the frequent updates of the repository to accommodate new documents. Moreover after two users have negotiated interest profiles, respective document lists are broadcasted to all neighbouring users, thus creating a flood of traffic. This inefficiency in the protocol design reduces the overall performance. The next chapter presents an Opportunistic Routing Protocol (ORP) that extends the CDDPP protocol by addressing node discovery and data sharing over multiple hops. ORP solves the problem of flooding by applying an adaptive approach of selective multicast messages to neighbouring nodes over a multihop topology while utilizing the store-carry-forward transmission paradigm.

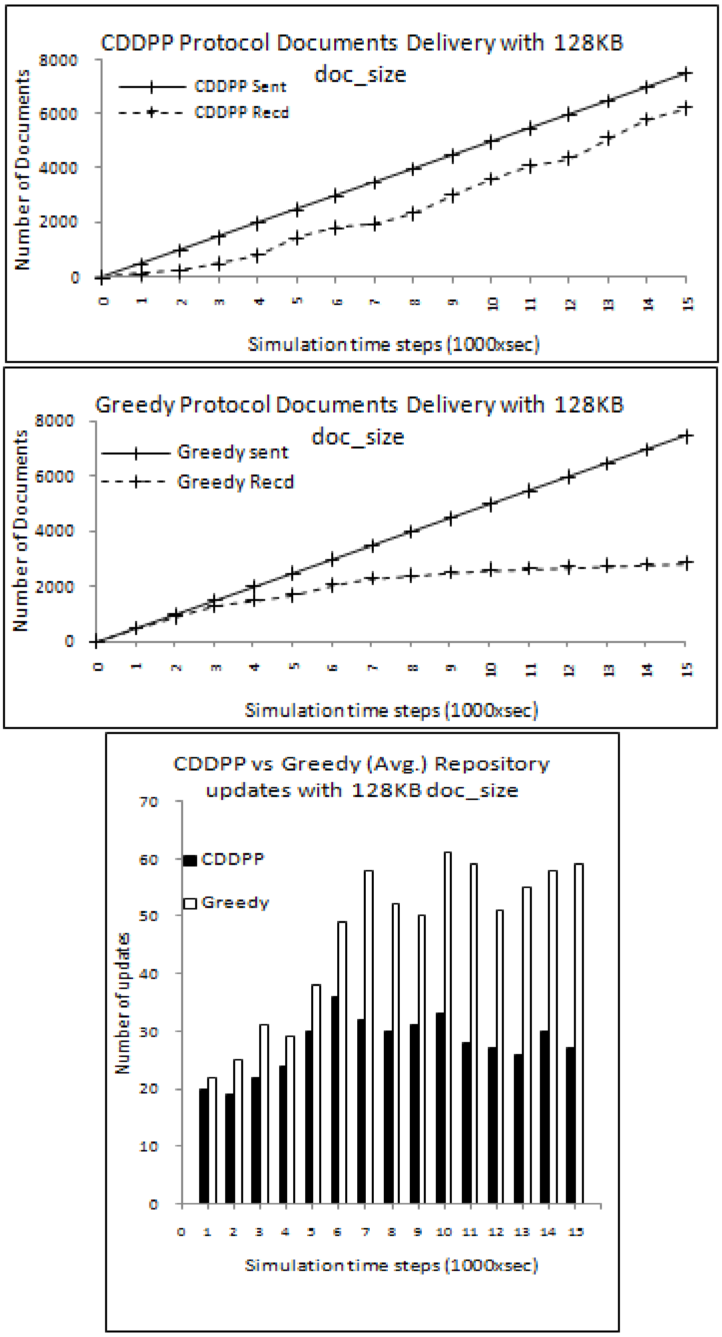


Figure 4-7: Comparison of Packets delivered over time between CDDPP and Greedy protocols along with repository updates.

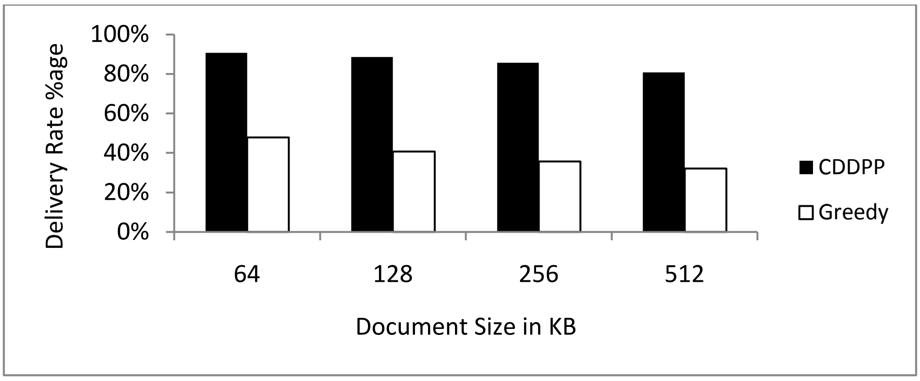


Figure 4-8: Comparison of Delivery Rate for CDDPP and Greedy protocols

Chapter 5

Opportunistic Routing Protocol (ORP)

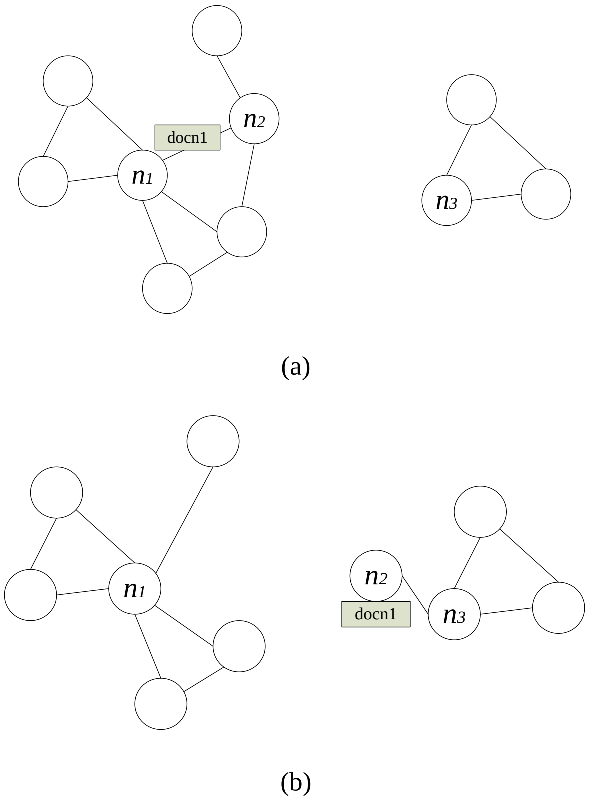
5.1 Introduction

The Content Driven Data Propagation Protocol (CDDPP) presented in chapter 4 is a simple and light weight protocol for content based node discovery and data sharing in a direct neighbour (single hop) paradigm. In a mobile P2P file sharing application, users do not necessarily have to interact with physically co-located nodes (neighbour / friend). It is possible that a user can communicate to a friend of a friend in a multi-hop manner, i.e. a node that is not necessarily present in the common physical location. Distant nodes can be reached by using the store-carry-forward ability of a node [SHAH03] explained in section 5.2. Given these limitations an improved version of the CDDPP protocol is designed that incorporates the store-carry-forward ability of nodes in a disconnected MANET over multiple-hops. The proposed Opportunistic Routing Protocol (ORP) implements routing of messages in a disconnected MANET where nodes can communicate based on user interests (content based data delivery) to distant nodes in a multi-hop communication model.

The ORP is designed to incorporate with the framework defined in chapter 3. ORP extends the communication layer defined as part of the framework in section 3.2.2. The improved version of the framework implements content driven data propagation using the opportunistic store-carry-forward paradigm over multiple hops in the mobile P2P network. ORP does not however, implement trust management module, yet it provides platform for facilitating trust management in the framework. This chapter defines the concept of store-carry-forward communication model for disconnected MANETs in section 5.2. Section 5.3 gives a detailed account of the components of the ORP protocol. Section 5.4 carries out simulations and evaluation of results. Summary of the chapter along with discussion on the results is presented in section 5.6.

5.2 Store-Carry-Forward Model in disconnected MANETs

As stated earlier, disconnected MANETs are severely limited in seamless connectivity due to the proneness of frequent disconnections. Given the inherent characteristics of these networks, if a connected path does not exist, the delivery process has to rely on store-carry-forward mechanisms. Opportunistic routing mechanisms can be used inside communities to spread these messages to a population (network). Figure 5.1 shows an example of a store-carry-forward model. Node n_1 is a part of network A and it needs to send a document to node n_3 in network B. Since there is no direct path between the nodes, n_1 forwards the document to n_2 considering the fact, when opportunity arises, n_2 will forward the document to n_3. In this manner node n_2 stores and carries the document until it sees a opportunity to forward it to the intended recipient i.e. node n_3 at a later time. The existing store-carry-forward routing methods in disconnected MANETs can be classified into two categories according to the mobility control. The first category exploits the mobility of nodes to transmit messages, but does not change their original random movement. The second category is controlled movement, where nodes may change their original trajectory to deliver messages.



(a)

(b)

Figure 5-1: An example of store-carry-forward in disconnected networks
(a) Node n_1 transfers document docn1 to n_2. n_2 stores and carries the document
(b) n_2 moves to become part of n_3's network and forwards the document to n_3

Epidemic routing [VAHD02] is the typical random movement scheme and has been used by many researchers in the area. Epidemic routing is a flooding-based algorithm, where nodes are all mobile and have infinite buffers. When a node has a message to send, it propagates the message to all nodes it meets, which continue to propagate the message. Eventually the data is delivered to the destination with a high probability in a bounded amount of time. An example of epidemic routing is the PRoPHET [LIND03] and the CAR [MUSO09] protocols. PRoPHET determines the best custodian store and carry node with the highest probability for delivery. CAR presents methodology for calculation of delivery probabilities. Socially-aware routing schemes such as Bubble-rap discussed in [HUIP08] and [COST06] describe forwarding protocols based on the social network structure of the individuals carrying the devices. These protocols can be very effective in places where social ties among members of communities are traditionally very strong. Moreover, in this case, the system should also support persistent caching and broadcasting of the messages for a certain interval of time on the relays (gateways) in order to be able to spread the messages to the devices of users in their proximity.

5.3 The Opportunistic Routing Protocol

Opportunistic Routing Protocol (ORP) is defined to contain three components, application component, content dissemination component and content store-carry-forward component. The application component supports the user interface and works as an interface for application layer in the network protocol layer stack. The content-dissemination component provides support for content driven data dissemination in the form of documents and messages. It manages sending and receiving messages to neighbouring nodes in the network, inquiring about common interests and validating a node to be a friend. A friend node is usually a neighbour with at least one similar interest. A neighbour must be within the range of the node thus being a member of the same group of nodes. It is also responsible for sending, receiving and storing documents in the repository of the node. The third component, content store-carry-forward component is responsible for data forwarding to distant nodes in a multihop manner. Figure 5.2 shows the three components of the protocol.

5.3.1 Application Component

For content based data propagation, users of an application such as mobile P2P file sharing must maintain a public interest profile. A typical interest profile may comprise of name, picture, contact information, gender, relationship status/interests, activities/hobbies, musical preferences, literature interests, group membership, and, of course, friendship information concerning user interconnection. A profile is designed to introduce a person to other members of the network announcing personal information, interests, location and a list of documents to share. If a user makes a search, his personal interests are matched in a database and query results are returned. The user may choose to select from a number of interested users and send an "invite". The invited user receives the invitation message, if interested he responds and the two users become friends. Friends can show their documents publicly and may even share them. A user announces his documents to a friend, if the friend is interested he can request a document which can be a range of mutual interest files that can be anything from personal information to audio/video clips. The Mobile File Sharing / Social Networking application already defined in section 2.3.3 and section 3.2.1 is used for evaluation purposes. The application implements interest profiles, document lists and a document repository for evaluation purposes.

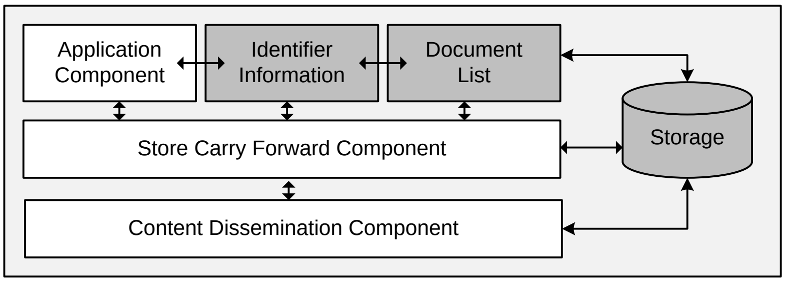


Figure 5-2: The ORP Protocol Components

5.3.2 Content Dissemination Component

Content Dissemination component defines interaction between neighbouring nodes. A neighbouring node is within the range of the node interested in communication, thus being a member of the same group of nodes. To make the model simple a three step process for all transmissions is followed. A node n_i periodically broadcasts a *announce*(n_i) message containing interest profile of the user. Neighbouring nodes n_k receive this announcement and process the interest profile. If willing, n_k sends an *invite*(n_i) message to n_i including document list of n_k. n_i responds with its own *invite*(n_j) including list of documents for n_i. Both nodes would parse document list and may tag documents to be shared. For a document with a unique identifier to be requested by n_i a *request*(n_k, *doc-j*, …) is made upon which n_k would *send*(*doc-j*, …) the required document. These three transmissions are detailed as follows.

Announcing and Receiving Interest Profile: In a neighbourhood of nodes announcements for personal interests are made. A host n_i periodically broadcasts *announce*(n_i) including its interest profile. Adjacent nodes receiving this announcement compare their own interest profile keywords and update the list of nodes maintained in the repository. This list simply acknowledges the presence of neighbouring nodes with similar interests that are also active. If a node does not reply to an announcement, the cleanup function removes the node from the list. If the receiving host n_k is interested, it sends an *invite*(n_i) invitation to the announcer. Consequently, if the receiving host is not interested, it simply ignores the announcement.

Inviting interested host: When an announcement from n_i reaches a node n_k, it compares common interests in the user's interest profile. If any of the keywords match, it can be implied that the receiving host n_k may be interested in starting a conversation. The sending host creates a *invite*(n_i) message to be sent to the originating node n$_i$. This invite contains a documents list including document attributes such as a unique identifier for the documents, document size, document type, ownership and a timestamp. It is assumed the size of the *invite*() may not exceed 300 bytes thus keeping the payload of transmission to minimal. When the originating node n_i receives the invite message from n_k, it temporarily stores the incoming document list from n_k and documents in the list are tagged if need be. n_k may send its own invite to n_j describing a list of n_i's documents. When both nodes receive each other's invite messages they can process the documents list to search for an interesting document to share. If such a document exists, it can be tagged for sharing among these two nodes. Any tagged document may be sent if requested. If a host doesn't receive any requests for sharing a document it is possible that either the pairing host is uninterested in the document list or perhaps has lost communication because of radio interference.

It is also possible that while invite messages are being sent, the nodes would physically go out of range or even into suspend mode when no communication is possible. In this case the neighbouring node may wait for a while for a retransmission, if there is no retransmission the message would be dropped. This failure of communication is of little consequence because it is clear that if a mobile host misses an opportunity, it may get another chance albeit with another neighbouring node in the future.

Requesting, Sending and Storing Documents: Nodes that had a chance to look at the document lists of each other can request or send documents. As described earlier a document-list contains attributes for each document stored in a node's repository. These attributes include a unique identifier for the document, document size, document type, ownership and a Timestamp. If the node n_i requires a document *doc-j* that is available in repository of node n_k it would send a *request*(n_k, *doc-j*) message to n_k. To process the request n_k would proceed by first receiving the request and confirming the availability of the requested document and looking for the possibility if the document in question could be shared. If all consideration are clarified the n_i proceeds by forwarding the document *doc-j* to the requesting node by embedding the document in the *send*(n_k, *doc-j*) message. This send message is forwarded to the requesting node n_k. When a document is received, it has to be stored in the node's repository and the documents list is updated. This document would still have the same attributes such as a unique identifier, timestamp and ownership. Since the document was received from n_k therefore its owner would be listed as n_k in the repository of n_i. Thus over an extended period of time a node's repository may contain many documents from neighbouring nodes.

5.3.3 Store-Carry-Forward Component

This component is responsible for storing and forwarding documents from immediate neighbours to distant nodes using neighbours over a multi-hop connection.

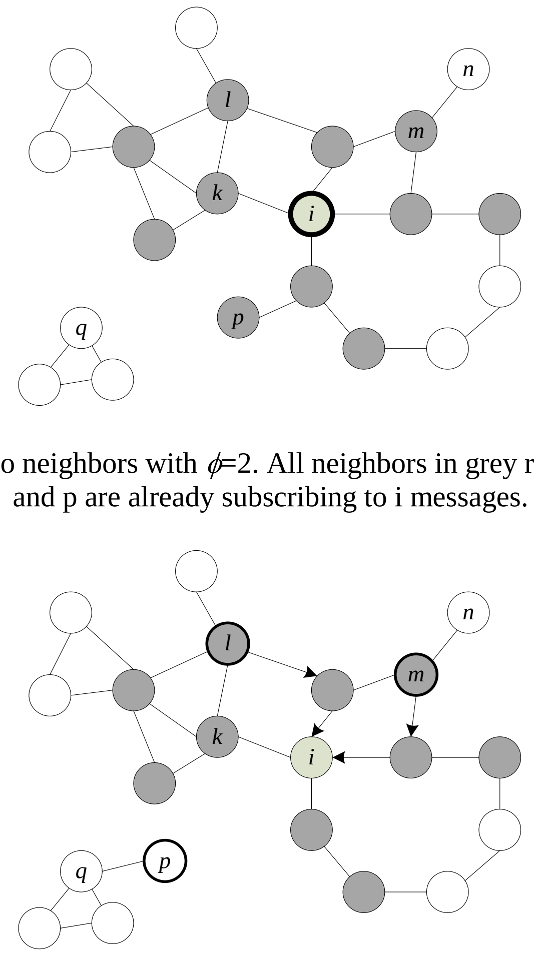
Multihop transmission: As mentioned earlier the protocol broadcasts *announce* messages to all neighbours. However it sends messages to *invite* requests only to the nodes requesting information. It has been seen that multicasting messages in a multihop manner could be more effective in sending the announcement across the maximum breadth of the adjacent nodes in the network. This can effectively eliminate repeated multiple broadcasts of announcements to neighbours. Multihop broadcasting in MANETs creates flooding and is considered to be a bandwidth consuming activity [HARR05], therefore the limit to number of hops towards a destination (ϕ) is defined. Broadcasting to only ϕ number of hops can limit the flooding of network and thus is very effective in controlling the overall traffic in the network [CAUS09]. Example of transmitting broadcast messages over the network is shown in Figure 5.3(a), where ϕ is set to be 2. Node n_i sends announcement to all neighbours up to single hop count. Notice all nodes in range are coloured grey; the message is relayed up to the grey nodes and not any further. Immediate neighbours of n_i receive the message and forward it further to their neighbours to the next level. Node n_l

and n_m receive the message and send invite requests to the originating node n_i as seen in Figure 5.3(b).

Store carry and Forward: Previously the send and receive procedures are defined for message delivery between neighbouring nodes. As soon as the invitation is accepted by a node, a list of document types with similar interests is compiled and sent to the invited node. A copy of requested document is sent to the interested user and therefore stored in the repository. This document can be forwarded when an opportunity arises such as shown in Figure 5.3(b). it is assumed that nodes n_i and n_p have already shared some documents, n_p moves towards the group of nodes including node n_q located farther away from n_i. n_p announces its interest profile to n_q, similar interests are found between the two nodes. A link is established and an invite is sent to n_q. n_p can now send documents owned by n_i that were previously stored in its repository to n_q therefore working as a relay between n_i and n_q.

Due to stringent constraints associated with MANETs the repository size is to be limited. As the documents are received they would be stored in the repository thus reducing the amount of space left. With the increase in document size, receiving and storing a large document in the limited capacity repository essentially required updates. For every update, each document in the repository is checked for time stamp, the least recently used document is removed to make room for the newly received document.

Multicasting messages: It is possible that many adjacent nodes would request the same document from a host; in this case a copy of the same message needs to be sent to all requesting nodes. This would greatly decrease the performance due to overhead of repeatedly sending the same message. As a solution to this problem *n-list* (list of adjacent neighbouring nodes) is used. If a simple majority of hosts request the same document, a broadcast message is sent to all immediate neighbours (ϕ = 1), instead of individual messages. As with the case of ad hoc networks a new or returning node can enter the range of n_i and start communication. If a node n_k enters the moment n_i sent the broadcast, n_k would receive a copy of the document, which can be saved in the repository of n_k. Experiments carried out in this work, present results of multicasting messages and prove that nodes receiving messages accidentally due to this broadcast tend to store the content for later use that is to be forwarded to other nodes.



(a) node i sends request to neighbors with ϕ=2. All neighbors in grey receive the request. l, k, m and p are already subscribing to i messages.

(b) nodes l and m send to requested documents to i. p has migrated to a different group and can forward documents from i to q.

Figure 5-3: Data forwarding in the ORP protocol

Researchers in [WIES00], [BANE03] and [LIUB08] have addressed the issue of energy efficiency in transmission of broadcast and multicast protocols for mobile wireless networks. [WIES00] proposed three multi-cast algorithms based on broadcast incremental power algorithm and compared the efficiency in transmission with broadcast protocol. The proposed algorithms determine a minimum cost multicast tree and use this information for data packet transmission. Results using simulation experiments show that sending multi-cast messages is more energy efficient when the group size for requesting nodes is smaller. With the increase in the multicast group size, the efficiency decreases. Furthermore, it has been shown that determining the minimum cost multicast tree is a difficult problem and can be modelled as NP-complete [WIES00]. In this work, it is assumed that broadcasting a single document requested by multiple hosts is better (in terms of energy efficiency in transmission) than sending multiple unicast messages to each requesting host, provided the number of requesting nodes is at least half of the number of nodes in the local *n_list*, otherwise multi-cast messages would be sent to the requesting nodes.

5.4 Simulation Environment

The proposed protocol in section 5.3 has been implemented in Java and interfaced with MADHOC [HOGI] simulation tool. A number of 15,000 iterations / seconds, simulations were run to study the various conditions of the protocol based on many parameters. These parameters are discussed as follows.

5.4.1 Mobility Model

In mobile networks, devices are usually carried by humans so their movement is necessarily based on human decisions and social behaviour. To capture this kind of behaviour, it is assumed that people carrying the devices may form groups or move individually in the simulation area. Since movement is driven by social relationships, the simulation area is divided into a grid of 5 x 5 in the experiments. Each host moves in the simulation area using a Random Way-Point mobility model (RWP) [BROC98] [BETT02],

RWP model is a very popular and frequently used mobility model in evaluation routing protocols for MANETs and has been extensively used in evaluation of routing protocols presented in [BALD05] [MAHE08] [GUID07] [MUTH05]. It is a simple and straightforward stochastic model that describes the movement behaviour of a mobile network node in a two–dimensional system area as follows:

- The initial positioning of the nodes is typically taken from a uniform distribution. The nodes are typically placed in a square or a circular (disc) area.
- A node randomly chooses a destination point in the area and moves with constant speed to this point.
- After waiting a certain pause time, it chooses a new destination, moves to this destination, and so on.
- The pause time durations are independent and identically distributed random variables.

Traveller nodes are also introduced to study the impact of higher mobility. Traveller nodes move between groups and share content with members of those groups.

5.4.2 Simulation Parameters

It is assumed that each user is equipped with a laptop device or a Wi-Fi enabled PDA device. Each device has an Omni directional transmission range of 100m. There are 100 users in a 1000m x 1000m environment. This environment consists of a 5 x 5 grid where each square size is 200m x 200m. The node speed is generated using a uniform distribution with values ranging [1, 5] m/s. The speed of the traveller node is set to 10 m/s. User may pause for up to 2 minutes to look for a destination.

32 different interest profiles are defined. Each user in the network would have to select up to four distinct interests; these interest profiles are randomly generated for the experiments. If one of the interests for two or more users is common, then these users are likely to start a conversation and share their documents. Since in real-life scenarios, users have various types' of interests and different types of documents to share (text documents, images, videos and audios), it is impossible to predict human behaviour and to the authors knowledge very few models exist that predict human social behaviour. For the sake of brevity, a set of interest profile determines the type of documents a user is interested in. Five types of documents each with a size limitation of up to 1024KB are defined. Every document created in the simulation is saved in the host's repository as (host_number_filenumber.ext) e.g. 4_F5_1.txt i.e. host number 4 creates document 1 of type F5. Documents are created every 100 seconds in the simulation as long as enough space is available in the repository. Figure 5.4 shows a matrix of interest profiles and type of documents a user may have as an interest. As an example, if a node has A0, C4, A3, and D2; as interest profiles, then it must be interested in document types 1, 2 and 4.

The size of the repository is set to 10MB maximum. Hosts broadcast an announce message every 15 seconds, this delay is introduced because at pedestrian speeds 15 seconds is generally considered as an adequate time for MANETs [HYYT06]. A node announces four interests in its profile, any neighbour with at least one of the similar interests, sends invite to share documents. At a certain time if the repository is filled and no further documents can be stored, the node in question would remove the least recently used document to make space for a newer document. This approach would permit a node to get rid of documents which have not been recently requested.

	Document types				
	1	2	3	4	5
Interest profiles	A0	A3	A6	B1	B4
	A1	A4	A7	B2	B5
	C0	A5	B0	B3	B6
	C1	C3	C6	D1	B7
	A2	C4	C7	D2	D4
	C2	C5	D0	D3	D5
	D7	D6	D6	D7	D6

Figure 5-4: Interest Profiles and document types

5.4.3 Compared Protocols

Content based routing in multihop networks has recently been an interesting research area. Not many researchers have worked in the area of content sharing in opportunistic data storing and forwarding. Some protocols have been presented for opportunistic data forwarding [PELU06] and routing in DTNs [JAIN04]. However these protocols do not address opportunistic content based data forwarding in delay tolerant MANETs. Content Based Multicast protocol (CBM) [ZHOU00], is also an opportunistic routing protocol but it also has an epidemic routing mechanism that relies on broadcasting messages, therefore is unsuitable for comparison. Baldoni et. al. in [BALD05] present a structure-less content based routing in MANETs. The proposed protocol uses frequent broadcasts for message delivery based on a complex estimation of proximity of potential subscribing nodes. The frequent usage of broadcasting increases flooding in the networks and reduces the effectiveness of the approach. Work presented in [YONE04] use a similar approach and report an adaptive content based routing protocol. The ORP protocol is specifically developed for content based routing in DTN while exploiting store-carry-forward mechanism for content delivery. Autonomous Gossiping (A/G) algorithm presented in [DATT04] is similar to the proposed protocol since it takes opportunistic approach for data transmission; nevertheless it relies on transmissions with the neighbouring nodes only. The A/G algorithm utilizes the epidemic algorithm to spread data items selectively based on vulnerability of other nodes (multicasting), instead of treating all nodes homogeneously and flooding the network. The A/G algorithm is considered for comparison with ORP protocol using transmissions only to neighbouring nodes. Table 5.1 shows the parameters used for comparison between the two protocols.

Figure 5.5 (a) shows a comparison of A/G algorithm with the ORP protocol, comparing the percentage of documents (content) delivered when strictly 2 or more profiles are matched.

As can be seen, A/G performs better because it utilizes selective broadcast and multicast in propagating messages over the network. ORP considers multicast messages to neighbouring users (existing in *n_list*) only if more than 50% users have requested a document. Over a period of time the accuracy of documents received by ORP is better compared to A/G algorithm. However as shown in Figure 5.5(b) A/G creates far more number of documents compared to ORP, and therefore floods the network. This proves that although ORP is slower compared to A/G but is more effective due to selective multi-casting capabilities. Another point to be noted is for a secure application where trust management is of high importance, a selective multi-casting based protocol would provide better privacy for applications in delivering messages compared to a broadcasting protocol.

Table 5-1: Parameters used for A/G and ORP comparison

Mobility model	Random Waypoint Mobility model
Number of nodes	100
Number of interest profile (keywords)	32
Repository size	10 MB
Number of hops	$\phi = 0$
Multicasting threshold	50%
Content (document) types used	5
File size used	32KB
Traveler nodes	0
Profiles used for matching	1, 2, 3
Simulation time	6000 s

The A/G algorithm broadcasts/multicasts documents available in the storage area to requesting users; regardless of the significance of document to the receiver. This allows unsolicited documents to be sent without any request thus creating spam, increasing the amount of traffic in the network. Comparatively ORP protocol allows users to browse documents before being sent. Only requested documents are sent, therefore minimizing the amount of traffic generated. Further results for the comparison of ORP with A/G algorithm can be found in appendix A.

5.5 Evaluation of ORP protocol

The results demonstrate the quality of information dissemination achieved using ORP. The quality of information dissemination is measured on the basis of standard metrics used such as message delivery rate corresponding to document sizes, message payload, repository updates, mobility parameters and communication overhead. All the simulations assume opportunistic profile matching for nodes to be at least one. The following criterion for evaluation of ORP protocol is discussed.

- Most significant evaluation parameter in this study is the data delivery rate; defined as the number of messages (announce, invite and send/receive) received compared to the total set of messages sent.
- A host having at least four interest profiles may be interested in a limited type of documents. User with similar interest profiles may share and forward documents; i.e. at the end of the simulation a user may have several documents received from other users and might have participated in forwarding own documents or forwarded documents to other hosts.
- Another factor for evaluation is the impact of various document sizes on the limited repository. Larger documents may require larger space availability in the repository, if the repository is full, room must be created for the new document thus increasing the repository update having a detrimental effect on the performance. Six different types of document sizes are used to analyze the performance of ORP.
- Since ORP protocol addresses the lack of multi-hop transmissions in A/G algorithm, the impact of data dissemination over multiple hops is a critical evaluation criterion. In the experiments, up to four hop counts are tested in the sparsely populated simulation area and the impact of messages delivered with various payloads is studied.
- The effect of higher mobility rates also suggests an interesting evaluation objective. Higher mobility of traveller nodes improves the chances of establishing contact with more number of nodes. Consequently, mobility may also affect the successful delivery rates lowering the performance of the protocol.

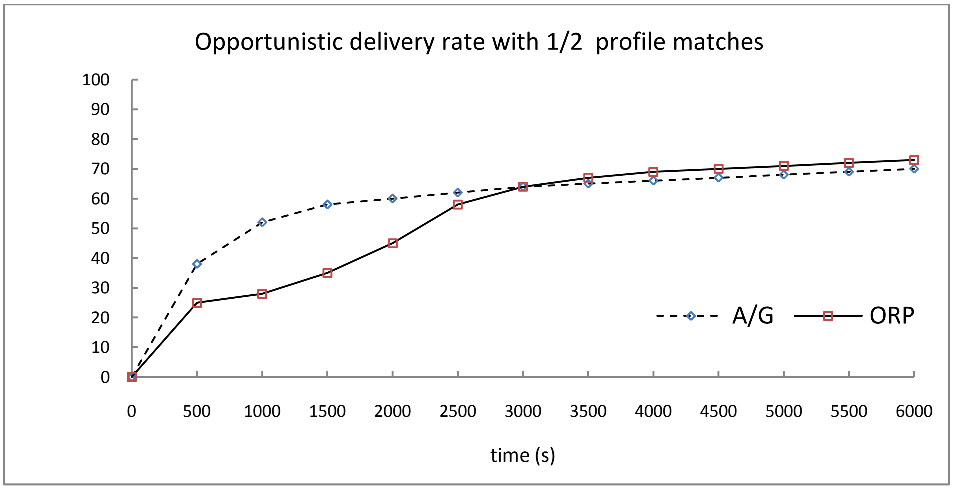
Table 5.2 shows the various parameters used in simulation to evaluate ORP protocol.

Table 5-2: ORP simulation parameters

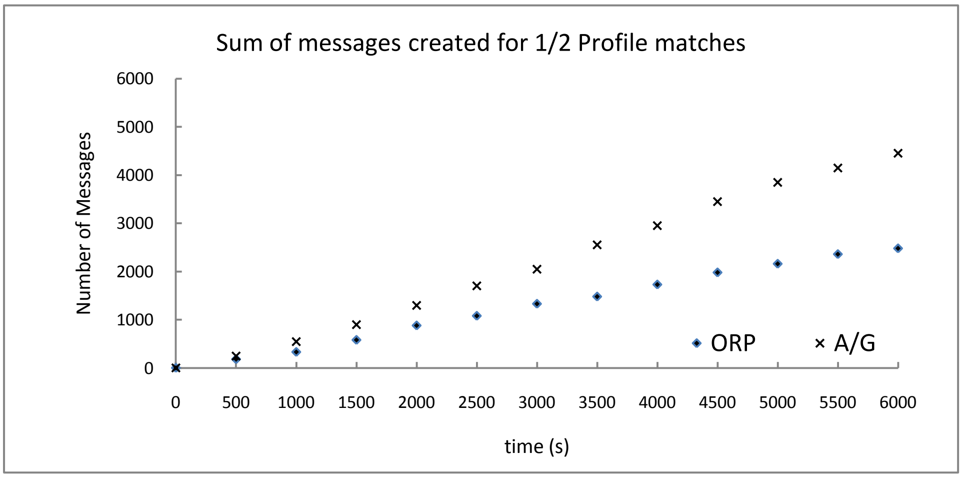
Number of nodes	100
Document sizes	32KB, 64KB, 128KB, 256KB, 512KB, 1024KB
Hop counts	0, 1, 2, 3, 4.
Repository size	10MB
Node speed	[1-5]m/s
Traveler nodes	0, 5, 10, 15, 20
Traveler node speed	10m/s
Profile used for matching	At least one profile (keyword) must match
Simulation time	15000s

5.5.1 Message propagation threshold

The most critical factor in performance evaluation is the propagation of messages in the network. This is determined by measuring the number of message copies forwarded over a multihop forwarding scenario. Ideally the messages delivered over multihop neighbours would be faster than forwarding to immediate neighbours at hop count 0 ($\phi = 0$). Over a number of simulations, it was observed that with ($\phi = 0$), on average, after 4000 sec. only 36% messages were delivered. After 8000 sec., on average, 74% messages were delivered to the requesting nodes and after 12000 sec; 86% were delivered. With ($\phi = 1$), on average, the message delivery at 4000 sec. was 75% and after 8000 sec, message delivery was 93%. This clearly shows that with 1 hop counts neighbours against 0 hop (immediate neighbours), during the same simulation time, message dissemination increased from 74% to 93%. Figure 5.6 shows relationship between delivery rates of messages at various values of ϕ for document size 1024.



(a) Percentage of message delivered with at least 2 matching profiles



(b) Sum of messages created with at least 2 matching profiles over time

Figure 5-5: Comparison of A/G and ORP protocols

With ($\phi = 0$) on average, almost all messages were received after 12000 sec. whereas only 88% messages were received with (ϕ = 1) to the end of simulation. Extending message delivery to (ϕ = 2, 3 and 4) hop counts leads to improved results. With (ϕ = 4) all messages were received at the 9000 sec interval which is a significant improvement over single hop (direct neighbour) scenario. However extending the communication chain to multiple hops increases the communication overhead in terms of message forwarding and repository updates for node lists and document lists.

5.5.2 Document Size and message payload

Transmission and retransmission of heavier payload documents can have negative effect on the storage ability of nodes therefore leading to poor performance of the network. However with the proposed store and forward policy documents can be acquired from nodes available over multiple hops. In further experiments the effect of forwarding documents over multiple hops with limited document size was studied. Figure 5.7, shows the payload of the documents received against the delivery rate. To study the effect of successful delivery of documents over multi-hop nodes with documents of various sizes; multiple simulations were run for 15,000 sec, with document sizes fixed to 32, 64, 128, 512 and 1024 Kilobytes. During the simulation if a node's repository reaches saturation, the least recently used document was removed from repository to make room for the newer documents. With immediate neighbours, the average delivery rate for all sizes of documents was above 97%, i.e. 97% of the time the documents successfully reached the intended destination after transmission.

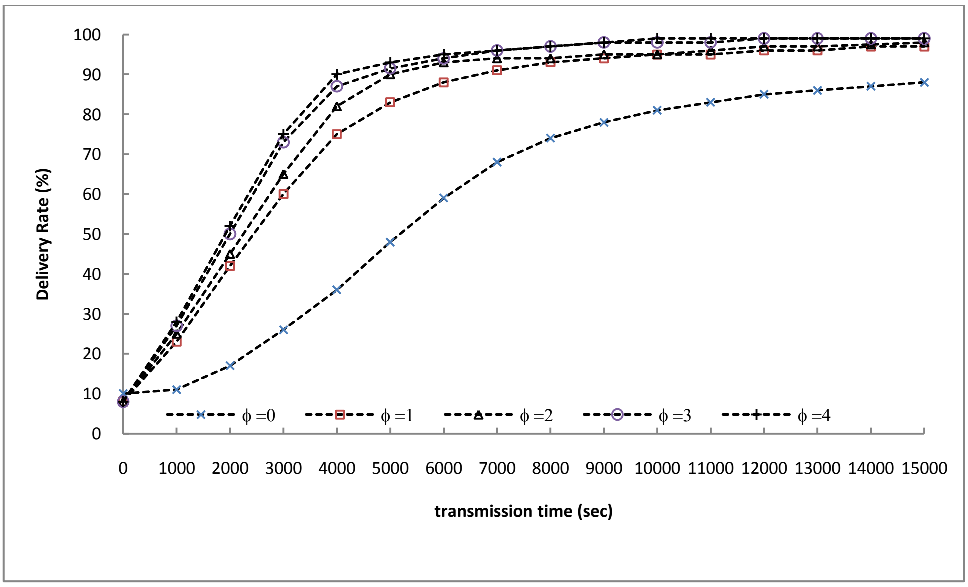


Figure 5-6: Delivery rate of messages over multiple hops (ϕ) with 1024KB size

At 1 hop counts (ϕ = 1), the delivery rate for files larger than 512 KB was 93%, however smaller files reached the destination with more than 95% delivery rate. With the increase in the hop counts, the delivery rate for larger files decreases. For instance in Figure 5.7, with 3 hop counts (ϕ = 3), delivery rate for smaller files with 32 KB payload is an acceptable 97% however with larger files having 1024KB payload, delivery rate is only 63%. This shows that smaller files are effectively delivered even from nodes farther than 4 hop counts therefore increasing the degree of connectedness from farther nodes which suits the network. Table 7.3 shows average percentage of delivery rates with various payloads and multiple hop counts.

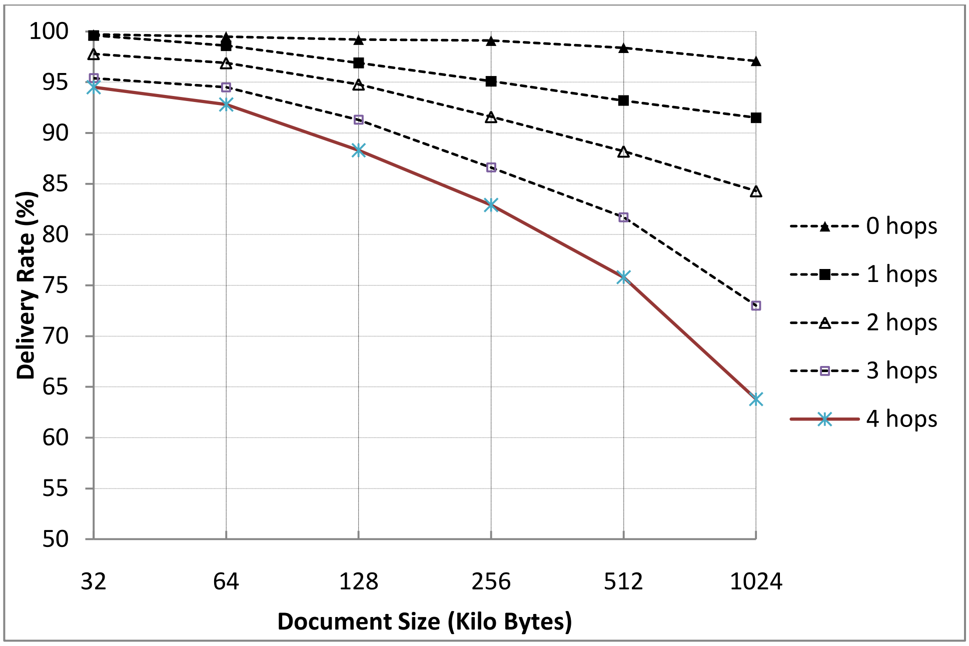


Figure 5-7: Delivery rate of documents over multiple hops (ϕ) with various document payloads

5.5.3 Updating repository with Least Recently Used Algorithm

Increasing the document size for transmission has a negative effect on the performance of the overall transmission due to limited repository size and the need to frequently update the repository. Figure 5.8 shows the relationship between the document size and the number of updates in the d-list and the repository. As document size increase, receiving and storing larger documents in the limited capacity repository essentially required updates. Every update requires each document in the repository to be checked for time stamp, the least recently used document is removed to make room for the newly received document. Simulation results show that with document sizes less than 512 KB, an acceptable rate of fewer than 10% for the updates occurs. However with larger document sizes such as 1024 KB, ORP protocol reports a minimum of at least 17% rate of updates. This of course depends on the limits set for the repository; devices with larger space available for document storage can effectively store documents with fewer updates.

5.5.4 Forwarded documents

An essential criterion for the evaluation of a content driven protocol is the effectiveness of procedure for forwarding content in the network. Precision in determining the percentage of the reached mobile nodes that are actually interested in the data item is essential to success of the protocol. At the end of the simulation the average number of documents in each host’s repository is calculated. Figure 5.9 shows the comparison of the average number of forwarded documents present in the host’s repository against various document sizes. It can be observed that with smaller document sizes (32KB) about 300 documents can be stored in the repository. An average of 43% documents stored, were forwarded documents received from neighbouring users. With larger document sizes (1024KB) this ratio decreases to about 8%. If a document type in the repository for a certain document is similar to the matching interest profiles, as described in Figure 5.4, it is considered to be a related document. The results show that most documents (93%) in the repository are related to the interest profile of a user, hence proving the accuracy of the ORP protocol in forwarding the correct type of content.

Table 5-3: Average delivery rates of various payloads over multiple hops

		Hop counts				
		ϕ = 0	ϕ = 1	ϕ = 2	ϕ = 3	ϕ = 4
Payload (in Kbytes)	32	99.7	99.6	97.8	95.4	94.5
	64	99.5	98.6	96.9	94.5	92.8
	128	99.2	96.9	94.8	91.3	88.3
	256	99.1	95.1	91.6	86.6	82.9
	512	98.4	93.2	88.2	81.7	75.8
	1024	97.1	91.5	84.3	73.1	63.8

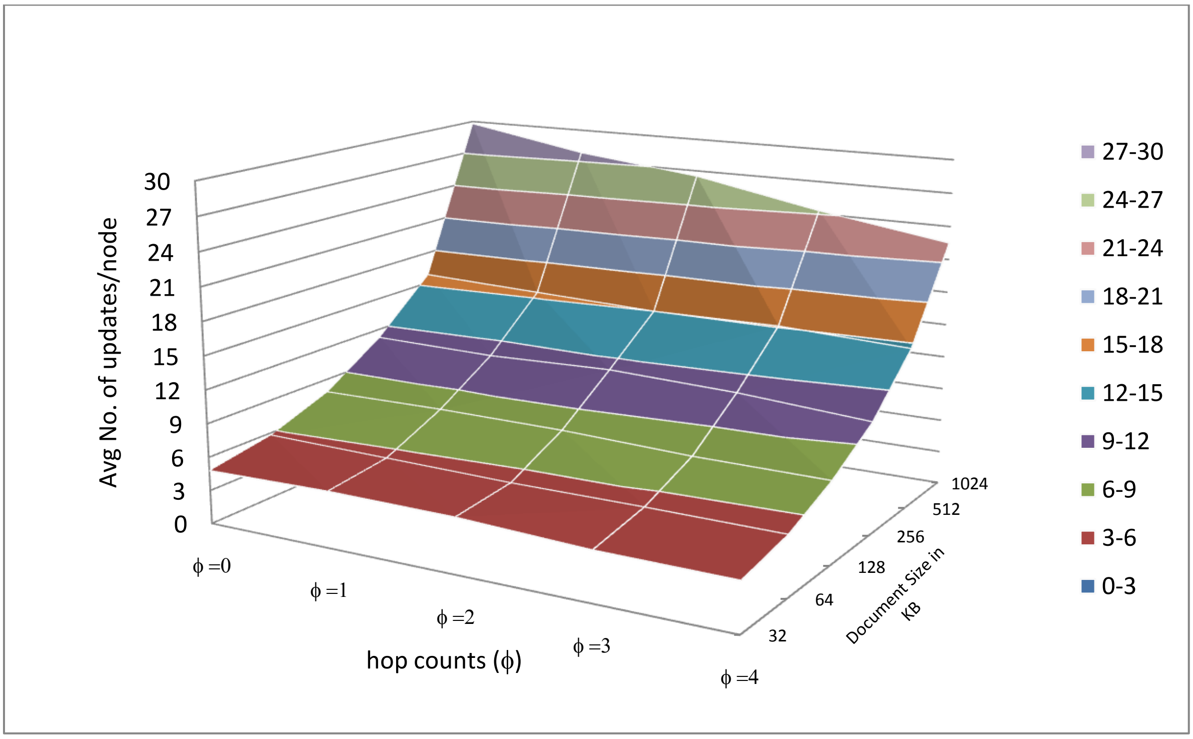


Figure 5-8: Rate of repository update over multiple hops (ϕ) with various document payloads

5.5.5 Mobility Parameters

Speed of hosts is an important consideration in the experiments since a user may travel with varying speeds. Earlier the availability of traveller nodes with greater speeds was discussed; here it is shown that as speed of limited number of users increases the delivery rate also increase, therefore the traveller nodes can efficiently disseminate messages in the network increasing overall documents availability. Table 5.4 shows relationship between number of traveller nodes and delivery rate with hop count ($\phi = 1$) and 32KB document size. It can be observed that as number of traveller nodes increase, the delivery rate for documents also increase. Another important fact is, with increasing number of travelling nodes the average number of forwarded documents received in a nodes repository also increase thus improving the rate of document dissemination in the network (social availability).

5.5.6 Communication overhead

The proposed approach for message forwarding over multihop routes show faster message dissemination in the network. Results also show that utilizing next to the neighbour nodes ($\phi = 1$) provides an effective improvement over communication done with the neighbouring nodes only. However with multihop transmission the communication overhead also increases essentially when intermediate nodes are used for forwarding messages leading to battery drainage and consumption of space in the repository. Due to the ORP protocol's selective message forwarding, the overall transmission cost is reduced. Limits on the repository size provide a bottleneck in a device's store carry forward ability. With smaller files ORP protocol is effective in storing forwarded files and transmitting when possible. However larger files require large amounts of storage availability resulting in frequent updates of the repository thus affecting the performance of the protocol.

Table 5-4: Effect of number of traveler nodes
on document delivery rate and %age of forwarded document in host repository

Number of traveler nodes	Avg. No. of forwarded documents in repository	Delivery rate
0	12%	98.1%
5	18%	90.2%
10	23%	88.2%
15	33%	91.9%
20	60%	96.3%

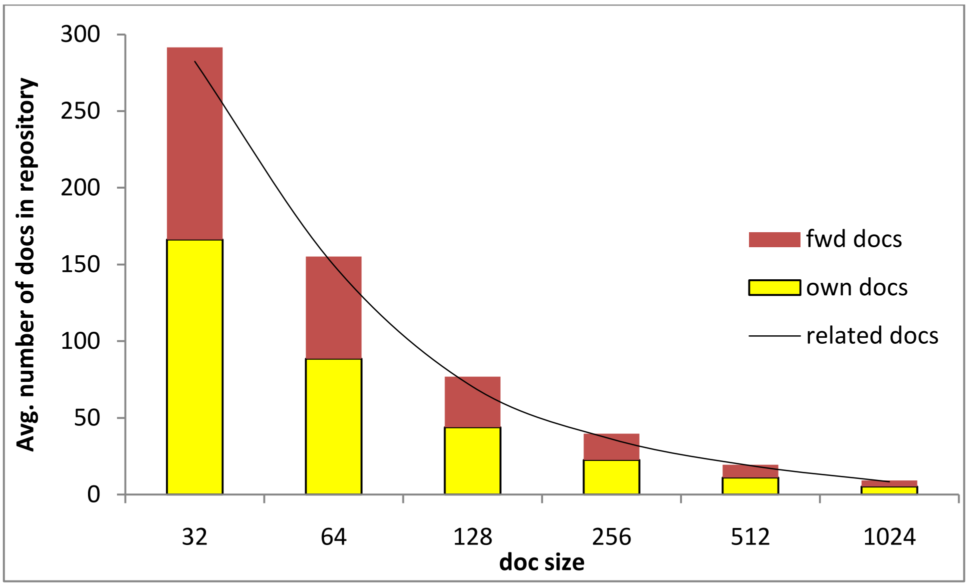


Figure 5-9: Comparison of average number of forwarded documents in a host’s repository again the various document sizes

5.6 Summary

In this chapter a content driven approach for selective data dissemination of information in a disconnected MANET is presented. The proposed Opportunistic Routing Protocol (ORP) is based on an opportunistic routing mechanism for content sharing between users with similar interest profiles. The protocol facilitates discovery of users, announcing of interest profiles, and file transfer between users without flooding the network. Although broadcasting is considered useful for discovering nodes in the mobile network, selective multi-casting is used instead of broadcasting when possible for all data transmissions (file transfer). ORP does not depend on any infrastructure or middleware for route maintenance for store carry forward instead it utilizes the self-organizing ability of nodes at local levels which perfectly suits the disconnected MANETs. Moreover the proposed protocol carries out multi-hop transmissions to extend the range of data dissemination to distant nodes.

The ORP protocol is compared to a similar Autonomous Gossiping (A/G) protocol that utilizes a variety of transmission methods for content based data dissemination. A/G performs better because it utilizes selective broadcast and multicast in propagating messages over the network. ORP considers multicast messages to neighbouring users (existing in *n_list*) only if more than 50% users have requested a document. Over a period of time the accuracy of documents received by ORP is better compared to A/G algorithm.

Comparatively A/G creates far more number of documents compared to ORP, and therefore floods the network. This proves that although ORP is slower compared to A/G but is more effective due to selective multi-casting capabilities. Moreover A/G algorithm broadcasts/multicasts documents available in the storage area to requesting users; regardless of the significance of document to the receiver. This allows unsolicited documents to be sent without any request thus creating spam, increasing the amount of traffic in the network. Comparatively ORP protocol allows users to browse documents before being sent. Only requested documents are sent, therefore minimizing the amount of traffic generated.

Simulations were carried out to test the performance of the ORP protocol with 32 distinct interest profiles and five different types of documents. Message delivery, content forwarding, document size and repository updates were considered for evaluation. Results show that P2P data transfer over multiple hops in the network present faster data dissemination in the network. It was shown that sharing of various sizes of documents over multi hop neighbours is possible with different degrees of success.

In the experiments with mobility of nodes in the network, the ORP protocol improves delivery rates of messages when specific nodes store and forward documents with greater speed into communities of users. Nodes with greater speeds disseminate messages in the network effectively; however with higher speeds seamless connectivity is not always possible therefore only smaller documents can be delivered with success.

The size of data files stored in the repository and the limitation of repository size itself also affects the performance. With larger files the repository needs frequent updates with the possibility of removing files that are to be carried and forwarded to other nodes. Comparatively with smaller file sizes i.e. less than 512KB, the protocol efficiency for repository update and data delivery rate is above 95%.

In the experiments with data forwarding to neighbours at multi hop distances, results show a minimum of 90% delivery rates with up to two hop counts and all sizes of documents. With larger files (greater than 256KB) the data delivery rate is reduced for distant nodes at three or more hop counts. The reason could be the disconnections due to mobility and the constant changes in the network topology. Also larger files need longer times for seamless connectivity in any transmission therefore the high percentage of connection drops. This also has a detrimental effect on the percentage of stored and forwarded documents in a nodes repository. Another factor is the limited size of the repository that requires frequent updates of finding and removing older files.

In the experiments with content type and percentage of documents forwarded in a user’s repository, it was noted that smaller document sizes have a high rate of being carried and forwarded. On the average 43% of documents found in a users repository were forwarded documents with document size of 32KB. As the document size increases the number of forwarded documents decrease, for document of size 1024KB the average number of forwarded documents is 8%. More than 93% of documents found in a host’s repository are related documents that correspond to the similarity of a users interest profiles. This shows the effectiveness of the delivery of correct content type to the intended destination using the proposed ORP protocol.

Chapter 6

Dynamicity Aware Graph Re-Labelling Approach to Trust Management

6.1 Introduction

In human society, trust has become the basis of almost all activities, such as communications, work, etc. People gradually form the standard of mutual trust, and they also refer to opinions of the third-party in assessing the trust. Trust can be regarded as a criterion for making a judgment under complex social conditions and can be used to guide further actions [LEWI85]. It is no surprise that some researches related to security or mutual cooperation paid particular attention to trust factor in various approaches [GUHA04], [BUCH02] and [DAVI06].

Popular P2P content sharing applications such as mobile social networking in mobile environment provide various challenges for researchers. Traditionally social networks have been implemented in a client / server environment. In mobile social networks, users socially interact with handheld mobile devices while on the move, membership in a group / community in a MSN is granted by a pre-existing member of a group; revoking membership of a group is a challenging task without the existence of a central authority. Recent advances in semi de-centralized P2P social networks have been proposed [SERE07] [MERW07]. These techniques rely heavily on encryption protocols in client to server communication but provide no security between P2P interactions. Trust management in a de-centralized P2P network is a challenging task in the absence of a lack of global knowledge for all users; any trust / reputation parameters for a user have to be computed locally [HUYN06] [SERE07]. Given the existence of trust models for distributed systems, there is a need of a framework for trust management in user driven content sharing applications. The goal of the work presented in this chapter is to identify trustworthy users and allow secure transmissions while isolating untrustworthy users from the community thus creating trust based communities.

This chapter presents a trust based framework to membership management in a mobile social network. Dynamicity Aware Graph Relabeling System (DA-GRS) presented in

[CAST05] [CAST06] is used to label nodes in the network with a trust level indicator. These trust labels are used to compute individual level trust ratings as well as community/group level trust ratings. A group of users utilize these trust-level indicators to communicate with new users and invite them to become members. The goal is to create communities/groups of users with high trust ratings while identifying untrustworthy users and isolating them from the community thus revoking their membership. Results show that this method of community based trust management is more effective in reducing the amount of computations required at a local level in a distributed environment. Algorithms based on greedy concept using the DA-GRS system are presented. Two cost functions to measure the trust-ability of a group of users in a network are also presented. Simulation results show that trust based greedy algorithms create a much better quality of trusted groups compared to the standard DA-GRS algorithm.

Section 6.2 details the dynamicity aware graph relabeling system. Section 6.3 details the trust requirement for membership management in mobile social networks as a case of disconnected MANETs. Algorithms based on greedy method for graph labelling are presented in section 6.4 followed by simulation and results discussion in section 6.5. Section 6.6 summarizes the chapter.

6.2 Dynamicity Aware Graph Relabeling System

The Dynamicity Aware Graph Labelling System (DA-GRS) presented in, [CAST06] is an extension of the Graph Relabeling System. DA-GRS is a model invented for the conception and the analysis of decentralized applications and algorithms targeting dynamically distributed environments like disconnected MANETs. Normally, such applications and algorithms are often very difficult to set up, describe and validate. Using DA-GRS is a convenient way to design algorithms for disconnected MANETs, since its outstanding properties are localized in a dynamic working manner. In the context of this study, DA-GRS approach allows a way of designing a decentralized algorithm for constructing and maintaining a spanning forest in disconnected MANETs, relying on a careful rule-based token management [PIYA08].

6.2.1 Defining the Network

The network is considered essentially as an undirected graph where edges connect nodes. G = (V,E), with V being the set of vertices representing the mobile units (or nodes) and E being the set of edges such that: $\forall x, y \in V, (x, y) \in E \Leftrightarrow$ x and y can communicate directly. The dynamicity of the network is represented by the fact that V and E can change anytime with the following meaning:

- A vertex v is added to (respectively deleted from) V if the corresponding mobile unit is turned *on* (respectively *off*). Note that the deletion of v is equivalent, from a communication point of view, to the deletion of all the edges incident to v in one step.
- An edge *e* = (*v1*, *v2*) is added to E if and only if vertices *v1* and *v2* are in communication range provided that $e \notin E$. Symmetrically an edge *e* is deleted from E if and only if *v1* and *v2* can no longer communicate.

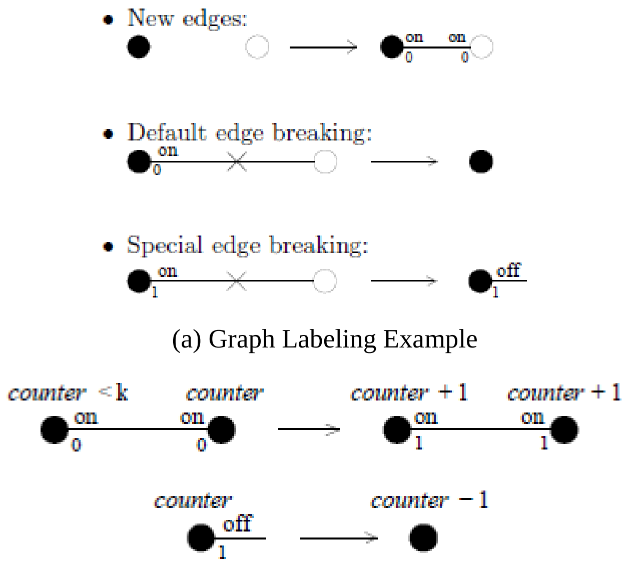
6.2.2 Labelling Vertices

The state of nodes and communication links are coded by means of vertex and edge labels. Each vertex has a state label for itself and another state label for each of its incident edges. An edge thus has a label on each side, which permits to code a non-symmetrical state. When an edge is added to the graph, it has an initial default label (noted 0). When an edge is deleted, its endpoint nodes add a special label to code the fact that the communication link has broken. This special label, noted *off* will allow applying some special operation to handle the deletion of the edge; thereafter, the edge is definitely and locally deleted. An illustration for the labelling mechanism is given in Figure 6.1 (a).

Figure 6.1(b) shows an example for adding and removing a node from the graph. Assuming that a node can connect to k neighbours, when a new node is encountered it is added by incrementing the value of counter (number of connections) provided the (counter + 1) < *k*. Similarly should a node fail to communicate in a given time frame, it is removed by simply decreasing the value of the counter.

6.2.3 DA-GRS Algorithm

The DA-GRS algorithm guarantees to maintain anytime a spanning forest that strives for a spanning tree, using only one-hop context information (*i.e.* it is a purely localized algorithm). Initially, each node is labelled J, i.e. I = {J}. The algorithm is composed of four rules, i.e. R = {*r1, r2, r3, r4*}. The algorithm is based on three operations on a token: circulation, merging and regeneration. Initially, each node has a token (and is labelled J), meaning that each node is a spanning tree in itself, containing exactly one node (itself), and being its own root. When two nodes labelled J meet each other, applying rule *r3*, the two spanning trees merge. Indeed, the labels 1 and 2 on an edge mean that it is part of the spanning tree. The use of two different labels allows a node to know the local route to the token. When rule *r3* applies, one of the two tokens is deleted and one of the nodes is relabelled N, that guarantees that there is at most one token per tree. The rule *r4* codes the circulation of the token in a tree of the forest. Note that the edge labels are switched to ensure that the local route to the token remains consistent. When a communication link is broken, i.e. when an edge is deleted, the node that is on the token side has nothing to do regarding the token maintenance, and simply applies rule *r2*. The node that had the deleted edge label to 1 has lost the route to the token, and is the only one of its remaining piece of tree to know that. It then regenerates a new token thanks to rule *r1*. Figure 6.2 shows the four rules for the DA-GRS algorithm.

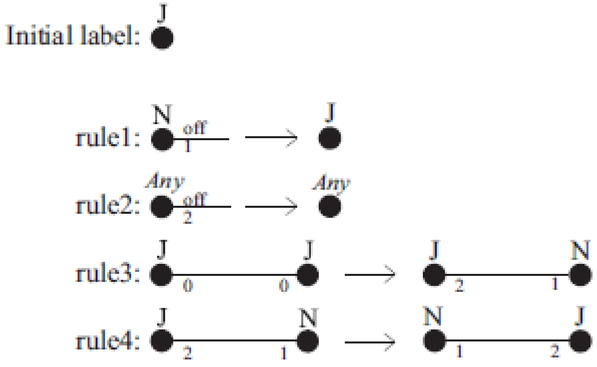


(a) Graph Labeling Example

(b) Adding and removing nodes from a graph

Figure 6-1: Graph labeling [CAST05]

The DA-GRS algorithm effectively handles four different scenarios, (a) tokens traversal in general case, (b) when a token meets another token, (c) partition occurs at a node which belongs to the spanning tree that possess the token and (d) partition occurs at a node which belongs to the spanning tree which does not possess the token.



```
label,J // initial state
// R1
v1.edgestate = off & v1.edgelabel = 1
v1.label = J & v1.edgelabel = 0
// R2
v1.edgestate = off & v1.edgelabel = 2
v1.edgelabel = 0 // allows the edge
// to be locally deleted
// R3
v1.label = J & v2.label = J
v1.edgelabel = 2 & v2.edgelabel = 1 & v2.label = N
// R4
v1.label = J & v2.label = N & v1.edgelabel ! 0
v1.label = N & v2.label = J & v1.edgelabel = 1 & v2.edgelabel = 2
```

Figure 6-2: Four rules for the DA-GRS algorithm [CAST06]

6.3 Membership criterion in MSN

Most of the online social networking services rely on a challenge / response authentication based on centralized certification authorities for membership [BEAC08] [CAUS09] [CHEN08] [LUGA07]. Membership in a P2P Mobile social network must rely on a decentralized reputation based configuration where nodes participate in labelling other nodes with a trust level [RAEN05] [ZIVN06]. Trust management within a partition of a DTN is very difficult because of its dynamicity, decentralized nature and non-permanent connection that can break up into two or more partitions at any moment. Although cooperative working manner among nodes / users within a DTN can be assumed, any trust management algorithm has to work at local level as global knowledge of the network cannot be acquired.

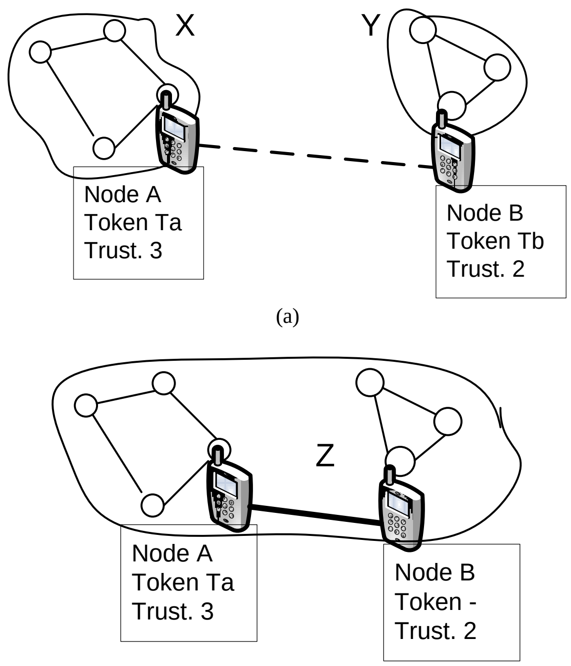
6.3.1 Trust Requirements

It is assumed that each node in the network is assigned with a unique identification, a token for labelling and a trust level indicator. The token is an essential part of the DA-GRS labelling system and is primarily used to randomly merge a node into a group. In this work the trust requirements are considered to be a combination of human social trust factors and the quality of service in a disconnected MANET.

A. Social Trust and reputation: Trust is one of the most crucial concepts for decision in making relationships in human societies. Trust is indispensible when considering interaction among users in online societies such as e-commerce, e-government etc. Many trust based schemes have been presented in the literature, however for de-centralized applications or networks, trust is defined to be based on a history of a user's encounters with other users [MUIL02] [SABA05] [HANG08]. Reputation based systems however compute trust based on recommendations from other users of the system [MUIL02] [YUB00]. In this chapter the concept of computing trust for an individual user as well as a group of users based on reputation is addressed. Section 6.3.2 shows detailed method for computing the trust values for both individual users and user as a part of a group.

B. Trust as a quality of service metric in MANETs: Trust level is also defined for a particular node to be a measure of its quality of service. It is based on criterion such as low battery, node being out of range, poor communication signal, etc. The trust level of a user is decreased if the user's device encounters one of the above problems. Users with a higher trust level have the luxury to stay connected for the longer periods of time and communicate with a large number of users. Such users are able to store and forward data from adjacent nodes while serving as an intermediate router. Nodes with lower trust level should not be permitted to store and forward data from other users due to the higher probability of a failed delivery, therefore must be isolated from the group.

C. Gaining membership: DA-GRS algorithm is utilized to discover and merge a node with others. Assuming users A and B have discovered each other and are willing to communicate. User A is already a member of a group *X*, where as B seeks membership of this group through A as shown in Figure 6.3(a). In this case user B can merge with the group *X* if the tokens of A and B, i.e. T_A and T_B can merge. If B was a part of a trusted group *Y*, then *X* and *Y* can merge into a larger group *Z* such as in Figure 6.3(b).



(a)

(b)

Figure 6-3: Nodes A in group X and B in group Y merge in to group Z

D. Trust labelling: A nodes trust level can be assigned in a cooperative manner by the trusted adjacent nodes based on a threshold. This threshold is determined by a set of factors such as running out of battery, node being out of range, poor communication signal or at user's discretion. Nodes with higher trust levels can connect to a larger set of nodes and share information where as nodes with lower trust levels are isolated. Trust for a group of nodes is computed using two cost functions, *group_cost* function and *isolation_cost* function as detailed in section 6.3.2.

E. Membership revocation: If a node's trust level falls to 0, consequently it is detached from the group and the membership of that user is effectively revoked. All members of the group remove the untrustworthy user from their respective list of trusted users.

6.3.2 Trust Computation

Trust level of a node is defined to take values from 0 (lowest) to 3(highest). Typically a node with a trust level 3 can be connected to a large number of nodes (higher degree) and have a low possibility of disconnection (high threshold) and therefore is more likely to complete its task. Alternatively a node with low trust level such as 1 is considered to be an isolated node and must therefore be marginalized. Table 6.1 shows a comparison of various trust levels.

Table 6-1: Definition of trust levels in nodes of the network

Trust Level	Degree	Threshold	Example
3	High	High	Trustable store & forward intermediate node
2	Low	High	Trustable intermediate node
1	High	Low	Isolated node
0	Low	Low	Nodes membership is to be revoked

A. Computing Trust for users: Recommendations from other users who have recently been in contact with the intended user are used to define trust for a user. Each user maintains a list of users with which they had a direct interaction. Every user has an opinion about another user and labels it as trustworthy, unknown or untrustworthy, taking the values +1, 0 and -1 respectively. Typically a user may trust another user or distrust him; a new user having no previous encounters with a trusted user is labelled as unknown, i.e. 0. Trust of a user is computed by the following equation

$$T(x) = \frac{\sum_{i \in t\_list} \left(Trust\ (i) * opinion\ _i(x)\right)}{\sum_{i \in t\_list} Trust\ (i)} \tag{6.1}$$

Whereas *x* is the node whose trust is to be computed; *i* is a node in the list of trusted users (*t_list*) and the function *opinion$_i$(x)* indicates the opinion of user *i* towards user *x*. Value for *T(x)* is always in the interval (1, -1), i.e. a Trust worthy user will obtain a positive value, whereas a negative value indicates a untrustworthy node. *Trust(x)* labels the node *x* with a trust value based on the value of *T(x)* given by

$$Trust(x) = \begin{cases} 3 & 1 \geq T(x) \geq 0.5 \\ 2 & 0.5 > T(x) \geq 0 \\ 1 & 0 > T(x) \geq -0.5 \\ 0 & -0.5 > T(x) \geq -1 \end{cases} \tag{6.2}$$

B. Computing Trust for a group of users: Trust level for a group is computed by two cost functions *group_cost* and *isolation_cost*. The trust level for the whole group indicates the quality of the trusted group therefore a higher value indicates a desirable trusted group. Values for these cost functions are computed to compare with the trust values of groups in various environment settings.

- ***Group_cost* function:** This function computes the cost of trust for the group. The cost of group G is determined by two factors, degree of trusted connections and trust level for each node in G. It is given by

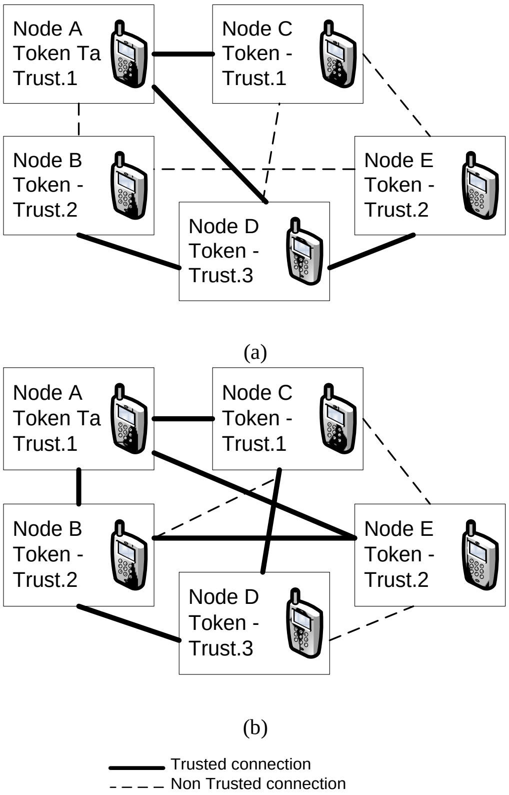
$$Group\_cost\ (G) = \sum\ (trust(x) * t\_conn(x)) \qquad (6.3)$$

  Where *t_conn* for a node x is the number of trustable connections to other nodes and trust(x) indicates the trust level of node *x*. As an example the *Group_cost* for the group shown in Figure 6.4(a) is 16. Similarly for the group in Figure 6.4(b) the group cost computed is 21. This shows that the group of users in Figure 6.4(b) has a higher trust ability compared to group in Figure 6.4(a). Having connections with nodes that have a higher trust level is desirable for long term communication. Node D in Figure 6.4(a) has a trust level of 3 and has 3 active trustable connections therefore is more trust able than node A in Figure 6.4(b) having a trust level of 1 and 3 active connections. Implicitly, it means, the higher the value of *group_cost* function the better quality of group in terms of number of trustworthy nodes. To have an optimal trust-level in a group, nodes with lower trust levels should be isolated with minimum number of connections while higher trust level nodes should be allowed to establish more connections.

- ***Isolation_cost* function**: To create better quality trusted groups, nodes with low trust levels (trust level <=1) and low number of connections have to be identified and consequently isolated. The *group_cost* is computed for low trust nodes in the group and subtracted from the *group_cost* of that group. As an example the *isolation_cost* for group G in Figure 6.4(a) would be 13, where as in Figure 6.4(b) is 16. The *group_cost* function and *isolation_cost* functions are computed by the node possessing the token.

6.4 Algorithms for Trust management in MSN

Due to the decentralized nature of mobile social networking in a delay tolerant environment maintaining a trust management in groups of nodes at a global level is very difficult; instead a trust management algorithm must work at a local level. The proposed algorithms modify the dynamicity aware graph labelling system algorithm (DA-GRS) to build communities of trusted users in the network.



(a)

(b)

Figure 6-4: Examples of a group of users in a MSN

6.4.1 Modified Dynamicity Aware-Graph Relabeling System (DA-GRS)

Trust level of a group is computed whenever a user / node seek to communicate to another user in a group, i.e. the tokens of the two nodes willing to communicate are compared. If the trust levels and the *group_cost* and *isolation_cost* values are acceptable the merger is completed and a larger group is formed. As an example consider Figure 6.5. Node A in group *X* has a trust level 3 while the *group_cost* value being 27 and *isolation_cost* value being 21. Node B in group *Y* has a low level of trust while the *group_cost* is 15 and *isolation_cost* is 6. Node A has a higher trust level in a group *X* that has a higher group trust level as compared to node B in group *Y*. Also in group *Y*, the ratio of *group_cost* versus *isolation_cost* is 15 to 6 indicating a high percentage of nodes that have a low level of trust and are isolated in the group. The DA-GRS algorithm in this case would allow groups *X* and *Y* to merge. It must be noted that this algorithm does not consider trust of individual nodes or the group trust level while merging.

6.4.2 Greedy Labelling

The Greedy DA-GRS algorithm is an improvement of the DA-GRS algorithm by adding the greedy algorithm concept. The idea behind this concept is to merge with nodes having a higher trust level therefore resulting in a robust trusted group communication. In Figure

6.5, the greedy labelling algorithm would merge node B with A. Since node B with a trust level 2 would prefer to merge with node A with a higher trust level of 3 instead of node C with a trust level of 1. The greedy labelling algorithm improves the overall trust level in the newly merged group.

6.4.3 High Group Trust Labelling

The High Group Trust (HGT) labelling algorithm focuses on group level trust rather than merging node's trust level. A group with a higher level of *group_cost* value can be considered as a robust trusted group with a long duration of time to live, i.e. the group in terms of performance has the longest available connection time and thus is more reliable. As an example, in Figure 6.5, node B prefers to merge with group *X* with a group cost of 27 rather than group *Z* with a group cost of 10. Larger groups with higher group trust cost can be considered most reliable. This algorithm is essentially a greedy algorithm based on DA-GRS where *group_cost* of a group is considered for comparison instead of individual node trust level.

6.4.4 Optimal Group Trust Labelling

The Optimal Group Trust (OGT) labelling algorithm focuses on quality of group trust. A group with lowest percentage of isolated nodes is preferable to larger groups with a high percentage of isolated nodes. As an example in Figure 6.5, group *X* has a ratio of 21 to 27; group *Y* has a ratio of 6 to 15 and group *Z* has a ratio of 8:10, this indicates that group *Z* has the highest optimal trust value, i.e. least number of isolated nodes. This algorithm is also a greedy algorithm based on DA-GRS. It focuses on quality of trusted groups in terms of group trust coherence. Figure 6.6 shows the three proposed algorithms.

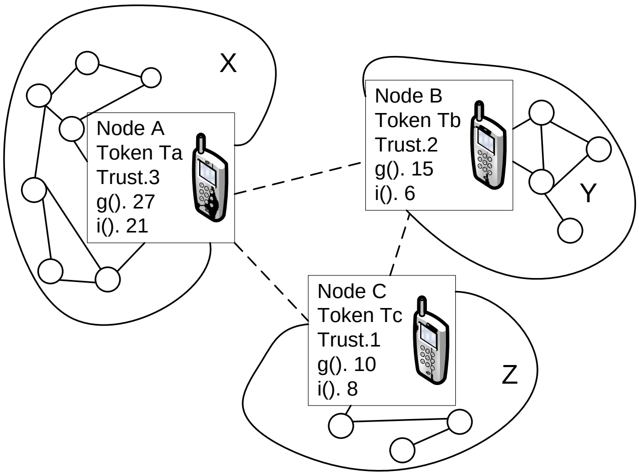


Figure 6-5: Merging of Groups.
Each node in a group has a token, a trust level, the *group_cost* g(*x*) and *isolation_cost* i(*x*).

```
void greedy(Tb){
  //Tb is the best trust value token in one hop neighborhood
  if (Tb != NULL)
    Merge_with_group(Tb, Tx);
  else Move_token(Tx);
}

void HGT(Tb,Gb){
  int g_cost;
  g_cost=compute_g_cost(x);
  if (Tb != NULL && g_cost < Gb)
     Merge_with_group(Tb, Tx);
  else
     Move_token(Tx);
 }

void OGT(Tb,Gb,Ib){
  int g_cost, i_cost;
  g_cost=compute_g_cost(x);
  i_cost=compute_i_cost(x);
  if(Tb != NULL && (g_cost - i_cost) < (Gb-Ib))
     Merge_with_group(Tb,Tx);
  else
     Move_token(Tx);
  }
```

Figure 6-6: Proposed algorithms for trust management

6.5 Simulation & Results

In a mobile social network it is assumed that every user is equipped with a mobile device. Each device has an Omni directional transmission range. Users are mobile and can communicate and stay connected while on the move. Simulation in this work considers three real-world environment categories. The categories are selected in terms of mobility and concentration of users. Users in the university campus and shopping mall networks are considered to be less mobile. Users in a city street are considered to be highly mobile. The networks used in this work are generated in the Madhoc simulator [HOGI].

To ensure validity of simulations three different networks are generated for each category of environment (9 networks in total). Table 6.2 shows the properties of each of these networks. Each network consists of 100 users. The total duration for each simulation was 20 seconds with 40 simulation steps taken at 0.5 seconds intervals. The simulation duration was selected carefully to reflect changes in networks that have higher mobility (street network). The initial trust values for each node is assigned following normal distribution with mean 0.25 and variance 0.1 for each set of values for *Trust(x),* i.e. 25% nodes receive trust ratings 0, 1, 2 or 3. It was noted that after 10 time steps a node in the simulation has an average degree of connections of 3.84 for city street networks. This indicates that most node were able to establish trust values for at least an average of ~4 nodes. For the other two types of networks, i.e. campus and shopping mall, the average degree of connectivity was slightly higher. Therefore based on this analysis the first 10 simulation steps are used as a trust ratings learning period, during which a node acquires trust ratings for members of the groups.

It can be seen that the changes in the city street network are more frequent than in campus or shopping mall networks. Figure 6.7 shows an example of each of the three types of networks. As stated before determining an optimal spanning tree for a decentralized dynamic network is extremely difficult. However since networks used in this study were generated using Madhoc simulator, the configuration of a network can be pre-determined. Therefore the robustness of suggested algorithms can be evaluated by calculating the *group_cost* function and the *isolation_cost* function of each of these networks. The experiments carried out simulation 400 times for each network.

A. Results for Campus Networks

Table 6.3 shows results for the average values of group and isolation cost functions for the suggested algorithms. The campus network is chosen due to its low mobility and high connectivity feature. From the results it can be seen that greedy labelling algorithm yields the highest group cost thus resulting in most number of trustable groups. The *isolation_cost* for High Group Trust (HGT) algorithm is higher than greedy algorithm therefore resulting in forming better quality groups. It must be noted that the group cost for Optimal Group Trust (OGT) algorithm is lower than both greedy and HGT algorithms but it provides the best *isolation_cost* thus creating the best quality trusted groups.

B. Results for Shopping Mall Networks

Results for the averages of group and isolation cost functions for shopping mall networks can be seen in Table 6.4. The shopping mall networks have slightly higher degree of mobility compared to campus networks. Due to higher mobility the average numbers of connections are lower. It can be seen from the results that greedy labelling algorithm performs better compared to HGT and OGT algorithms in creating trustable groups. The ratio of *group_cost* and *isolation_cost* indicates that OGT performs better in terms of creating high quality trusted groups. It can also be seen that the *group_cost* function for HGT yields almost similar values for OGT.

Table 6-2: Properties of three sets of each category of networks (campus, shopping mall and city street). Total number of users in each network is 100.

	Campus1	Campus2	Campus3
Max no. of connections	20	40	60
Min no. of connections	0	0	0
Avg. no. of connections	5.8	19.1	33.2
Total no. of connections	708	1045	1389

	Mall 1	Mall 2	Mall 3
Max no. of connections	20	40	60
Min no. of connections	1	1	1
Avg. no. of connections	4.2	17.3	28.6
Total no. of connections	688	943	1073

	Street 1	Street 2	Street 3
Max no. of connections	50	70	90
Min no. of connections	2	2	2
Avg. no. of connections	9.2	11.6	12.8
Total no. of connections	322	379	437

C. Results for City Street Networks

Results for the averages of group and isolation cost functions can be found in Table 6.5. Users moving in a city street are considered to be highly mobile compared to the earlier defined networks. Results show that the dynamicity of the network yields fewer trusted connections therefore the average cost functions values are lower compared to campus and mall networks. An interesting fact observed in simulation indicates that due to higher mobility the group cost for OGT is not similar to HGT. A possible reason could be decrease in performance due to the cost of computing the ratios. Apart from this issue, OGT still performs better in terms of creating better quality trusted groups.

Table 6-3: Averages of group and isolation cost functions for campus networks

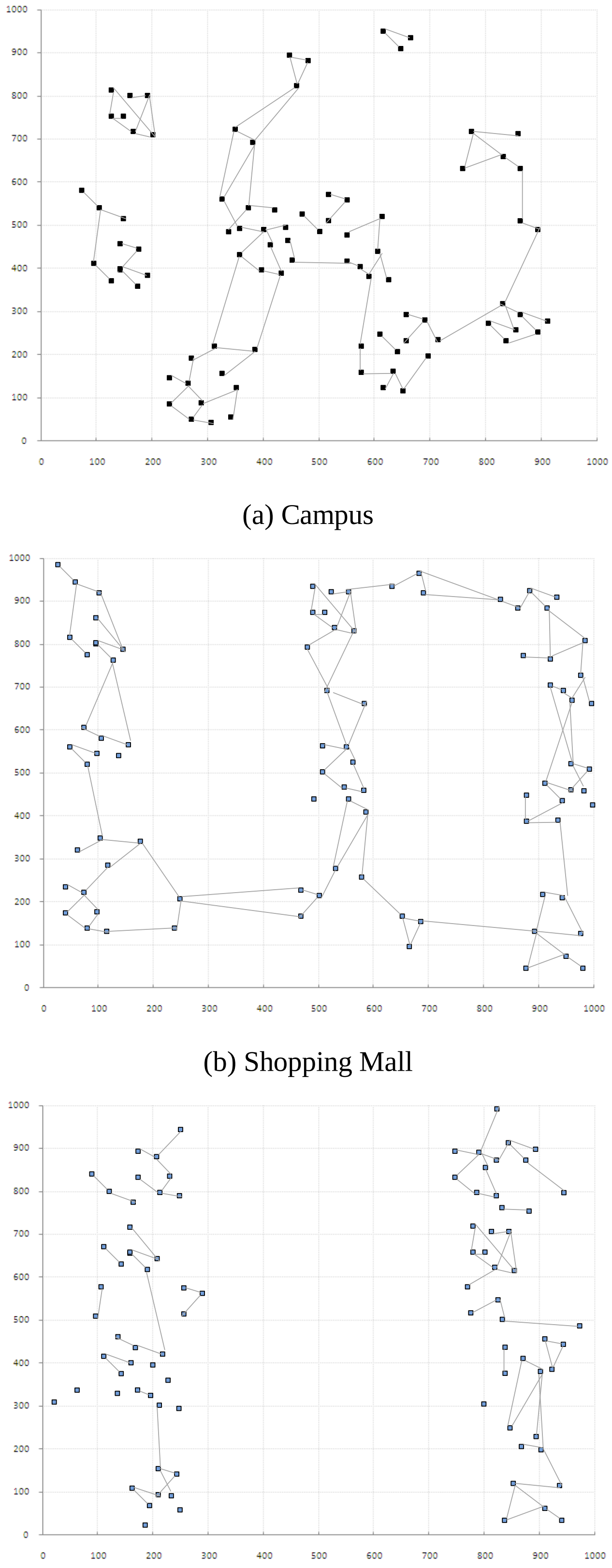
	Campus1		Campus2		Campus3	
	Group_ cost	Isolation_ cost	Group_ cost	Isolation_ cost	Group_ cost	Isolation_ cost
DA-GRS	559.2	455.3	881.7	718.2	1165.1	927.6
Greedy labeling	683.3	581.4	991.4	871.3	1359.4	1198.7
High Group Trust (HGT)	635.6	588.1	915.3	877.9	1298.7	1207.3
Optimal Group Trust (OGT)	621.0	603.9	908.6	896.8	1269.3	1216.9

Table 6-4: Averages of group and isolation cost functions for Shopping Mall networks

	Shopping Mall1		Shopping Mall2		Shopping Mall3	
	Group_ cost	Isolation_ cost	Group_ cost	Isolation_ cost	Group_ cost	Isolation_ cost
DA-GRS	433.8	327.4	630.1	498.7	851.8	608.6
Greedy labeling	592.5	497.7	889.0	770.9	1024.9	878.6
High Group Trust (HGT)	549.0	511.3	861.7	768.1	989.2	881.9
Optimal Group Trust (OGT)	544.9	529.2	812.5	782.1	965.0	912.8

Table 6-5: Averages of group and isolation cost functions for City Street networks

	City Street		City Street		City Street	
	Group_ cost	Isolation_ cost	Group_ cost	Isolation_ cost	Group_ cost	Isolation_ cost
DA-GRS	315.8	241.6	491.3	327.0	701.5	489.5
Greedy labeling	483.2	311.7	634.8	505.7	794.1	650.1
High Group Trust (HGT)	422.5	351.9	591.6	501.3	779.2	661.8
Optimal Group Trust (OGT)	404.8	378.1	578.9	538.1	744.8	688.0



(a) Campus

(b) Shopping Mall

(c) City Street

Figure 6-7: Screen-shots of Networks used from the MADHOC simulator
(a) Campus Network, (b) Shopping Mall and (c) City Street

6.6 Discussion and summary

Trust management in dynamic decentralized mobile networks is receiving attention due to its immense application. This chapter presents algorithms for decentralized trust management in Mobile social networks based on a dynamicity aware graph relabeling system. The proposed algorithms are based on greedy concept and the results affirm the benefits of using this approach. Although simulating human behaviour for trust and reputation assignment is unpredictable, a method was presented to compute trust of users based on a reputation model where users recommend their opinion about other users. Two cost functions to measure the trust-ability of a group of users in a network were also presented.

The results show that trust based greedy algorithms create a much better quality of trusted groups compared to the standard DA-GRS algorithm. Extensive simulations also show the quality of proposed algorithms when tested in scenarios such as campus, shopping mall and city-street. The greedy algorithm, High group trust (HGT) and Optimal group trust (OGT) all outperform the DA-GRS algorithm. It must be noted that the greedy algorithm is best in terms of creating useful groups; however its weakness is in isolating low trust nodes. It can be seen that the performance in terms of number of isolated nodes for HGT and OGT is higher compared to the greedy algorithm, although the greedy algorithm is less computation oriented therefore is faster and makes larger groups. The HGT and OGT both outperform greedy algorithm in terms of making quality trusted groups. It was also noted that the effect of mobility plays a great part in the performance of the suggested algorithms. The values of the *group_cost* function and *isolation_cost* function, both decrease due to higher mobility; however the suggested algorithms maintain the ability of managing highly trusted groups even in high mobility networks. Further results can be found in appendix B.

As stated before, determining optimal group trust values for a decentralized dynamic network is extremely difficult. However since networks used in this study were generated using Madhoc simulator, the configuration of a network was pre-determined. It must be noted that although the random waypoint mobility model was used to determine the mobility of users in the network, determining the user mobility in real time environments is unpredictable and is an ongoing research area.

Chapter 7

FIRE+ Model for Collusion-free Trust Management in disconnected MANETs

7.1 Introduction

Trust is one of the most crucial concepts driving decision making and establishing relationships. Trust is indispensible when considering interactions among individuals in artificial societies such as electronic commerce [YUB00]. As an important concept in network security, trust is interpreted as a set of relations among nodes participating in the network activities [RAMC04] [LIMC09]. Trusted relationships among nodes in a network are based on different sources of information such as direct interactions, witness information and previous behaviours of nodes. Trust management in distributed and resource-constraint networks, such as disconnected MANETs and sensor networks, is much more difficult but more crucial than in traditional hierarchical architectures, such as the Internet and access point centred wireless LANs. Generally, this type of distributed network has neither pre-established infrastructure, nor centralized control servers or trusted third parties. Unlike traditional networks, where packets are forwarded along fixed links, disconnected MANETs allow packet forwarding along intermittent links. Consequently, traditional stable-link-based routing and packet forwarding protocols are not applicable to disconnected MANETs, since a contemporaneous end-to-end path may never exist. Therefore, nodes use an underlying store-and-forward model of routing to cope with unstable paths, usually caused by high mobility and a low density of nodes. The dynamically changing topology and intermittent connectivity of disconnected MANETs establish trust management more as a dynamic systems problem [BARA05]. Furthermore, resources (power, bandwidth, computation etc.) are limited because of the wireless and ad hoc environment, so the trust evaluation procedure should only rely on local information.

Reference [SABA01] categorized computational trust and reputation models based on various intrinsic features. Trust and reputation models vary in terms of individual behaviour assumptions; in some models, cheating behaviours and malicious individuals are not considered at all whereas in others possible cheating behaviours are taken into account. Trust and reputation models might use different sources of information such as direct experiences, witness information, sociological information and prejudice [LIJ08] [LIMC08]. Direct experience and witness information are pertinent to this chapter. Direct experiences deal with node-to-node interactions while witness information is information that comes from members of the community about others.

In a witness-based collusion attack, an unreliable witness provider, in spite of being cooperative in its direct interactions provides high ratings for other malicious nodes (other members of the colluding group), thus resulting in motivating the victim node to interact with them [KERR09]. This lack of study on witness-based collusion attacks motivates the work reported in this chapter. FIRE+, an extended version of FIRE trust and reputation model [HUYN06], for decentralized distributed networks such as disconnected MANETs is proposed. Contributions in this work address the vulnerability of FIRE model to collusion attack from a group of malicious nodes. The proposed FIRE+ multidimensional model is based on direct and witness trust interaction for detecting collusion attack. FIRE+ defines a mechanism for periodically detecting the confidence in direct and witness information received from recommending nodes and storing it in a rating history database for identifying collaborative behaviour in recommendations. Based on this information trust aware nodes can use policies to reduce the level of encountered risk of an attack.

7.2 FIRE trust and reputation model

FIRE [HUYN04][HUYN06] presents a modular approach that integrates up to four types of trust and reputation from different information sources, according to availability: interaction trust, role-based trust, witness reputation, and certified reputation. FIRE model classifies users in a network as *Agents*, a set of users participating in trust interaction; *Targets*, users whose trust and reputation is being sought in an interaction and *Evaluators*, users requesting trust information about a target. Since FIRE defines a node in a network as an agent, nodes and agents are therefore used interchangeably in this chapter. Each time agent *i* gives a rating, it will be stored in the agent's local rating database. Ratings in this database will be retrieved when needed for trust evaluation or for sharing with other agents. However, an agent does not need to store all ratings it makes. Old ratings become out of date due to changes in the environment and may not be stored in limited amount of memory. Each agent will store a maximum number of ratings given the permissible size of the database. In FIRE, trust rating is calculated based on recommendations from direct interaction, witness interaction or rule based interaction.

A. **A Direct interaction trust:** The evaluator uses its previous experiences in interacting with the target agent to determine its trustworthiness. This type of trust is most frequently used [WANG08] [SRIV06] and is called Direct Interaction Trust (DIT).

B. **Witness interaction trust:** Assuming that agents are willing to share their direct experiences, the evaluator can collect experiences of other agents that interacted with the target agent. Such information will be used to derive the trustworthiness of the target agent based on the views of its witnesses. Hence this type of trust is called Witness Interaction Trust (WIT).

C. **Role-based rules**: Besides an agent's past behaviours (which is used in the two previous types of trust), there are certain types of information that can be used to deduce trust. These can be the various relationships between the evaluator and the target agent or its knowledge about its domain (e.g. norms, or the legal system in effect). For example, an agent may be preset to trust any other agent that is owned, or certified, by its owner; it may trust that any authorized dealer will sell products complying to their company's standards; or it may trust another agent if it is a member of a trustworthy group.[4] Such settings or beliefs (which are mostly domain-specific) can be captured by rules based on the roles of the evaluator and the target agent to assign a predetermined trustworthiness to the target agent. Hence this type of trust is called Role-based Trust.

D. **Third-party references provided by the target agents:** In the previous cases, the evaluator needs to collect the required information itself. However, the target agent can also actively seek the trust of the evaluator by presenting arguments about its trustworthiness. However, in contrast to witness information which needs to be collected by the evaluator, the target agent stores and provides such certified references on request to gain the trust of the evaluator. Those references can be obtained by the target agent (assuming the cooperation of its partners) from only a few interactions, thus, they are usually readily available. This type of trust is called Certified Reputation.

7.2.1 Trust Formula

In order to calculate the trust value (rating) of a target agent, its rating for past encounters with its neighbours need to be collected. In [HUYN06], researchers describe a way to estimate that value by calculating it as the arithmetic mean of all the rating values in the set of witness ratings form the neighbours.

$$T(a,b) = \frac{\sum_{ri\ \in R(a,b)} \omega(ri) * vi}{\sum_{ri\ \in R\,(a,b)} \omega(ri)} \qquad (7.1)$$

Where *T(a, b)* is the trust value that agent a has in agent b. *R(a, b)* is the set of witness ratings collected by agent a for agent b. $\omega(ri)$ is the rating weight function that calculates the reliability of the rating r_i ($\omega(ri) \geq 0$); and v_i is the value of rating r_i. In short, the trust value is calculated as the sum of all the available ratings and normalized to the range of ［−1， 1］ (by dividing the sum by the sum of all the weights).

7.2.2 Direct & Witness Interactions

FIRE assumes the direct and witness reputation of a target agent is built on observations about its behaviour in interaction with other agent’s. In order to evaluate the reputation of an agent *b*, agent *a* needs to find the agents that have interacted with *b* in the past. Here, it is assumed that agents in a network are willing to share ratings that they made and to help others search witnesses.

The system assumes that each agent has a measure of the degree of likeliness with which an agent can fulfil an information query about witness information and witness locating. It is assumed that an agent may know local agents (those that are near to it) better and, therefore, the distance between an acquaintance and the target agent is used as a knowledge measure. Thus it can be said that the nearer to the target agent, the more likely the acquaintance is to know it. This measure is used in the referral process to help locate witnesses. It is also assumed that the farther a agent is from the target the chances of knowing each other are lesser and therefore less reputable.

The process of acquiring trust witnesses for a target agent is shown in Figure 7.1. Four steps are followed to acquire witnesses from the neighbourhood.

1. Agent *a* (evaluator) asks for reputation ratings from direct neighbours (acquaintances) for agent *b* (target).
2. The Direct neighbour who received the request finds its own reputation for *b*. If found, it forwards the reputation rating to *a*.
3. If not found, it forwards the "referral" (information about direct neighbours of the neighbour) to *a*.
4. The process repeats until *a*, has acquired sufficient number of witnesses.

It should be noted here that in this process [HUNY06] implicitly assume that agents in *a*'s referral network are willing to help *a* finding the required witness ratings. The set of ratings collected from the referral process, denoted by R_W(a, b, c), is then used to calculate the witness rating of agent *b* using the Trust formula given in (7.1).

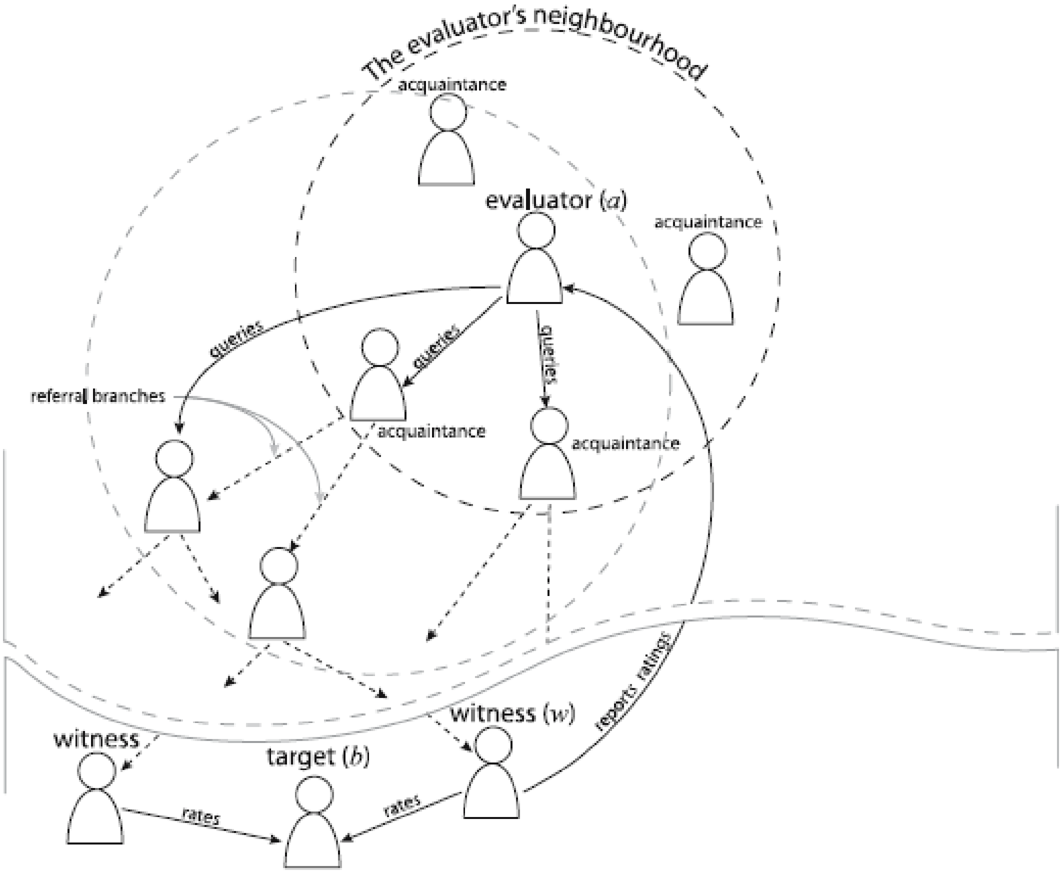


Figure 7-1: Witness referral process [HUYN06]

7.2.3 Modes of Trust and Reputation

In a witness based trust and reputation system, an evaluator can have three modes of acquiring trust information about a target from various witnesses. These modes are trust from direct interaction, reputation from direct witnesses and reputation from indirect witnesses. In all three cases it is assumed that an agent requesting trust values is the *evaluator* of a potential *target* agent. In case no direct interaction is possible an intermediate agent also known as *referrer* is going to be a *witness* for a target agent and provide a trust value. As an example in Figure 7.2, A evaluates target agent E, where agents B and C are witnesses for trust value of E; that is to be referred to agent A.

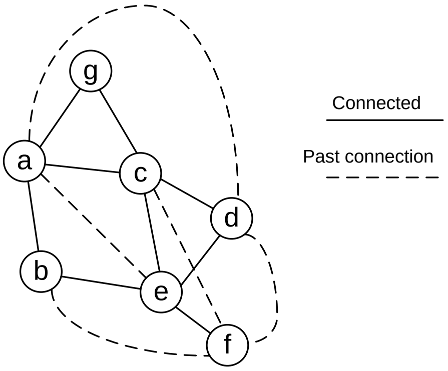


Figure 7-2: Example of direct trust and witness based trust

Trust from Direct interaction An evaluator assesses another agent's direct trustworthiness from its history of past interactions. For instance, Figure 7.2 shows an example of trust in direct interaction among agents. Agent A (*evaluator*) is interacting with agents B and C. A maintains a list of all encounters with these two agents and records trustworthiness of the subject agents based on service characteristics such as successful delivery, timeliness and cost. Assuming that A requires a resource R which both B and C posses, from its interaction history, agent A can determine that B has in the past completed 90% of the transactions compared with C that has completed only 50%. From this comparison agent A can choose to accept resource R from agent B.

Reputation from direct witnesses In Figure 7.2, agent A has interacted with agent E in the past and maintains a list of reputation including trust recommendation for E. However due to changing topology A can no longer maintain a direct communication (single hop) with E. As figure shows, B and C are direct neighbours of agent E and can recommend E based on their past direct encounters. A can either rely on the reputation from direct interaction which may not be recent or can request updated reputation value for E from its neighbours B and C. In this case B and C can forward recommendation for agent E to the evaluator A. These recommendations from B and C are witness trust values for agent A. Agent A would consider the three recommendations values and compute a new trust value for agent E.

Reputation from indirect witnesses: Considering the case where agent A needs to interact with agent F, A needs to find witnesses to recommend F. As shown in Figure 7.2, direct neighbours of A which are B, C and G do not posses trust value from direct recommendations, therefore would have to rely on recommendations from extended neighbours. Agent F has a trusted relationship with agent E; therefore agent E can be a witness and forward the trust recommendation for F to agents B and C. This "referral" of recommendations from agent E, for target agent F, is used to compute witness trust values for agents B and C, which can later be forwarded to agent A as the original evaluator.

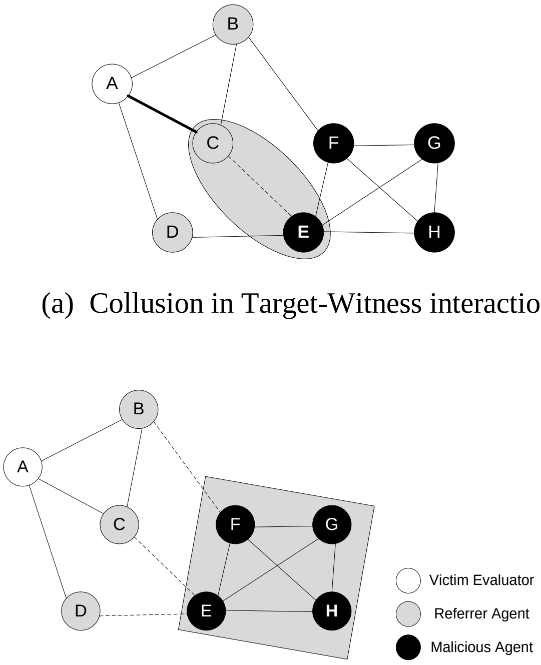
7.3 The Collusion Attack in disconnected MANETs

Researchers in [LIJ08] [LIAN07], have identified the existence of cheaters (exploitation) in artificial societies employing trust and reputation models and the existence of inaccurate witnesses. This inaccurate information can challenge the integrity of the reputation system based on witness information leading to misleading trust information and possibility of collusive behaviour to promote or sideline a user or group of users. Collusion can be defined as "*a collaborative activity that gives to members of a colluding group benefits they would not be able to gain as individuals*" [SALE09].

Collusion attacks occur when one or more agents conspire together to take advantage of breaches in trust models to defraud one or more agents. It can be the case that agents in the colluding group adopt a sacrificial stance in collusion attacks in order to maximize the utility of the colluding group. Collusion attacks often work based on the basic idea that one or more agents show themselves as trustworthy agents in one type of interaction (usually direct interaction). Afterward, they will be untrustworthy in other type of interaction (e.g., witness interaction) by providing false information in favour of other members of the colluding group. This false information usually encourages a victim to interact with members of the colluding group. Then, the members of the colluding group will cheat the victim. This section shows two kinds of collusion attacks on witness based trust management.

7.3.1 Collusive behaviour in Target-Witness Interaction

This behaviour of colluding users applies to an agent requesting trust values for a target through a witness. Figure 7.3(a) shows an example of agent A (evaluator) requesting trust values for agent E (target). Only intermediate agents C and D have direct interaction with both evaluator A and the target E and therefore posses a trust value. Both C and D can pass on the trust recommendations for E, to the evaluator A. B can also provide a trust rating for E, but since it doesn't interact directly with E, it has to rely on witness recommendations from F, therefore a direct referral from C or D would be preferable. C can collude with malicious target E to provide false positive recommendations to the evaluator subsequently promoting target E as a trustable user.



(a) Collusion in Target-Witness interaction

(b)Collusion in witness-witness interaction

Figure 7-3: Collaborative behavior in providing false recommendation values from malicious nodes

7.3.2 Collusive behaviour in Witness-Witness Interaction

A group of malicious agents can collaborate to recommend false trust values for a member of group to gain access to resources. In case when an evaluator agent cannot find direct recommendations from immediate neighbours it relies on recommendations from witnesses. Figure 7.3(b) shows collusive behaviour among witnesses. The evaluator A, obtains recommendations for target H. As before A has no prior knowledge of trust values for H. B, C and D can all provide independent trust values to A, honestly, based on recommendations from agents E and F. It can be seen that both witness providing agents can collude to provide false values to promote H or to present H as an untrustworthy user. Figure 7.3(b) shows collusive behaviour among malicious agents collaborating to pass false information to B, C and D, thus affecting trust values for evaluator A.

In both aforementioned cases it can be observed that when the victim / evaluator agent bases its assessment of witness information on the co-operations (trustworthiness) in direct interactions, the collusion attack will be successful. In particular, the success of this attack is the result of the inappropriate assumption that whoever is cooperative (trustworthy) in direct interactions will be cooperative (trustworthy) in providing witness information regarding other agents. FIRE+ trust model hypothesize that the witness based collusion attack can be prevented if the evaluator agent can utilize a multi-dimensional trust model. In its essence the evaluator agent will assess the witness providers based on their cooperation in witness interactions.

7.4 FIRE+ Trust Model

This section presents a multidimensional trust model FIRE+, based on FIRE trust and reputation model to counter the threat posed by colluding groups of agents in a network. Trust and reputation variables are defined to determine to connect to an agent. Also a mechanism to store trust information in each agent for quick retrieval is defined. Furthermore a graph building mechanism to determine colluding and misbehaving agents with the help of trust policies for connection, interaction and referral are presented.

7.4.1 Trust Variables

Based on the FIRE trust and reputation model, trust variable $T_{i,j}(t)$ is defined to identify the level of trust an evaluator *i*, has for a target agent *j* after *t* interactions between agent *i* and agent *j*, while $T_{i,j}(t) \in [-1, +1]$ and $T_{i,j}(0) = 0$. One agent in the view of the other agent can have one of the following levels of trustworthiness: Trustworthy ($1 >= \lambda >= 0$), Not Yet Known ($\lambda >= 0 >= \mu$), or Untrustworthy ($0 >= \mu >= -1$), where λ and μ and upper and lower thresholds.

Direct Interaction Trust

Direct interaction trust (DIT) is the result of direct interaction with agents. Each evaluator agent n_i maintains a direct trust value DIT$_{i,j}$ for each target agent n_i. Based on the number of positive interactions α or negative interactions β the trust value for a target n_i is updated using the principle defined in [YUB00].

```
if $DIT_{i,j} > 0$ and $(\alpha > \beta)$ then
$DIT_{i,j} = DIT_{i,j} + (1 - DIT_{i,j})$
if $DIT_{i,j} > 0$ and $(\alpha < \beta)$ then
$DIT_{i,j} = DIT_{i,j} + ((\alpha - \beta)/\min(\alpha, \beta))(1 - DIT_{i,j})$
if $DIT_{i,j} < 0$ and $(\alpha > \beta)$ then
$DIT_{i,j} = DIT_{i,j} + ((\beta - \alpha)/\min(\alpha, \beta))(1 + DIT_{i,j})$
if $DIT_{i,j} < 0$ and $(\alpha < \beta)$ then
$DIT_{i,j} = DIT_{i,j} – (1 + DIT_{i,j})$
```

Where $(\alpha - \beta) > 0$ is positive evidence and $(\beta - \alpha) >0$ is negative evidence. The value of DIT$_{i,j}$ determines if n_i is trustworthy, untrustworthy or not yet known.

Witness Interaction Trust

Witness interaction trust (WIT) is the result of indirect interaction with agents. An evaluator agent n_i also maintains a list of witness interaction trust WIT$_{i,j}$ with a target agent n_j that has no direct interaction but are referred to by a witness agent(s) n_w. Updating scheme for WIT$_{i,j}$ is similar to direct interaction trust DIT$_{i,j}$ with the exception of positive evidence $(\alpha - \beta)$ and negative evidence $(\beta - \alpha) >0$ for witnesses referrals. The value of WIT$_{i,j}$ determines the level of trustworthiness for target n_i.

```
if $WIT_{i,j} > 0$ and $(\alpha > \beta)$ then
$WIT_{i,j} = WIT_{i,j} + (1 - WIT_{i,j})$
if $WIT_{i,j} > 0$ and $(\alpha < \beta)$ then
$WIT_{i,j} = WIT_{i,j} + ((\alpha - \beta)/\min(\alpha, \beta))(1 - WIT_{i,j})$
if $WIT_{i,j} < 0$ and $(\alpha > \beta)$ then
$WIT_{i,j} = WIT_{i,j} + ((\beta - \alpha)/\min(\alpha, \beta))(1 + WIT_{i,j})$
if $WIT_{i,j} < 0$ and $(\alpha < \beta)$ then
$WIT_{i,j} = WIT_{i,j} – (1 + WIT_{i,j})$
```

The value of WIT$_{i,j}$ determines if n_i is trustworthy, untrustworthy or not yet known.

7.4.2 Defining History and Reputation Variables

Let us consider representation of an evaluator agent's n_i history of trust values for other agents. Since trust values for other agents change, based on trustworthiness of number of interactions agent n_i maintains a partial history of interactions with other agents declared as $H_i = \{ n_i , n_j, \alpha, \beta, T_{ij}, ttl , \gamma , r\}$, where n_i is the evaluator agent, n_j is the target agent, α and β are the positive and negative number of interactions. T_{ij} is trust value of agent n_i for agent n_i in the range [+1, -1], *ttl* is the time stamp when the trust value is determined, γ is the confidence in trust value and *r* is a Boolean variable indicating if the recommendation is based on direct interaction or witness referral. In case of a witness referral from agent n_w, the referrer n_w is stored instead of evaluator n_i. Based on number of interactions, the confidence γ of n_i in a trust value for n_k shows the experience of interactions. Higher confidence predicts more positive interactions in the future. An evaluator may opt to consider recommendations from agents with higher confidence compared to low confidence agents. This recommendation confidence is utilized in the trust graph building to determine collaborating agents.

As the evaluator takes into consideration recommendations to decide about trustworthy agent selection, it updates its recommendation trust in the witnesses and also records the interaction results in its history. The interaction history gives a reflection of the relevant past transactions of an agent. Since each record in the history has a timestamp ttl value for each trust recommendation, older values can be discarded to reduce the size of the history database. To determine if a service performed in an interaction was to the desired expectation, the desired value of service to the actual value after the interaction is compared and the values of α and β are incremented accordingly.

In FIRE model, witness-based reputation for a specific agent is calculated based on the ratings of other agents. FIRE+ computes trust values using the same witness based trust formula defined in FIRE; however reputation for both direct interaction and witness interaction to discover possible collusive behaviour is calculated. Assuming agent n_i has no direct interaction with target agent n_k, it requests trust rating for target agent n_k from a referrer agent n_j. n_j provides the requested rating T_{jk} (n_j's trust rating for n_k). Given that many such ratings about target n_k can be obtained from potential referrers $R_{i,k}$ defines the reputation rating of n_k for evaluator agent n_i such that these ratings are available in the n_j's trust History database H_i. Value of φ(x) determines if agent n_k is trustworthy, untrustworthy or not yet known.

$$DR_{i,k} = \frac{\sum_{j \in Hi}(T_{jk}\ .\ \varphi(DIT_{ij}\ ))}{\sum_{j \in Hi}\varphi(DIT_{ij}\ )} \tag{7.2}$$

$$WR_{i,k} = \frac{\sum_{j \in Hi}(T_{jk}\ .\ \varphi(WIT_{ij}))}{\sum_{j \in Hi}\varphi(WIT_{ij}\ )} \tag{7.3}$$

where φ(x) is given by

$$\varphi(x) = \begin{cases} 0 & \mu > x \geq -1 \\ (x-\mu)/(\lambda-\mu) & \lambda \geq\ x \geq \mu \\ 1 & 1 \geq\ \lambda \end{cases} \tag{7.4}$$

As the evaluator takes into consideration recommendations to decide about target node selection, it updates its recommendation trust in the witnesses and also records the interaction results in its history. The interaction history gives a reflection of the relevant past transactions of a node. The evaluator may apply a decay function to the older interactions to give higher importance to the more recent ones although this phenomenon is not considered in the experiments.

7.4.3 Trusted agents network

As an evaluator interacts with direct agents and witnesses, it gathers information about interactions and relationships to build an agent network to better understand its environment. Three graph structures are considered to represent an agent's environment: direct agents graph, witness graph, and a combined direct-witness graph. The nodes represent agents and the edges correspond to links between agents, including the strength of the link in terms of experience (confidence γ). An evaluator constructs the combined direct-witness graph from its own direct interactions and inferred interactions between other agents from the recommendations it receives. Algorithm 7.1 shows how part of the agent graphs is constructed and updated where r_x is the currently updated and processed recommendation. For a direct recommendation an edge is created in the direct-graph for evaluator agent n_i and the positive interaction count is incremented. For each edge created in the direct-graph, edges are added to the witness-graph of n_i for every further agent that has a direct interaction in the chain of witnesses.

Figure 7.4 shows an example of three graphs constructed for nodes *a*, *b* and *c* respectively, based on combined information from direct interactions with neighbours and recommendations from witnesses. In trust graph for evaluator node a, there are two direct neighbours *b* and *c*, one indirect neighbour *d*, and two witness recommendations for nodes *e* and *f*. In trust graph for node b, there are two direct neighbours *a* and *d*, and three indirect neighbours *c* (recommended by *a*) and *e*, *f* (both recommended by *d*). It must be noted that *b* does not receive any indirect witness recommendations. For the trust graph of node *c*, there is one direct neighbour *a*, one indirect neighbour *b* and three witness recommendations *d*, *e* and *f*.

Algorithm 7.1 Direct and Witness graph updates

```
for all $r_x$ such that $r_x \in H_i$ do
   if $r_x$ is $DIT_{ix}$ then
      $a_i$.graphd.addedge($a_i$,$a_x$)
      increment $\alpha_{r_x}$
   for all $r_y$ such that $r_y \in H_i$ do
      if $r_y$ is $WIT_{iy}$ then
         $a_i$.graphw.addedge($a_i$,$a_y$)
      end if
   end if
for all $a_i$.graphd($r_x$) such that $r_x \notin H_i$ do
   for all $a_i$.graphw($r_y$)  such that $r_y \notin H_i$ do
         $a_i$.graphw.removeedge($a_i$,$a_y$)
   $a_i$.graphw.removeedge($a_i$,$a_x$)
```

7.4.4 Detecting Collusion Attack

As mentioned earlier, the evaluator agent continuously maintains it's direct and witness graphs throughout the period of interaction with other agents. The graphs contain a summary of the links between two agent nodes. For instance, the graph edges in direct graphs record the number of positive and negative interactions between the two agents and confidence of interaction. Meanwhile, the witness graph edges consist of the number of accurate and inaccurate recommendations by the witnesses, both for direct and indirect opinions.

Figure 7.5 shows an example of discovery of collusive behaviour in witness interaction recommendations. The evaluator A seeks recommendations for target E from direct interaction agents B, C and D. Since there is no past interaction with E, depicted by dotted line in Figure 7.5 (a), therefore agent A doesn't have a trust value for E. Direct-graph and witness graphs are constructed to discover recommendations for E that can be obtained from B, C and D as shown in Figure 7.5 (b). The extended and combined graph in Figure 7.5 (c) shows further interactions between nodes E, F, G and H. The values of DIT and WIT are computed for recommendations R for direct interaction agents and subsequent witness interactions. In due course the values of trust and reputation are updated and the confidence measure for each edge in the graphs is incremented or decremented based on the number of positive or negative interactions.

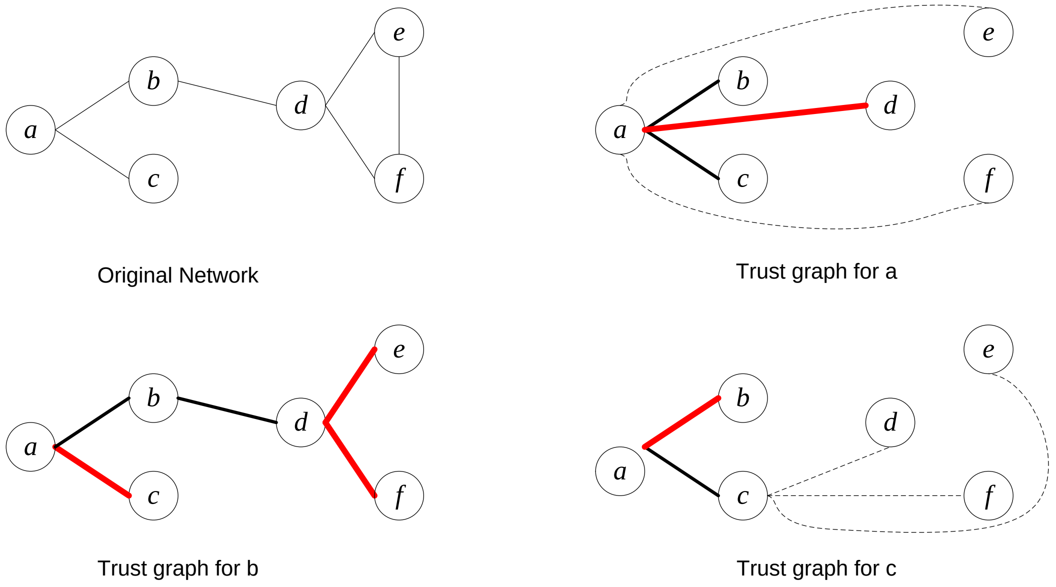
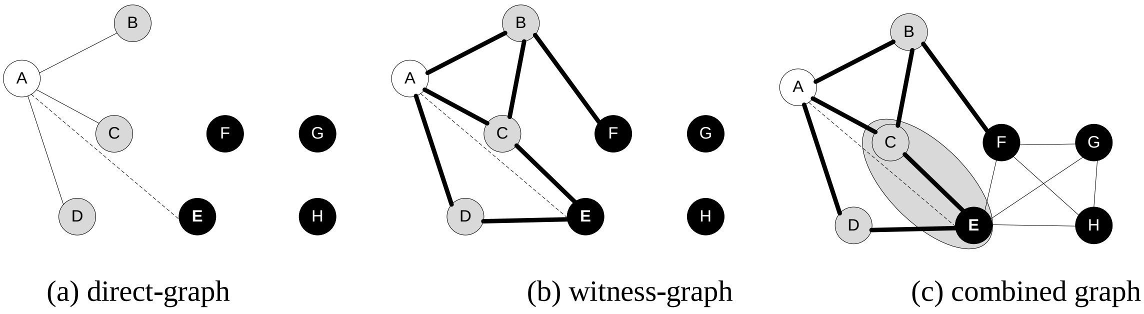


Figure 7-4: Example of trust graphs for nodes a, b and c in the network.
Thick black lines indicate direct interaction, Red lines indicate witness interaction and dotted lines indicate witness referrals.



(a) direct-graph (b) witness-graph (c) combined graph

Figure 7-5: Collusive behaviour in direct interaction with witness recommendations

Frequent similarity of recommendations from C and D, compared to other recommenders could suggest a potential case of collusion between these witnesses, especially if the opinions are inaccurate compared to the actual agent interaction. This is depicted by the Circle around C and E in Figure 7.5(c). Although B and D appear to have links to E, the comparison of their recommendations helps determine that C and E are potentially collusive. Agent B can also help identify the collusion between C and E by comparing trust recommendations from agent F. Witnesses collude, for example, to lower the trustworthiness of the target as viewed by the evaluator to prevent the target from being swamped with interaction requests, which could potentially increase competition for the witnesses' to interact with the target.

Three values are defined for the collusive behaviour of agents based on the local confidence value γ of a target agent. If there is a wide discrepancy in the value of recommendations for a target as received from more than one agent, there is a *probable* chance of collusive behaviour, in this case the value of confidence γ is reduced. If the difference is less, the chances of collusion are *doubtful* and therefore the value of γ is not modified. If the γ value of an agent is 0 or less that agent is considered to be untrustworthy and *definitively* involved in collusion. Algorithm 7.2 shows the process for determining collaboration between agents for a possible collusive behaviour. Incoming recommendations are stored in a waiting queue and the Witness based reputation is WR is calculated using equation 7.3. If the recommender agent is present in the History database H_i with a confidence $\gamma > 0$, it recommendation is incremented or decremented based on the acceptable value of difference δ in Trust value stored in H_i with calculated value of WR. If the confidence is repeatedly decremented so that it becomes 0 or less, the recommending agent is considered to be untrustworthy and is removed from H_i.

Algorithm 7.2 Updating γ for collusion detection

```
Calculate $WR_x$ based on eq. 3 for all $r_x \in$waiting_queue
for all $r_x \in$waiting_queue do
  if($r_x.a \in H_i$ and $r_x.\gamma > 0$ then
    if $(|T_{ix} - WR_x|) > 0$ and $(|T_{ix} - WR_x|) < \delta$ then //doubtful
      increment $r_x.\gamma$
    end if
    if $(|T_{ix} - WR_x|) > \delta$ then //probable
      decrement $r_x.\gamma$
    end if
    update $H_i.r_x$
  end if
  if($r_x.a \in H_i$ and $H_i.r_xa.\gamma = 0$) then //definitive
    $H_i$.remove($r_xa$)
    Waiting_queue.remove($r_x$)
  end if
```

7.4.5 Interaction Policies

Trust variables, history keeping and reputation variables are defined to determine the trustworthiness of agents in their interactions. Interaction policies use agent opinions and trust models to decide about which agents to interact with and which agents to ignore. Three kinds of policies, direct interaction policy, witness interaction policy and connection decision policy are defined.

Direct interaction policy

This kind of policy assists an evaluator agent in making decision about an agent based on direct interactions. Three direct interaction policies are,

- Always cooperative (AC). Agents using the AC policy for their direct interactions will always cooperate with their neighbours in direct interactions regardless of the action of their neighbour.
- Always defective (AD). Agents using the AD policy for their direct interactions will always defect with their neighbours in direct interactions regardless of the action of their neighbour.
- Limited Cooperation (LC) Agents using this kind of policy will cooperate only as long as they are trustworthy to each other. As soon as an agent becomes untrustworthy the agent will immediately disconnect from the neighbour.

Witness interaction policy

This kind of policy assists an agent in making decision about an agent based on witness interactions. Three categories of this policy exist.

- Replying policy: This policy assists in deciding what information should be given to a requesting agent. An agent can give true trust values (Honest), manipulate the trust values (Mislead) or provide false trust values (Lie). An agent employing the Lying policy (Lie) gives manipulated ratings to other agents by giving high ratings for untrustworthy agents and low ratings for trustworthy ones. The (Mislead) policy ranks all other agents as trustworthy but the honest (Honest) policy always tells the truth to everyone.
- Asking policy: This policy assists in deciding who should be selected to ask for information and where to look for trust values. The agent asks for trust values from its direct neighbours regarding target agents. Target queue stores agents whose reputation is to be investigated. Direct neighbours provide the trust values if they choose to provide target trust values (based on their respective replying policies). All the replies about a target are kept in waiting queue. Based on the recommendations received the target agent in question would be added to the

history database if it is deemed to be trustworthy. Any untrustworthy targets are removed from the subsequent queues. For a target whose trust recommendation is not yet known, i.e. it is neither trustworthy or untrustworthy, the target is added back to the target queue for a re-request of the information from direct neighbours of this particular target; assuming a target request can stay in the target queue for a specific amount of time.

- Update policy: This policy assists in deciding how to add/update the received information in the history database based on the reputation formulas for direct interaction reputation $DR_{i,k}$ and witness interaction reputation $WR_{i,k}$.

Connection decision policy

This policy helps determine if an agent should make a request for connections or accept/reject to a request for connection. The decision depends on the local trust value and the confidence in the requested agent. Two connection policies exist, conservative and greedy.

- Greedy Connection Policy. The evaluator agent connects to the agent that gives the most number of recommendations, believing that fact that it may have more connection to other agents thus increasing its chances to reach the target agent. However this kind of policy accepts connections regardless of the trust and confidence ratings of the agents.
- Conservative Connection Policy. The evaluator agent confirms other agents before making connections regardless of the number of recommendations received from a particular agent. If the recommending agent is already present in the history database of the evaluator agent and has a γ value larger than 0, connection can be made to the recommending agent.

7.5 Experiments and Results

FIRE+ is empirically analyzed to study the collusive behaviour of agents in an interactive society. FIRE+ is compared to the FIRE model with the parameters defined in Table 7.1. Although FIRE model simulates all of its four components, direct interaction, witness interaction, role-based interaction and third party certified interaction, only two kinds of interaction for evaluation is considered since the collusion problem exists in direct and witness interactions between agents.

Experiments study the connection drop rate to analyze effectiveness of proposed collusion prevention strategy. D_x is the average of dropped connections for agents of type x at a time interval t is given by

$$D_x(t) = \frac{\sum_{a \in x} D_{total}\ (a,t)}{N_x} \tag{7.5}$$

Where $D_{total}(a,t)$ is the total number of connections broken for agent a from start time to time t. N_x is the total number of agents of type x in the simulation.

Five kinds of agents are defined with various properties in the simulations as can be seen in Table 7.2. Honest, Lying and Misleading agents follow the policies defined in the previous section. Two kinds of trust aware agents are defined in the experiments; TA1 agents allow only direct interaction with targets. The second kind of agents are defined as TA2, these agents utilize the multi-dimensional model defined in section 7.4 and allow both trust and reputation calculations based on direct and witness interactions.

Table 7-1: Parameters considered for FIRE, FIRE+ comparison

Parameter	Value
History Size	20
Max Referral Chain Length	5
DIT range	1 < DIT < -1
WIT range	1 < WIT < -1
Total Agents	200
Trust Aware Agents	150
Honest Agents	10
Misleading Agents	20
Lying Agents	20
Initial trust ratings distribution value	0.25, 0.75
Initial trust ratings distribution variance	0.1
Expected number of new transactions per time step	10 to 30

In TA2, witness interactions are possible for a referral chain of up to 5 agents. The initial trust values are assigned for both TA1 and TA2 agents following a normal distribution with mean 0.25 and variance 0.1. For honest agents the trust ratings are generated using normal distribution with mean 0.75 and variance 0.1. Also for malicious agents (lying and misleading), trust ratings are generated using normal distribution with mean -0.5 and variance 0.1. The resulting random values are rounded off in the range [+1,-1] to three decimal places.

Table 7-2: Agent types and specifications

	Agent Types				
Trust Policies	**Honest**	**Lying**	**Misleading**	**TA1**	**TA2**
Direct	AC	AD	LC	LC	LC
Replying	Honest	Lying	Mislead	Honest	Honest
Connection	Conservative	Greedy	Greedy	Conservative	Conservative
Trust & Reputation				DIT & DR	DIT, DR, WIT & WR

7.5.1 Dropped Connections for TA1 agents

The number of dropped connections is compared for FIRE and FIRE+ in a network with 200 agents. 20% agents in this network are malicious (10% misleading, 10% always lying), 5% agents are honest and the rest of 75% are TA1 agents. The objective of this experiment is to study the effect of variation in the value of confidence for collusion detection (connectivity with risky agents) δ and compare the results with FIRE model. The simulation is run for 200 time steps and the value for D_{TA1}(t) is calculated against δ. Figure 7.6 shows, in comparison, FIRE model creates less number of dropped connections with FIRE+. The number of connections dropped is higher with the value of δ =0.2 and decrease with a higher value of δ. This shows that more connections are dropped when the value of δ is lower thus improving the quality of network by applying the conservative connection policy and a lowering the threshold for connectivity with risky agents.

Figure 7.7 shows the network diagram after 200 time steps. Black coloured agents are the TA1 agents, red coloured agents are malicious (lying or misleading) agents. The network diagram with FIRE model applied to agents can be seen in Figure 7.7 (a). Collusion prevention method defined in section 4 is not applied to FIRE, so malicious agents can maintain connections with TA1 agents and collaborate to decrease the overall Direct Interaction Trust (DIT) of the network. Figure 7.7 (b), shows the network diagram after 200 simulation steps for FIRE+ with TA1 agents at δ =0.2. It can be seen that the red coloured malicious agents are isolated from the majority of network. There is a high density of connections among the malicious agents depicting the collaborative behaviour but there is a very low number of connections between malicious and TA1 agents. This shows that the policies defined for FIRE+ are effective in reducing the risky collaboration among malicious agents and TA1 agents thus preventing the collusion attack for direct interaction trust in agents. As the value of δ =0.2 is increased the number of connections with malicious agents also increase, as can be seen in Figure 7.7 (c) and (d), however the degree of connections is far less than the original FIRE model.

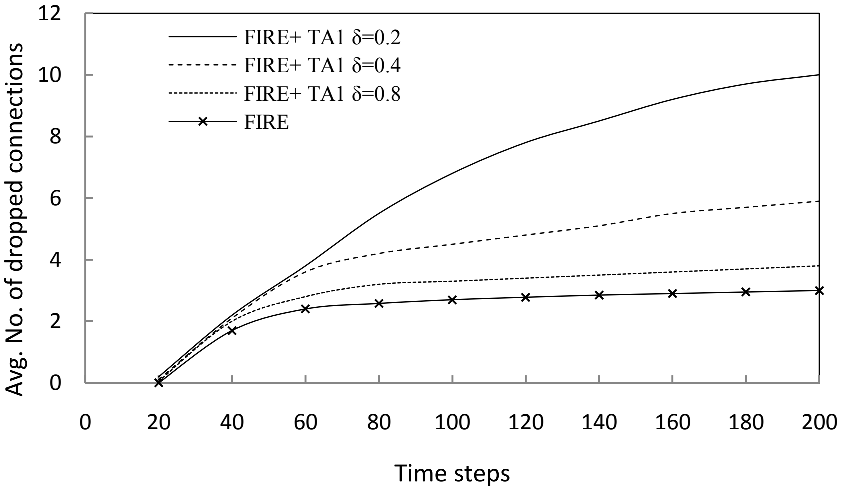
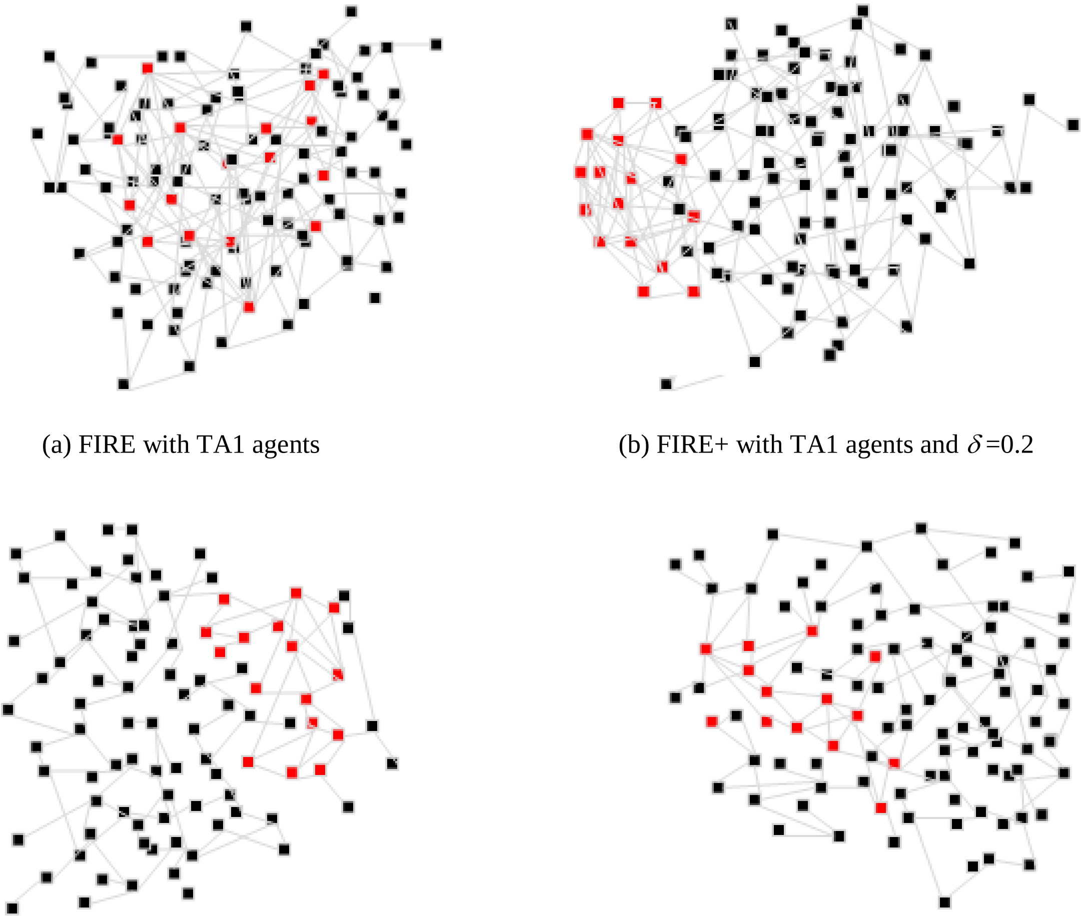


Figure 7-6: Comparison of Average number of Connections dropped for FIRE and FIRE+ (δ=0.2 δ=0.4 and δ=0.8) with TA1 agents

7.5.2 Dropped Connections for TA2 agents

Using the same parameters in simulation for TA1 agents, this experiment simulates the TA2 agents using the FIRE+ multidimensional model and the value for $D_{TA2}(t)$ is calculated against δ using equation 7.5. TA2 agents also utilize the conservative connection policy as with TA1 agents. Note that the difference between TA1 and TA2 agents is that TA1 utilizes the uni-dimensional model with direct interaction trust variable while TA2 agents use multidimensional model with witness based trust variables (WIT) and witness based reputation (WR) variables.

The objective of this experiment is to demonstrate the benefit of using a multi-dimensional model when there are witness based collusion attacks. Using the witness interaction trust and witness based reputation can decrease the impact of malicious agents (colluding groups) on aggregating the ratings.



(a) FIRE with TA1 agents

(b) FIRE+ with TA1 agents and δ=0.2

(c) FIRE+ with TA1 agents and δ=0.4

(d) FIRE+ with TA1 agents and δ=0.8

Figure 7-7: Final network diagram after 200 time steps. FIRE, FIRE+ with TA1 agents

As a result, the TA2 agents will expose themselves to a lower level of risk. As can be seen from Figure 7.8, TA2 agents have a significantly low number of average dropped connections (3.64) after 200 simulation time steps compared to TA1 agents (9.98) in the first experiment with confidence value of $\delta = 0.2$ thus lowering the risk of being exploited in a witness based collusion attack.. It can also be noted that the rate of connections dropped by FIRE and FIRE+ with TA2 agents is very similar; this shows that using TA2 agents with FIRE+ counters the time penalty in additional calculations done with FIRE+. Policies used by TA2 agent type result in successful acceptance/rejection of connection requests. In this sense, TA2 agents expose themselves to smaller numbers of untrustworthy agents and consequently lower the level of risk of being exploited by these agents. Figure 7.9(a) shows malicious agents (red) in FIRE can collaborate in witness interactions and influence witness recommendations thus reducing the overall witness interaction trust (WIT) of the network. Due to the enforcement of the set of policies defined in FIRE+, TA2 agents with δ=0.2 can effectively detect malicious and colluding agents as can be seen from Figure 7.9(b) therefore forming a non-collusive society by isolating malicious nodes. It can be seen the degree of connectivity of TA2 agents with malicious agents is minimal. Figures 7.9 (c) and (d) show FIRE+ with δ=0.4 and δ=0.8 respectively.

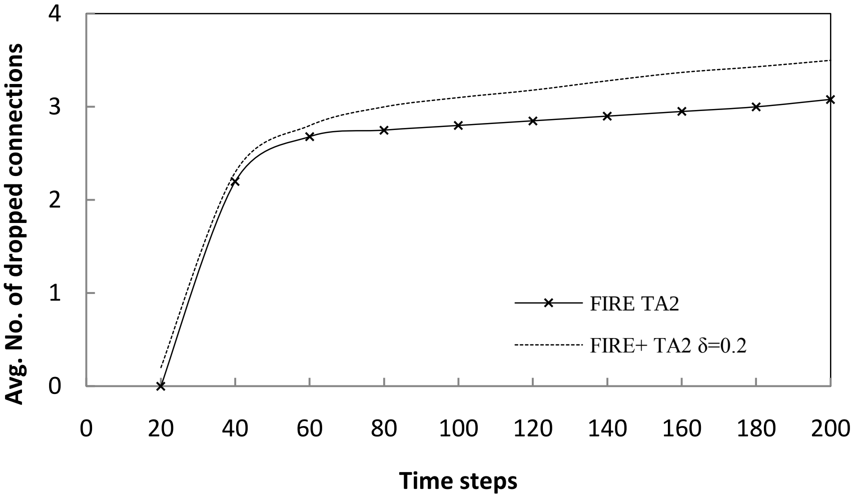


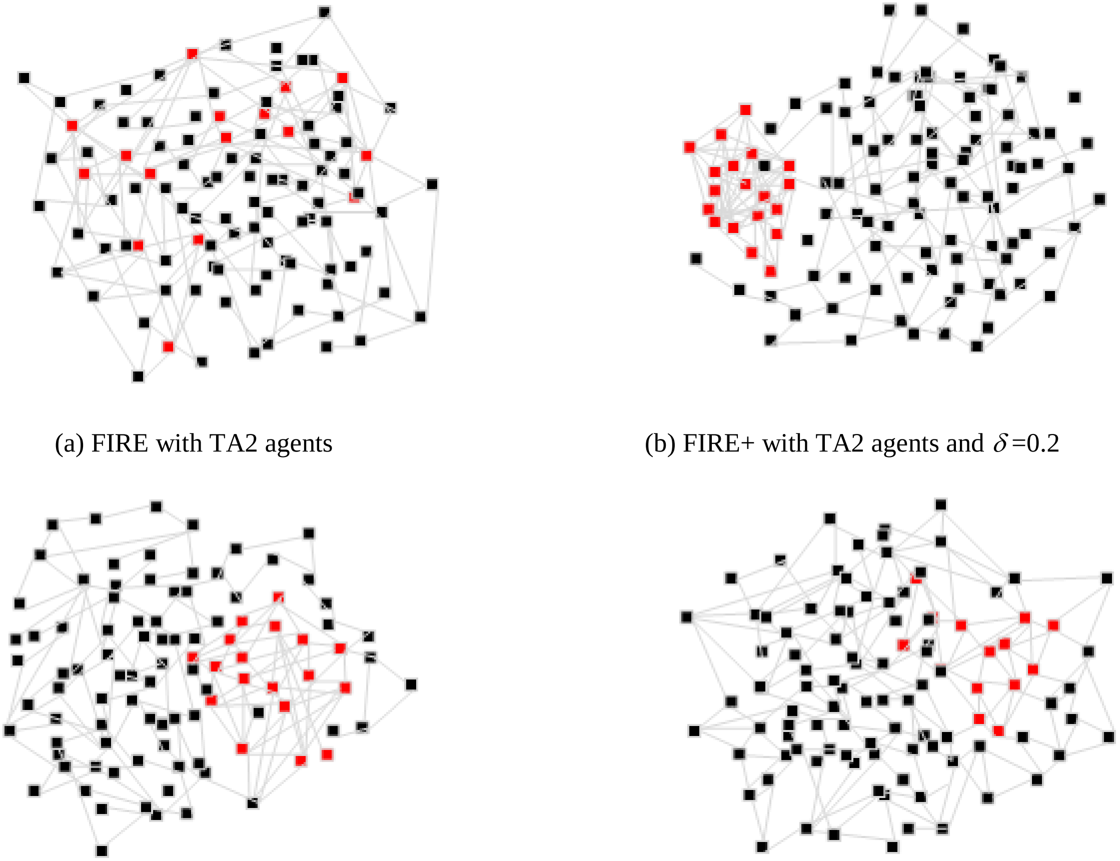
Figure 7-8: Comparison of Average number of Connections dropped for FIRE and FIRE+

7.5.3 Number of Connections with Malicious Agents

To better understand the degree of connectivity of an agent (TA1 or TA2) with a malicious agent, an overall average of number of connections and an average of number of connections with malicious agents is determined. Table 7.3 shows a relationship between the two types of agents TA1 and TA2 with FIRE and FIRE+. It can be seen that an average of 63% connections in a TA1 agent's history are made to malicious nodes compared to 76% connections with a TA2 agent, while both are using FIRE. This shows the vulnerability of FIRE to collusion attack. FIRE+ however shows much better results with both kinds of agents. TA1 agents using FIRE+ with δ=0.2 made an average of 0.15 connections with a malicious agent, meanwhile TA2 agents with the same parameters using FIRE+ and with δ=0.2 made an average of 0.22 connections, i.e. FIRE+ fails to detect collusion among agents less than 1% of times. This clearly shows the effectiveness of FIRE+ compared to FIRE.

7.5.4 Direct and Witness Trust Variables

In case of a collusion attack, the malicious agents falsely increase or decrease the trust values by providing false information to requesting agents. To study the effect of collusion on trust values stored in an agent's history, the average Direct Interaction Trust (DIT) value and the average of Witness Interaction Trust (WIT) values are computed for all agents in the network. FIRE and FIRE+ are compared with the δ=0.2, δ=0.4 and δ=0.8. Table 7.4 and Table 7.5 show the results from the comparison.



(a) FIRE with TA2 agents

(b) FIRE+ with TA2 agents and δ=0.2

(c) FIRE+ with TA2 agents and δ=0.4

(d) FIRE+ with TA2 agents and δ=0.8

Figure 7-9: Final network diagram after 200 time steps. FIRE, FIRE+ with TA2 agents.

A valid DIT value ranges in [-1, +1], while +1 being highly trustable, whereas -1 being untrustworthy. As can be seen from Table 7.4, the average DIT value for TA1 agents in FIRE is 0.7 while in FIRE+, is 0.2; this implies that the malicious agents have successfully increased the overall DIT value of the network. This suggests that most of the agents in FIRE deem each other trustworthy and fail to detect collusive behaviour in agents.

FIRE+ comparatively has an acceptable +0.1 DIT values which suggests that most agents are cautious in making connections with malicious agents therefore preventing the malicious agents from falsely increasing the trust values. Similarly, Witness Interaction Trust (WIT) value in Table 7.5 indicates an average of witness trust values stored in the history of an agent.

Table 7-3: Comparison of FIRE and FIRE+ in terms of average number of connections with malicious agents.

	Average Number of connections with Malicious Nodes per agent				**Average Number of connections per agent**	
Network Agent Type	δ=0.2	δ=0.4	δ=0.8	FIRE	FIRE	FIRE+
TA1	0.15	0.36	0.75	2.10	3.31	2.89
TA2	0.22	0.39	0.78	3.86	5.95	5.65

Table 7-4: Comparison of average Direct Interaction Trust (DIT) values for FIRE+

Agent Type	**Number of Agents**	**FIRE**	**FIRE δ=0.2**	**FIRE δ=0.4**	**FIRE δ=0.8**
Honest	10	0.5	0.2	0.3	0.4
Lying	20	-0.2	-0.6	-0.3	-0.1
Misleading	20	1.0	0.9	0.8	0.7
TA1	150	0.7	0.2	0.3	0.5
TA2	150	0.8	0.1	0.2	0.4

Table 7-5: Comparison of average Witness Interaction Trust (WIT) values for FIRE+

Agent Type	**Number of Agents**	**FIRE**	**FIRE δ=0.2**	**FIRE δ=0.4**	**FIRE δ=0.8**
Honest	10	0.9	0.1	0.3	0.6
Lying	20	-0.1	-0.7	-0.4	-0.1
Misleading	20	1.0	1.0	1.0	1.0
TA1	150	n/a	n/a	n/a	n/a
TA2	150	0.9	0.1	0.1	0.3

For TA2 agents that employ FIRE+ model with policies for collusion detection, it can be seen that the average WIT values is 0.1 compared to 0.9 in FIRE model. This clearly shows that FIRE is severely handicapped when dealing with witness based collusion attack whereas FIRE+ is far more effective in detecting the preventing collusion attacks. Further results can be found in appendix B.

7.6 Summary

FIRE+, a multi-dimensional trust and reputation model, is presented as an extension of FIRE trust and reputation model to detect and prevent direct interaction and witness interaction collusion attacks. In these attacks, agents which are trustworthy in their direct interactions, collude with malicious agents by providing a good rating for them and thus increase the trust ratings of a malicious group of agents. Results show that FIRE is susceptible to collusion attacks at direct interaction and witness interaction levels. Its inability to determine collaborative behaviour among malicious nodes results in agents forwarding false trust ratings and therefore increasing the overall DIT and WIT values.

The FIRE+ trust and reputation model defines mechanism for keeping a history of trust ratings and measure of confidence in ratings received from direct and witness interactions. The trust network graph determines the reliable ratings provided by direct and witness agents utilizing experience of interactions while synthesizing unreliable ratings from colluding / malicious agents with dubious recommendations. The determination of the value of confidence in trust values is crucial to the success of FIRE+. Various policies are defined to determine collusive behaviour and experiments show that TA2 agents using a multidimensional trust and reputation model while utilizing the trust policies can counter the risk of a direct interaction and witness interaction collusion attack by malicious agents. Finally, as a conclusion, multi-dimensionality is a crucial factor for resistance against witness-based collusion attacks.

Chapter 8

M-Trust: A Trust Management Scheme for Mobile P2P Networks

The rapid growth of emerging techniques for mobile open-access resource sharing, content sharing, mobile social networks, and complex cyber-physical systems poses significant challenges of efficient trust management. Many trust management schemes have been proposed recently to counter the security threat on P2P systems. However, due to the difficulties caused by system mobility, wireless communications, limitations of pervasive devices and the ever-changing network topology, there is an increasing requirements of decentralized and distributed trust management schemes. This chapter investigates, analyzes and compares various existing distributed and decentralized trust management schemes. Based on the analytical results, an efficient, accurate, robust and scalable light weight trust ratings aggregation scheme, referred to as M-trust is presented for mobile P2P networks. Results obtained from extensive simulations show that this proposed method can decrease the time required to compute the list of trust ratings and reduce the required storage space. The extensive comparison with other schemes shows that M-trust possesses the excellent overall performance in terms of accuracy, reliability, convergence speed, rate of detecting malicious peers under various constraints of mobility, trust threshold and network out-degree.

8.1 Introduction

The increasing popularity of online P2P services such as resource sharing, social networks and content/information retrieval has extended to mobile devices. The emergence of wireless networks, opportunities offered by 3G services, and rapid proliferation of mobile devices, have stimulated a general trend towards extending P2P characteristics to wireless environments. As a result, the P2P paradigm has migrated to pervasive computing scenarios.

Many P2P systems do not have the central administration and the peers are autonomous, making them inherently insecure and untrustworthy [BARA05]. To handle the trustworthiness issues of these services in open and decentralized environments, many trust and reputation schemes have been proposed to establish trust among peers in P2P systems. In a trust and reputation system, the historical behaviours and activities are recorded for each entity, and these statistics are used to predict how the entity is likely to behave in the future [ZHOU07]. Many studies have recently developed the decentralized trust and reputation systems and addressed various issues of trust and reputation management, such as GossipTrust – a gossip-based aggregation scheme [ZHOK07], FIRE - a decentralized trust model [HUNY06], H-Trust – a selective aggregation scheme [ZHOU08], FuzzyTrust [SONG05] and a reputation based trust management system [SELC08]. Moreover, several studies [LIJ08] and [LIMC08] have contributed to the framework design and middleware architecture for trust management.

Mobile P2P systems pose greater challenges in trust management due to the frequent topology changes in the network. To deploy a Mobile P2P system a straight forward approach is to mount a P2P system over a MANET [WUJ05]. MANETs are temporary wireless networks where the transitory sets of mobile nodes dynamically establish their own network on the fly. Nodes in a MANET are constrained by a limited amount of energy, storage, bandwidth and computational power. These limitations prove to be a hindrance in seamless connectivity with other peers and thus reducing the effectiveness of many trust and reputation systems. Since a reputation-based system requires trust ratings from other peers to evaluate or update trust scores, it is imperative that such a trust management system should be decentralized and can effectively aggregate trust ratings despite of delays, connection loss and malicious behaviour from peers. Moreover, as it is impossible to establish the global trust ratings for peers, any trust management scheme must take into account trust ratings at a local level and build a peer’s reputation based on accumulated ratings.

To ensure trustworthiness in Mobile P2P trust management systems, section 8.2 presents the effectiveness of various distributed and decentralized trust ratings aggregation schemes on MANETs. To this end, the popular trust schemes including the received ratings aggregation [LIMC08], weighted average of ratings [HUNY06], Bellman-Ford based algorithm [ZHAO09], total trust and ultimate trust schemes [BAHT10] are thoroughly investigated and compared. Based on the analytical results, an efficient, accurate, robust and scalable light weight trust ratings aggregation scheme, referred to as M-trust, for mobile P2P networks is presented in section 8.3. A trust ratings aggregation algorithm is proposed that acquires trust ratings not only from direct recommendations but also from recommendations from distant nodes. Results obtained from extensive simulations show that this proposed method can decrease the time required to compute the list of trust ratings and reduce the required storage space. The extensive comparison with other schemes shows that M-trust possesses the excellent overall performance in terms of accuracy, reliability, convergence speed, rate of detecting malicious peers under various constraints of mobility, trust threshold and network out-degree, as presented in section 8.4 followed by conclusion of this work.

8.2 Comparison of Existing Trust Ratings Aggregation Schemes

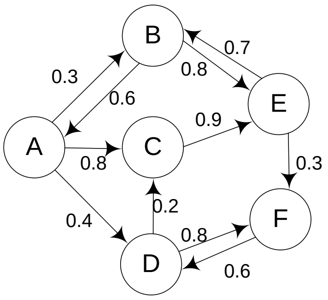
In an open and decentralized P2P environment, peers do not have any centralized authority to maintain and distribute reputation information. A full-aggregation reputation system calculates the reputation score of a peer by considering the opinions from all other peers who have interacted or non-directly interacted with this peer. Usually a full aggregation reputation system is of high accuracy. However, the aggregation approach involves a trade-off between the accuracy and overload. In an unstructured P2P network, the overload of the full reputation aggregation is quite heavy when the network expands very large. In addition, the reputation convergence is not fast. In the selective aggregation, reputation ratings are derived from a subset of the existing opinions in a distributed P2P network. In Mobile P2P networks, users with a higher trust level have the luxury to stay connected for the longer periods of time and communicate with a large number of users. Such users are able to store and forward data from adjacent nodes while serving as an intermediate router. This chapter addresses the trust ratings aggregations schemes. The trust rating values can be obtained by applying different functions to consider the importance of all the history transactions, date, service quality, etc.

Figure 8.1 illustrates a trust overlay network. The vertices in the graph correspond to peers/nodes in the network. An edge between peers A and B represents a connection between the peers if and only if A was a client of B in direct interaction. The real number *r* $\in$ [0,1] reflects how much A trusts B (T_{AB}=0 means A considers B as untrustworthy, T_{AB}=1 indicates A fully trusts B). As opposed to direct interaction trust, witness interaction trust is used to compute trust of a peer if no direct connection exists. In this case, all nodes that have a direct interaction with the evaluator node are asked to provide a trust rating for the target node. As an example in Figure 8.1(a), B has a direction trust interaction with A and E. If B seeks trust ratings for node C, it forwards the request to immediate neighbours A and E. Since A has a direct interaction with C, A can provide the trust rating for C. It must be noted that node E may have trust ratings for C made available through a longer path (E→F→D→C). All local and received trust ratings are stored in a table called trust list, *t_list*. Figure 8.1(b) shows a t_list for node B using the received ratings aggregation scheme presented in this chapter.

In what follows, a description and comparison of various trust schemes including received ratings aggregation [LIMC08], weighted average of ratings [HUNY06], Bellman-Ford based algorithm [ZHAO09], Total trust and Ultimate trust schemes [BAHT10] is presented.

8.2.1 Received Ratings

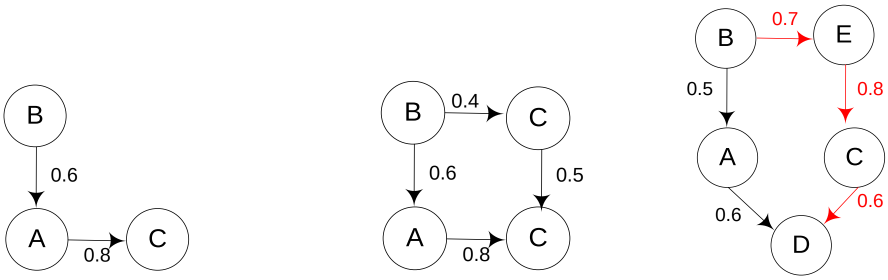
The received ratings aggregation scheme is based on the work presented in [LIMC08]. If the witness node has a high trust rating ($\tau_{ij} > threshold$) then the local peer's ratings are overwritten with the ratings provided by the witness. As an example shown in Figure 8.2, node B receives ratings for C from A. Assuming the threshold is set to 0.4, since T_{AB}=0.6 is larger than the threshold value therefore, the ratings provided by node A can be trusted. Node B subsequently updates/overwrites its own rating for C to 0.8. In case the trust ratings for witness node is less than the threshold, the two ratings are multiplied and the result is stored in the local trust list (e.g. if τ_{AB}=0.2 then new τ_{AB} = 0.2 * τ_{AC} =0.16, where τ_{AC}=0.8). This technique is simple to use and reduces the overall computation and updating overhead.



t_list for B using Ʈij			
Peer	T	witness	hops
A	0.6	-	0
E	0.8	-	0
C	0.48	A	1
C	0.02	D	2
D	0.24	A	1
D	0.14	E	2
F	0.24	E	1
F	0.19	A	2

(a) Trust Overlay Network (b) t_List with trust ratings based on $Ʈ_{ij}$

Figure 8-1: Trust Overlay Network



(a) Witness ratings (b) More trustable path (c) Longer trustable path

Figure 8-2: Trust ratings aggregation for witness interactions

8.2.2 Weighted average

The weighted average trust ratings aggregation scheme [HUNY06], has been widely used in the related work. Using this scheme, the witness information is sought if the direct interaction trust ratings are unavailable. All the received ratings are aggregated and a new trust value φ_{ij} is computed using the formula below

$$\varphi_{ik} = \frac{\sum_{j \in t\_list}(T_{jk} \cdot \varphi_{ij})}{\sum_{j \in t\_list} \varphi_{ij}} \tag{8.1}$$

where T_{jk} is the received trust ratings and φ_{ij} is the weighted average trust ratings stored in the *t_list*. The advantage of using this technique is the computation of trust based on an average function. As opposed to the received ratings technique, the witness information available from longer paths is not heavily penalized. A drawback of this approach is the high frequency need for computation; however this can be adjusted using an efficient aggregation algorithm. This requirement motivates the research into the M-trust ratings aggregation technique.

8.2.3 Bellman-Ford algorithm based scheme

As described in [ZHAO09], the trust aggregation scheme based on Bellman-Ford algorithm computes trust using the direct and trust transfer method. For each direct transaction in the system, the participating peers generate a direct trust link and assign a trust rating to represent the quality of this transaction. Each transaction in the system can either add a new directed edge in the trust graph, or re-label the value of an existing edge with its new trust value or a compound value of both old and new trust ratings. For witness interaction trust, all trust ratings on a path is multiplied to compute the trust value. As an example in Figure 8.2(c), if B seeks trust ratings for D, since there is no direct link, no trust rating for D exists in *t_list* stored at B. Therefore, B requests trust ratings from A and E. A has a 2 hop connection to D and returns a trust value $\omega_{BD} = \omega_{AD} * \omega_{BA} = 0.5 * 0.6 = 0.3$. On the other hand, $\omega_{BD} = \omega_{BE} * \omega_{EC} * \omega_{CD} = 0.7 * 0.8 * 0.6 = 0.34$.The trust aggregation algorithm [ZHAO09] considers the most trustable path instead of the shortest path for computing witness interaction trust. A drawback of using this approach is that it can cause the occurrence of trust loops because the Bellman-Ford algorithm does not prevent loops from happening. This can be countered by adding a counter to count the number of hops and setting a max hops limit.

8.2.4 Ultimate and total trust

Bahtiyar, Cihan, Aglayan presented a method to calculate the ultimate trust for P2P overlay networks [BAHT10]. This method considers reputation based on various factors such as confidence in interaction along with risk factor. The values of these variables are determined by positive and negative interactions with other peers. The ultimate trust ratings U_{AB} is computed over a period of time where peers adjust trust ratings based on interactions. The risk factor of a node increases if an expected service was not provided, whereas the confidence increases after completion of desired service. One of the drawbacks of this scheme is that it is computation-intensive and requires time to compute trust from all possible witnesses.

8.3 M-trust Trust Ratings Aggregation Scheme

In a Mobile P2P network, some peers join or leave the network frequently, which leads to the dynamic topology changes. Due to these changes, a trust management system needs to frequently update trust ratings, which in turn can increase the communication overhead. Pervasive devices that are resource-constrained need to avoid unnecessary updates and thus decreasing the overall communication overhead. In this section an efficient trust ratings aggregation scheme that reduces the frequency of updates in acquiring trust ratings is presented. M-trust integrates parts of the trust aggregation schemes presented in section 8.2 and apply these to direct and witness interaction trust aggregation in the proposed scheme.

Algorithm 8.1 describes the procedure of the proposed trust ratings aggregation scheme, M-Trust. This scheme takes into account the trust ratings based on direct and witness interactions. In a scenario with two possible paths, this proposed algorithm considers the best path based on confidence,γ. The value of $\gamma \in [0,1]$ is calculated by individual peers based on the number of positive and negative interactions with a peer. γ is used for trust ratings aggregation from witness interactions only. Since it is possible that a peer can act maliciously and provide false trust ratings, the value of γ can determine the behaviour of peer in recent interactions. A newly joined peer can send a trust request to the network. Peers that receive the trust request choose to send back their trust lists. When a new peer receives this reply, its initial local trust list is established. It is possible that a trustworthy peer gets disconnected due to power shortage or physical location change in the MANET environment, for this reason each entry in the *t_list* has a *ttl* variable indicating time to live. If the *ttl* has exceeded a threshold and no connection could be established for a node, its trust ratings are removed from the *t_list*.

Algorithm 8.1 Trust Rating Aggregation Algorithm

```
initialize t_list for all peers
loop
for each request($x$) do
   reply($x$, t_list)
   if x is direct peer and $x \notin$ t_list then
      t_list.add(request($x$))
   end if
end for
for each $x$ where $x \in$ t_list do
   if x is direct peer then
      if t_list.value <= request(x).value and t_list.value > threshold then
         t_list.update(request(x).value)
      else
         request (y) where $y \neq x$ and $y \in$ t_list
      end if
   end if
   if x is not direct peer then
       t_list.update(weighted average function)
   end if
   if t_list.ttl(x) = 0 then
      if t_list.update(request($x$))= no reply then
         remove.t_list(x)
      end if
   end if
end for
end loop
```

If x is a direct peer and rating of x in the local *t_list* is higher than threshold then x can be trusted. Recommendations from x for peer y are stored in the local *t_list*. If x is a direct peer but has poor trust ratings, then a second opinion about recommendations is sought from other direct neighbours. Any direct neighbour in a position to give recommendations about y, while having a higher trust rating compared to x is considered and the local t_list is updated accordingly. In case of witness recommendations, if the recommending node can be trusted by a direct neighbour then a weighted average of the recommendations is used to calculate trust. This is due to the fact that multiple ratings would be received for witness based interaction and a weighted average would provide balanced trust ratings. γ (Confidence) is defined to update trust ratings with the distant peer based on the behaviour of this peer. If the number of completed interactions is larger than incomplete interactions, the value of γ is increased. Alternatively as a consequence to a large number of incomplete transactions this value is decreased. In witness interactions, the value of γ is multiplied to the calculated trust in order to obtain the witness trust interaction value. If the ttl for a trust rating stored in the local *t_list* expires, an update request is made to node x to provide the latest *t_list*. If no reply is received, the node is assumed to have been disconnected, any subsequent trust ratings are therefore removed from the local *t_list*.

After the aggregation process, each local peer has established a trust list, which represents the current local view of the network. When there is a need to obtain a trust value on a remote peer, trust search will initiate. As an example, B in Figure 8.1 needs to acquire trust ratings for C. Since there is no direct interaction trust rating available for C, a search is requested to direct peers A and E. A receives the request and replies with the value T_{AC}=0.8, since it can be found in the local *t_list* of A. if the trust threshold was set to 0.5, this indicates that A can be trusted by B, therefore B overwrites its trust rating for C, T_{BC}=0.8. On the other hand, E has a path to C given by E→F→D→C. Since there is no direct path and E has to rely on witness information therefore the weighted average is used to calculate the overall rating for C. Moreover, T_{DC}=0.2 and T_{EF}=0.3 indicate that these peers are considered to be untrustworthy; therefore the confidence value γ is to be used to calculate the trust ratings. If no previous encounters with the target node C exists then the value of γ is assumed to be 0.5.

8.4 Experiments and Performance Evaluation

Extensive simulation experiments have been conducted to evaluate the performance of the proposed trust ratings aggregation scheme (M-Trust) using a modified Madhoc simulator

[HOGI]. Madhoc is a metropolitan MANET simulator that allows the simulation of large networks. The mobility of users and therefore that of the mobile hosts they are carrying are simulated using a variant of the random waypoint model: a user can remain motionless for a while; afterwards he/she begins to move towards a set destination, which is selected randomly in the simulation area. The experiments carried out, consider a simulation scenario in which 200 users move within a 1 km × 1 km area with a constant speed from a source to a random destination.

8.4.1 Initialization of simulation experiments

For honest peers, the initial trust values follow a normal distribution with mean μ_n= 0.85 and variance σ_n^2=0.1. However, for malicious peers the initial trust values follows a normal distribution with mean μ_n= 0.15 and variance σ_n^2=0.1. The out-degree, D, represents the number of connections a peer can make. Initially, the peer out-degree D=6 is determined by a normal distribution with mean μ_D= 6 and variance σ_D^2=1. New transactions are continuously generated according to a Poisson distribution with an arrival rate λ = 10 to 30 transactions per minute, between a random source node and a random destination node. The simulation generates network topology, and initializes local trust values with the given distribution. Table 8.1 summarizes the various simulation parameters. Initially the trust ratings are acquired from direct neighbours only. After the initial trust values are established, the further trust ratings can be requested over a multi-hop chain of nodes with a maximum chain length of 3 hops. To make the figures that depict the performance results clear, symbols τ, ω, φ, U, and T are used to represent the results of the received ratings technique, Bellman-Ford based algorithm, weighted average technique, ultimate trust technique, and the proposed M-trust scheme, respectively.

8.4.2 Congregation time and t_list size

The initialization time of trust values is the number of iterations taken for the trust management technique to obtain the trust ratings of other nodes. In a dynamic system, it may be impossible to obtain the trust values for all nodes in a limited time; therefore it is assumed that a congregation state, C_x, occurs if the value of C_x reaches a pre-set threshold value. C_x is given by

$$C_x = \frac{\sum_{a\epsilon x} S_{t\_list\ (a,t)}}{N_x^2} \qquad (8.2)$$

where $S_{t\_list(a,t)}$ is the number of entries in *t_list* of node *a* at time *t*, and N_x is the number of nodes in the simulation.

TABLE 8-1: SIMULATION PARAMETERS

Parameter	Value
Mobility speed	0.5m/s, 2m/s
Pause time	30s – 2 min
Max connections (D)	3, 6, 9
Network size	200 nodes in 1km x 1km grid
trust *threshold*	0.1 to 1.0
Malicious peers θ	15%, 30%

Figure 8.3 reveals the comparison of the trust ratings aggregation techniques with the out-degree D=3, 6 and 9. The value of congregation threshold is set to 0.2. It can be seen from the figure that with the higher degree of connectivity D, the more connections are made thus increasing the number of entities in the trust lists. This is reasonable since more connections permit a node to acquire the trust ratings from a larger set of nodes per iteration. The results also show that as the complexity of the network increases the congregation time decreases.

The proposed trust ratings scheme, represented by curve T in the figure, fares slightly better than other techniques for all the selected values of out-degree D. This is primarily due to the fact that this technique acquires trust information both from direct neighbouring nodes and witness providing nodes. Comparatively the received ratings technique τ, and Bellman-Ford based algorithm ω, utilizes trust ratings from immediate neighbour nodes only. The weighted average technique φ, relies on witness information if available and is therefore very similar to the proposed technique. The ultimate trust technique U is the slowest due to its reliance on acquiring trust ratings from neighbouring nodes with the highest confidence.

The average *t_list* size is shown in Figure 8.4(a). After 50 simulation iterations, the weighted average trust ratings technique has the largest average number of entries in the *t_list*. M-trust has the second highest compared to the ultimate trust which has the smallest *t_list* size. The algorithm for the proposed technique reduces the size of *t_list* by removing the entries that have exceeded the time limit. This process slows the overall performance of the algorithm; however this is effective in reducing the length of the *t_list* which can improve the query rate for trust ratings in M-trust.

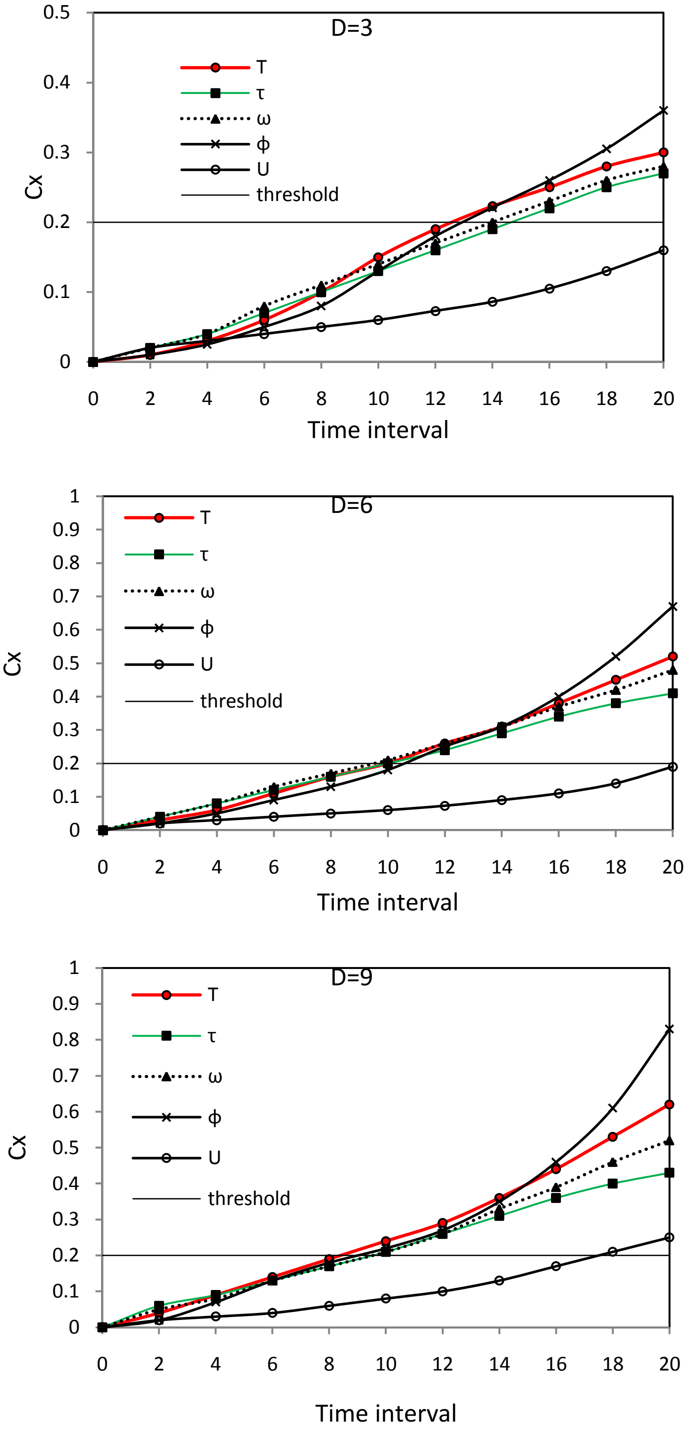
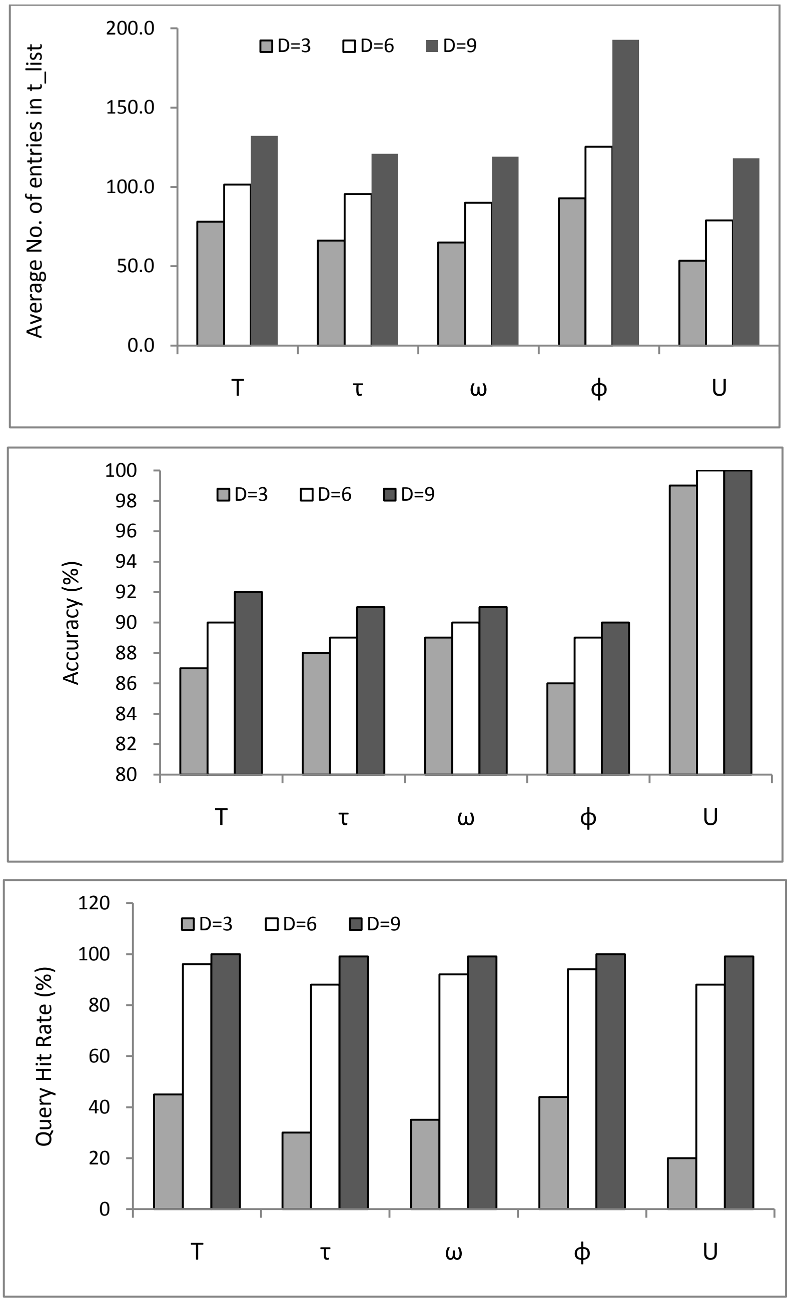


Figure 8-3: Comparison of trust management techniques with out-degree D = 3, 6 and 9.

8.4.3 Query hit rate and accuracy

The accuracy of trust ratings in a node is determined by comparing its inferred trust ratings based on its experience in behaviour of other nodes. The global average trust ratings are determined at the end of the simulation. Figure 8.4(b) shows the comparison of the accuracy of all the trust ratings aggregation techniques. All the trust ratings aggregation techniques provide an accuracy of at least 85%. The ultimate trust management technique is the most accurate as it determines the trust value from the nodes with the highest trust ratings. The other techniques are comparatively close to the M-trust in terms of accuracy.



(a) Avg. No. of entries in t_list (b) Accuracy (%) of inferred trust values (c) Query hit rate (%) of trust rating inquiry

Figure 8-4: Comparison of trust management techniques with D = 3, 6 and 9.

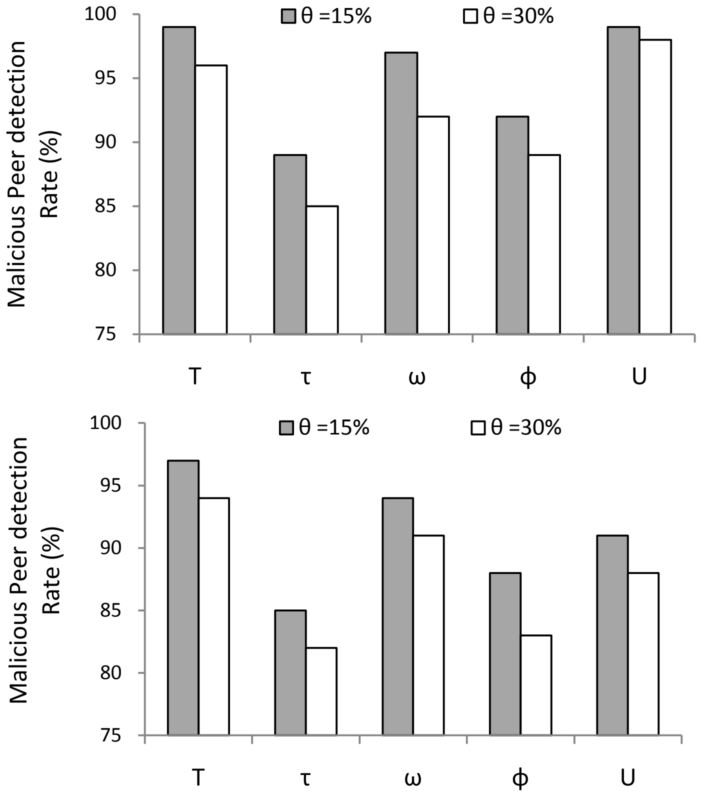
It can be seen that M-trust provides an accuracy of 92% when the degree of connectivity D is 9. However, as the value of D decreases, the network trust rating aggregation accuracy is reduced to 87%. If an accuracy rating of 90% to be acceptable, it can be seen that the degree of connectivity has a significant effect on the accuracy. This is due to the fact that with lower number of connections, trust paths become longer, which leads to a higher degree of inaccuracy due to the existence of malicious activity in the network. Query hit rate is defined as the percentage of the number of successful queries in *t_list* for a trust rating request. A higher value of query hit rate indicates that the request was fulfilled and further requests are not needed, effectively reducing the overall amount of traffic in the network. In mobile networks this is crucial to the success of effective transmissions and bandwidth control. A higher value of D yields the better percentage of query hits. Figure 8.4(c) shows the effect of network connectivity on the successful query hits.

8.4.4 Malicious peer detection rate with Mobility

Malicious peers are introduced with a set of 15% and 30% malicious nodes in the network. Figure 8.5(a) shows the performance results when the detection rate D=6, the number of malicious nodes is 15% and 30%, respectively, and the node mobility is set to 0.5 m/s. It can be seen that the ultimate trust ratings aggregation technique provides the best peer detection rate. Comparatively the performance of M-trust is 99% for θ =15% with a mobility of 0.5 m/s. With θ =30% the malicious peer detection is an acceptable 96%. Figure 8.5(b) reveals the effect of mobility on the performance of all techniques. It can be seen that the ultimate trust provides the poorer performance for a higher mobility of nodes. This is due to the fact that the ultimate trust aggregation depends heavily on trust recommendations from peers with high trust ratings. Comparatively, M-trust manages a 93% malicious peer detection rate with higher mobility and larger number of malicious nodes. The results demonstrate that M-trust is suitable for a network with a decentralized topology such as MANETs.

8.4.5 Trust threshold confidence

Figure 8.6(a) reveals a comparison of results for M-trust based on different trust threshold values. The higher value of trust threshold means that the fewer nodes are considered trust worthy. This reduces the overall accuracy of M-trust. With a higher out-degree value D=9 and high trust threshold, the accuracy of M-trust is close to 100%. On average, the accuracy of M-trust is above 90% for all trust threshold values larger than 0.4 and with out-degree D=9. Figure 8.6(b) shows the comparison of value of C_x versus trust threshold value. It can be seen that with a high trust threshold the average number of entries in the trust list is reduced. Due to the fewer entries and inadequate information the accuracy of M-trust is suffered.



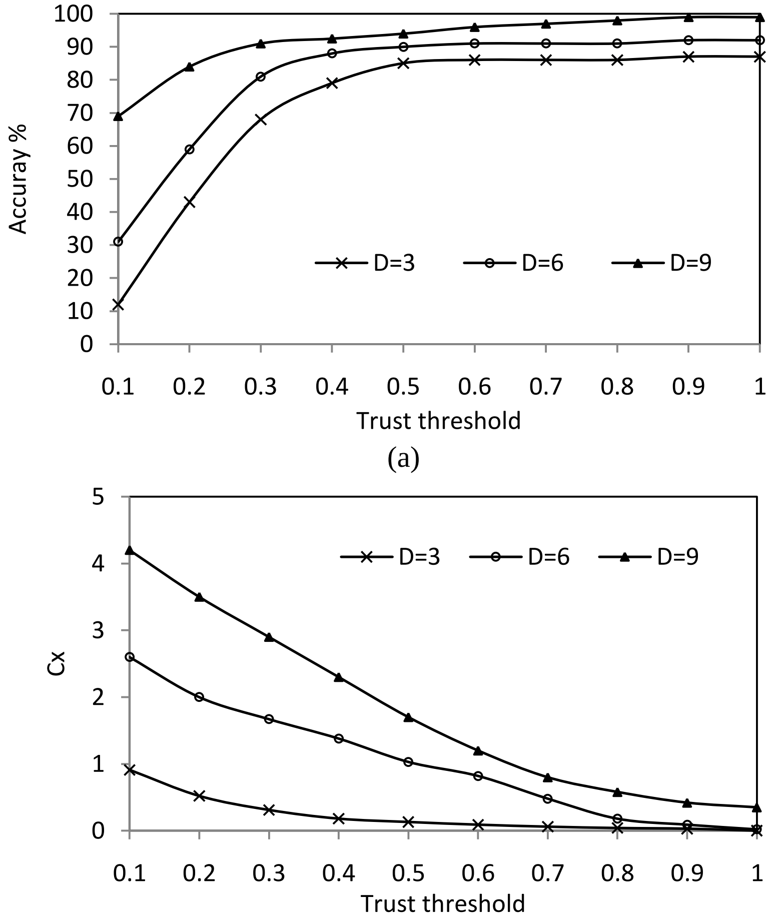
(a) D=6 with 0.5 m/s (b) D=6 with 2 m/s

Figure 8-5: Malicious peer detection rate with mobility

On the other hand, with the lower trust threshold the value of C_x approaches 4.3 with D=9. This means, on average a trust list contains a large number of entries. Since mobile devices are incapable of handling the large amounts of storage, it is recommended that a value of 0.45 should be used for the trust threshold with D=6, which gives an acceptable accuracy of 90% and C_x value of 1.12 which is almost $O(N_x)$ where N_x is the total number of nodes in the simulation.

8.5 Summary

This chapter presented a new trust management scheme (M-trust) for mobile P2P networks. M-trust relies on direct trust ratings and witness recommendations from reliable peers to determine trust ratings for a node using a proposed trust ratings aggregation algorithm. Simulation results demonstrate that the overall performance of M-trust is accurate, reliable and robust for detecting malicious peers in P2P mobile networks. Four trust management techniques with different trust rating aggregation algorithms were compared with M-trust to analyze the performance of the proposed technique. M-trust performs better in terms of obtaining trust ratings over a fixed period of time. M-trust also removes the redundant and out-of-date information from the trust lists and thus reducing the amount of storage required. Although the ultimate trust technique is most efficient in terms of accuracy, it requires heavy computation and is dependent on trust ratings from most reliable nodes only. Moreover, it proves to be inconsistent with mobility due to less number of interactions and frequent disconnections. The accuracy of M-trust is acceptable compared to other techniques under various conditions of mobility and different combinations of trust threshold, query hit rate and network out-degree.



(a)

(b)

Figure 8-6: Trust threshold values for M-trust

(a) Accuracy % with trust threshold and out-degree D (b) Value of C_x with trust thresholds and out degree D

Chapter 9

Conclusions and Future Work

9.1 Conclusions

P2P mobile application services implemented by popular online service providers such as file sharing (gnutella), social networking (facebook) and health monitoring (medapps), simply extend the user interface to mobile devices without realizing the inherent problems of mobile communication. To date very few de-centralized mobile P2P services have been implemented due to the enormous challenges posed by the dynamic nature of the networks. Mobile ad-hoc networks and P2P systems are technologies which share a common underlying decentralized networking paradigm. This work addresses the issue of developing an overlay abstraction for P2P applications in a delay tolerant disconnected MANET environment. A framework that allows forming of trust based communities in a disconnected and delay tolerant MANET is proposed in chapter 3. Users can share content and transfer files in an opportunistic manner utilizing store-carry-forward paradigm. The framework is designed in J2ME Personal Profile and tested on devices using Windows Mobile 6.0. Through experiments and user trials, the framework successfully constructs communities between nodes that contact each other opportunistically in close proximity and ad hoc manner. The work also shows, using a light weight trust model, to identify trustable and untrustworthy users based on social contacts.

The shortcomings of the framework were analyzed and two protocols for implementing the store-carry forward mechanism and improving the overall performance of file delivery are proposed. The Content Driven Data Propagation Protocol (CDDPP), in chapter 4, is a light weight data propagation protocol suitable for content driven profile based P2P applications. The protocol fully exploits the store-carry-forward mechanism to deliver files to intended destinations. Extensive simulations are carried out to study the impact of various factors on the performance of CDDPP. The Adaptive Opportunistic Routing Protocol (ORP), an enhanced version of CDDPP is presented in chapter 5. The ORP protocol is based on an opportunistic routing mechanism for content sharing between users with similar interest profiles (content). Simulation results show that P2P data transfer over multiple hops in the network present faster data dissemination in the network. It was shown that sharing documents of various sizes over multi hop neighbours is possible with different degrees of success. The work carried out also experiments with mobility of nodes in the network, ORP protocol improves delivery rates of messages when specific nodes store and forward documents with greater speed into communities of users. The effect of size of data files stored in the repository and the frequency of the need to update repository was studied. With smaller file sizes i.e. less than 512KB, the protocol efficiency for repository update is above 95%. Experiments carried out also studied data forwarding to neighbours at multi hop distances. Simulation results of the adaptive opportunistic routing protocol show a minimum of 90% delivery rates over a multi-hop DTN, which is acceptable for data sharing in this kind of networks.

Trust is one of the most crucial concepts for decision in making relationships in human societies. Trust management in dynamic decentralized mobile networks is receiving attention due to its immense application. In early stage of trust and security on MANETs, several trust and security establishments relied on cryptographic methods, authentication codes and hashing chains for their solutions. Although these schemes are effective, they are essentially centralized systems which are not applicable to disconnected MANETs because of the dynamic movement of nodes and the lack of pre-existing infrastructure. Recently reputation based trust management systems have gained popularity. A reputation based trust management system computes trust based on a history of nodes' encounters with other nodes and recommendations from other users of the system. In this work algorithms for decentralized trust management in P2P Mobile applications based on a dynamicity aware graph relabeling system were presented in chapter 6. The proposed algorithms are based on greedy concept and the results affirm the benefits of using this approach. Simulations show that the proposed algorithms successfully create groups with higher trust levels and isolated nodes that have low trust ratings. Chapter 7 presents an extension of FIRE trust and reputation model, to detect and prevent direct interaction and witness interaction collusion attacks. In these attacks, agents which are trustworthy in their direct interactions, collude with malicious agents by providing a good rating for them and thus increase the trust rating of a malicious group of agents. It has been shown that FIRE is susceptible to collusion attacks at direct interaction and witness interaction levels. Its inability to determine collaborative behaviour among malicious nodes results in agents forwarding false trust ratings and therefore increasing the overall DIT and WIT values. The FIRE+ trust and reputation model defines mechanism for keeping a history of trust ratings and measure of confidence in ratings received from direct and witness interactions. The trust network graph determines the reliable ratings provided by direct and witness agents utilizing experience of interactions while synthesizing unreliable ratings from colluding / malicious agents with dubious recommendations. The determination of the value of confidence in trust values is crucial to the success of FIRE+. Various policies were defined to determine collusive behaviour and experiments carried out, show that TA2 agents using the FIRE+ multidimensional trust and reputation model while utilizing the trust policies can counter the risk of a direct interaction and witness interaction collusion attack by malicious agents. Multi-dimensionality is a crucial factor in resistance against witness-based collusion attacks, for P2P applications, in delay tolerant MANETs.

This thesis also presented a new trust management scheme (M-trust) for mobile P2P networks as reported in chapter 8. M-trust relies on direct trust ratings and witness recommendations from reliable peers to determine trust ratings for a node using a proposed trust ratings aggregation algorithm. Simulation results demonstrate that the overall performance of M-trust is accurate, reliable and robust for detecting malicious peers in P2P mobile networks. Four trust management techniques with different trust rating aggregation algorithms were compared with M-trust to analyze the performance of the proposed technique. M-trust performs better in terms of obtaining trust ratings over a fixed period of time. M-trust also removes the redundant and out-of-date information from the trust lists and thus reducing the amount of storage required. Although the ultimate trust technique is most efficient in terms of accuracy, it requires heavy computation and is dependent on trust ratings from most reliable nodes only. Moreover, it proves to be inconsistent with mobility due to less number of interactions and frequent disconnections. The accuracy of M-trust is acceptable compared to other techniques under various conditions of mobility and different combinations of trust threshold, query hit rate and network out-degree.

9.2 Contributions

In summary the following contributions were made to the ongoing research in trust management for P2P applications in disconnected DTNs.

1. A trust based generic decentralized P2P services framework for disconnected MANETs was presented. The architecture of the proposed framework is based on

three layers, application layer, trust layer and content manager layer. The proposed framework works as an overlay, on a disconnected delay tolerant MANET and is designed for applications utilizing the opportunistic connectivity for communication. A store-carry-forward protocol is presented along with trust based connectivity and content sharing mechanism. Three generic P2P applications for testing purposes were built. The framework is successfully tested using Bluetooth communication medium on Personal Digital Assistant (PDA) Devices.

2. A Content Driven Data Propagation Protocol (CDDPP) is proposed. CDDPP is a light weight protocol for profile based file sharing P2P applications (Mobile Social Networks). Results prove the effectiveness of CDDPP with simulations in a delay tolerant MANET using various parameters.
3. An Adaptive Opportunistic Routing Protocol (ORP) for content driven data dissemination in disconnected MANETs (for file sharing app) is presented. The protocol fully exploits the store-carry-forward mechanism for data transmission in a multi-hop disconnected MANET. Various simulations show that P2P data transfer over multiple hops in the network present faster data dissemination in the network. Results show that sharing of various sizes of documents over multi hop neighbours is possible with different degrees of success. ORP protocol improves delivery rates of messages when specific nodes store and forward documents with greater speed into communities of users. This work also studied the effect of size of data files stored in the repository and the frequency of the need to update the repository
4. A decentralized distributed trust management scheme is presented for P2P applications using the Dynamicity Aware Graph Relabeling System (DAGRS). The DA-GRS uses a distributed algorithm to identify trustworthy nodes and generate trustable groups while isolating misleading or untrustworthy nodes. Several simulations in various environment settings show the effectiveness of the proposed scheme.
5. FIRE+, an extension of FIRE trust and reputation model to detect and prevent direct interaction and witness interaction collusion attacks is presented. This work shows that FIRE is susceptible to collusion attacks at direct interaction and witness interaction levels. Its inability to determine collaborative behaviour among malicious nodes results in agents forwarding false trust ratings and therefore increasing the overall DIT and WIT values. The FIRE+ trust and reputation model defines mechanism for keeping a history of trust ratings and measure of confidence

in ratings received from direct and witness interactions. The trust network graph determines the reliable ratings provided by direct and witness agents utilizing experience of interactions while synthesizing unreliable ratings from colluding / malicious agents with dubious recommendations. Several policies were defined to determine collusive behaviour. Simulation results show that agents using FIRE+ multidimensional trust and reputation model while utilizing the trust policies can counter the risk of a direct interaction and witness interaction collusion attack by malicious agents.

6. To ensure trustworthiness in Mobile P2P trust management systems, this work presents an effective distributed and decentralized trust ratings aggregation schemes for MANETs. The popular trust schemes including the received ratings aggregation, weighted average of ratings, Bellman-Ford based algorithm, total trust and ultimate trust schemes are thoroughly investigated and compared. Based on the analytical results, an efficient, accurate, robust and scalable light weight trust ratings aggregation scheme, referred to as M-trust, for mobile P2P networks is presented. A trust ratings aggregation algorithm is proposed that acquires trust ratings not only from direct recommendations but also from recommendations from distant nodes. Results obtained from extensive simulations show that this proposed method can decrease the time required to compute the list of trust ratings and reduce the required storage space. The extensive comparison with other schemes shows that M-trust possesses the excellent overall performance in terms of accuracy, reliability, convergence speed, rate of detecting malicious peers under various constraints of mobility, trust threshold and network out-degree.

9.3 Future Work

The following considerations can be studied as future research directions to the work presented in this thesis.

1. The routing protocols presented in this thesis were tested (simulations) using the random waypoint mobility model. Since most mobile P2P applications are utilized in a social setting, the human mobility models need to be applied to study the impact of various models on an efficient framework design. Work presented in chapter 6 characterizes three mobility scenarios; although random waypoint mobility model is used, new mobility models need further investigation.

2. The current version of the ORP protocol works under limitations of file sizes to be transferred to other nodes. In future I am considering utilizing the mechanism of splitting larger files in numerous blocks for transfer instead of a larger file. This merits investigation into torrent style distributed file sharing in disconnected MANETs.
3. Since most commonly used P2P services in the mobile networks are based on social interaction between users, future work should consider social-aware or social inspired wireless networks where the knowledge of social network users is exploited for the benefit of wireless network design.
4. Embedding FIRE+ trust management model in the framework to study the effectiveness of the model in delay tolerant MANETs. Developing a larger test bed to test the framework with larger set of devices and users.
5. The DA-GRS based greedy algorithms allow users to create groups; trust ratings are associated with individual users and groups. In future the impact of group dynamics on mutual trust ratings needs further investigation. Studying group trust in group dynamics can lead to improved protocol design for disconnected MANETs based on social aware routing mechanism.

References

[ALCH08] Alchaita M. (2008), “Link longevity and mobility in self-dependent multi-hop mobile environments”, International Journal of Mobile Communications Vol. 6, No.4, pp. 519 – 539.

[ALMA06] Almenarez F., Marın A., Dıaz D. and Sanchez J. (2006). “Developing a Model for Trust Management in Pervasive Devices”. In Third IEEE International Workshop on Pervasive Computing and Communication Security (PerSec’06) held in conjunction with IEEE PerCom 2006. March,2006.

[ALMA08] Almenarez, F. Marin, A. Diaz, D. Cortes, A. Campo, C. Garcia-Rubio, C.,(2008) “A Trust based middleware for providing security to ad hoc peer to peer applications”, in proceedings of the sixth annual IEEE International Conference on pervasive Computing and communications (PERCOMM’08). pp 531-536.

[AMAZ] Amazon Site. (http://www.amazon.com) World Wide Web

[ANDR04] Andersen F. et.al (2004), “An architecture concept for mobile P2P file sharing services” in proc. of the workshops at Informatik 2004 - Algorithms and Protocols for Efficient P2P Applications pp 229-233. 2004

[ANNA09] Anna-Kaisa P., Oliver E., LeBrun J., Varghese G., Diot C., (2009), “MobiClique: middleware for mobile social networking” Proceedings of the 2nd ACM workshop on Online social networks WOSN in SIGCOMM 2009. pp 49-54.

[AWER02] Awerbuch B., Holmer D., Nita-Rotaru C., and Rubens H. (2002), “An On-demand Secure Routing Protocol Resilient to Byzantine Failures”. *Proceedings of the ACM Workshop on Wireless Security*, pp. 21-30, 2002.

[BAHT10] Bahtiyar S., Cihan M., and Aglayan M. (2010), “A Model of Security Information Flow on Entities for Trust Computation”, Proceedings of the 10th IEEE International Conference on Computer and Information Technology (CIT 2010) Bradford UK, pp.803-809, 29 June -1 July, 2010.

[BALA07] Balakrishnan V., Varadharajan V., Tupakula U. and Lucs P. (2007), “Trust and Recommendations in Mobile Ad hoc Networks “, Proceedings of the Third International Conference on Networking and Services, pp 64 -70, 2007.

[BALD05] Baldoni R., Beraldi R., Cugola G., Migliavacca M., Querzoni L., "Structure-less content-based routing in mobile ad hoc networks", In Proc. of the IEEE International Conference on Pervasive Services (ICPS05). 2005.

[BANE03] Banerjee, S. Misra, A. Jihwang Yeo Agrawala, A., "Energy-efficient broadcast and multicast trees for reliable wireless communication", Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2003), pp.660 - 667, March 2003.

[BARA05] Baras J., Jiang T. (2005) "Managing Trust in Self-organized Mobile Ad Hoc Networks" in the proc. of the 12th Annual Network and Distributed System Security Symposium (NDSS) workshop, February 2005, San Diego.

[BEAC08] Beach A. et. al.(2008), “WhozThat? Evolving an Ecosystem for Context-Aware Mobile Social Networks”, IEEE Network, Vol. 22, No. 4, pp-50-55.

[BETT02] Bettstetter C., Hartenstein H., Perez-Costa H.(2002),"Stochastic Properties of the Random Waypoint Mobility Model: Epoch Length, Direction Distribution, and Cell Change Rate", Proceedings of the 5th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems, MSWiM 2002, Atlanta, Georgia, USA, pp.7-14, September 28, 2002.

[BISW05] Biswas S. and Morris R.(2005). “Exor: opportunistic multi-hop routing for wireless networks. *SIGCOMM Comput. Commun. Rev.*, Vol. 35 No.4 pp.133– 144, 2005.

[BLAZ98] Blaze M., Feigenbaum J., and Keromystis A. (1998), "The KeyNote Trust Management for Public-Key Infrastructures" Cambridge 1998 Security Protocols International Workshop, England, 1998.

[BORI01] Borisov N., Goldberg I. and Wagner D. (2001),” Interception Mobile Communications: The Insecurity of 802.11”. *Conference of Mobile Computing and Networking*, 2001.

[BROC98] Broch J., Maltz D., Johnson D., Hu Y., and Jetcheva J. (1998), "A performance comparison of multi-hop wireless ad hoc network routing protocols", In Proc. ACM International Conference on Mobile Computing and Networking (Mobicom’98), Dallas, TX, October 1998.

[BUCH02] Buchegger S. and Boudec J. (2002), “Performance analysis of the CONFIDANT protocol: Cooperation of nodes—fairness in dynamic ad-hoc networks,” in MobiHOC, IEEE, June 2002.

[BUCH04] Buchegger S. and Boudec Y. (2004), “A Robust Reputation System for P2P and Mobile Ad-hoc Networks,” in Proceedings of the 2nd Workshop on Economics of P2P Systems, June 2004.

[CAES06] Caesar M. et.al (2006) “Virtual ring routing: network routing inspired by DHTs”. In SIGCOMM ’06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 351–362, 2006

[CALE08] Caleffi, Marcello (2008) “Mobile Ad Hoc Networks: the DHT paradigm”, PhD Thesis Università degli Studi di Napoli "Federico II".

[CAOH05] Cao H. et.al. (2005), "MOBI-DIC: MOBIle DIscovery of loCal Resources in P2PWireless Network", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 28, No. 3, Special Issue on Database Issues for Location Data Management, Sept. 2005, pp. 11-18

[CARZ01] Carzaniga A. and Wolf A. L. (2001) "Content-based Networking: A New Communication Infrastructure," in *NSF Workshop on an Infrastructure for Mobile and Wireless Systems*, No. 2538 in LNCS, (Scottsdale, Arizona), pp. 59–68, Springer-Verlag, Oct. 2001.

[CAST05] Casteigts A., Chaumette S.(2005), "Dynamicity aware graph relabeling systems (da-grs), a local computation based model to describe manet algorithms," International Conference on Parallel and Distributed Computing Systems, pp. 231–236, November 2005.

[CAST06] Casteigts A. (2006), "Model driven capabilities of the da-grs model," ICAS '06: Proceedings of the International Conference on Autonomic and Autonomous Systems, p. 24, 2006.

[CAUS09] Caus T., Christmann S., Hagenhoff S. (2009) . "Development of context-aware mobile services: an approach to simplification", International Journal of Mobile Communications 2009 - Vol. 7, No.2 pp. 133 – 153.

[CERF06] Cerf V., Burleigh S., Hooke A., Torgerson L., Durst R., Scott K., Fall K., and Weiss H. (2006), "Delay-tolerant network architecture," in *Internet-draft: draft-irtf-dtnrg-arch-04.txt, DTN Research Group*, 2006.

[CHAI08] Chaintreau A., Fraigniaud P., Lebhar E.,(2008) "Opportunistic spatial gossip over mobile social networks", Proceedings of the first workshop on Online social networks WOSN in SIGCOMM 2008. pp 73-78

[CHAI09] Chaintreau A., Hui P., Crowcroft J., Diot C., Gass R., and Scott J., (2009). "Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding". Technical Report UCAM-CL-TR-617, University of Cambridge.

[CHEN08] Chen G. and Rahman F.(2008), "Analyzing Privacy Designs of Mobile Social Networking Applications", Paper presented in 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Shanghai, China, December 17-20,2008, pp 83-88.

[CHIA99] Chiang, C. et al. (1999) "On-Demand Multicast Routing Protocol". In *Proc. of IEEE WCNC*, 1999.

[CHU04] Yuechun Chu and Aura Ganz, "A Mobile Teletrauma System Using 3G Networks", IEEE Transactions on Information Technology in Biomedicine, Vol. 8, No. 4, December 2004, pp. 456-462.

[CLAU03] Clausen T. and Jacquet P. (2003), “Optimized Link State Routing Protocol (OLSR) Project”, Hipercom, INRIA, www.ietf.org/rfc/rfc3626.txt, RFC-3626, 2003.

[CONT03] Conti M. (2003). *Body, personal, and local ad hoc wireless networks*. CRC Press, Inc., 2003

[CONT05] Conti M., Gregori E., Turi G., “A cross-layer optimization of gnutella for mobile ad hoc networks”, Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing, New York, NY, USA, pp. 343-354, 2005

[CORS99] Corson M., Macker J., and Cirincione. G. (1999) Internet-based mobile ad hoc networking. IEEE Internet Computing, 3(4):63–70, 1999.

[COST06] Costa P., Musolesi M., Mascolo C., Picco G. P. (2006), “Adaptive Content based Routing for Delay tolerant Mobile Ad Hoc Networks”, Research notes RN/06/08, University College London Press, pp. 1-12.

[DALY07] Daly E. and Haahr M. (2007), “Social network analysis for routing in disconnected delay-tolerant MANETs”, in MobiHoc ’07: Proceedings of the 8th ACM int. symposium on Mobile ad hoc networking and computing, 2007.

[DALY10] Daly E., Haahr M., “The challenges of disconnected delay-tolerant MANETs” Ad Hoc Networks (8) 2. March 2010. Pp 241-250.

[DATT04] Datta A., Quarteroni S., and Aberer K. (2004), “Autonomous Gossiping: a Self-Organizing Epidemic Algorithm for Selective Information Dissemination in Mobile Ad-Hoc Networks,” Proceedings of the International Conference on Semantics of a Networked World (IC-SNW’04), Paris, Frace, June 2004, pp. 126–143.

[DAVI06] David S. and Pinch T. (2006), “Six degrees of reputation: The use and abuse of online review and recommendation systems,” First Monday, Vol. 11, March 2006.

[DECL01] DeCleene B. et al (2001), “Secure Group Communications for Wireless Networks”, MILCOM 2001.

[DELM05] Delmastro F. (2005). “From pastry to crossroad: Cross-layer ring overlay for ad hoc networks” in proc. of IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 60–64, 2005.

[DIPI03] Di Pietro R., Mancini L., Jajodia S., (2003) “Providing secrecy in key management protocols for large wireless sensor networks”, Elsevier Ad Hoc Networks, Vol. 1, No. 4, pp.455-468, 2003.

[DODGE] Dodgeball project, Internet http://www.dodgeball.com

[DTNRG] Delay Tolerant Network Research Group Internet http://www.dtnrg.org

[EAGL05] Eagle N. and Pentland A. (2005), “Social serendipity: Mobilizing social software”. IEEE Pervasive Computing, Vol.4 No.2, 2005.

[EAGL06] Eagle N. and Pentland A., (2006). “Reality mining: Sensing complex social systems,” *Personal and Ubiquitous Computing*, Vol. 10, No. 4, March 2006, pp. 255–268.

[EBAY] Ebay Internet website www.ebay.com

[ELDE09] El Defrawy K., Solis J., Tsudik G. (2009), "Leveraging Social Contacts for Message Confidentiality in Delay Tolerant Networks", 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC), Seattle, July 20- 24, 2009.

[EMULE] eMule project, Internet http://www.emule-project.net

[ETSI] HiperLAN web site

[FACE] Facebook Developer Resource, http://developers.facebook.com/resources.php

[FALL03] Fall K. (2003), "A delay-tolerant network architecture for challenged internets" Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM Press, August 2003, pp. 27–34.

[FREE01] Freebersyser J. and Leiner B. (2001). "A *DoD perspective on mobile Ad hoc networks*" Addison-Wesley Longman Publishing Co., Inc., 2001.

[GUHA04] Guha R., Kumar R., Raghavan P., and Tomkins A. (2004), "Propagation of trust and distrust," International World Wide Web Conference (WWW2004), 2004.

[GUID07] Guidec F. and Mahéo Y., "Opportunistic Content-Based Dissemination in Disconnected Mobile Ad Hoc Networks", in proc. International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM07). pp49-54, 2007.

[HAAS02] Haas Z., Pearlman M., Samar P., "The Zone Routing Protocol (ZRP) for Ad Hoc Networks", IETF MANET Working Group, INTERNET-DRAFT, July, 2002

[HAIL08] Haillot J. and Guidec F., 2008. "A Protocol for Content Based Communication in Disconnected Mobile Ad Hoc Networks", *Proceedings of 22nd International Conference on Advanced Information Networking and Applications, AINA 2008,* pp 188- 195.

[HANG08] Hang C., Wang Y., and Singh M.(2008), "An adaptive probabilistic trust model and its evaluation," in proceedings of AAMAS , 2008, Vol 3. pp. 1485–1488.

[HARR05] Harras K. Almeroth K., Belding-royer E. (2005), "Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding Schemes in Sparse Mobile Networks", Proceedings of the IFIP Netwoking (2005). Waterloo, Canada, pp 1180-1192.

[HOGI] Hogie L., Bouvry P. and Guinand F., "The MADHOC simulator", http://agamemnon.uni.lu/~lhogie/madhoc/

[HONG02] Hong X., Xu K., and Gerla M. (2002). Scalable routing protocols for mobile ad hoc networks. IEEE Network, Vol.16 No.4 pp.11–21, July 2002.

[HUBA01] Hubaux J., Buttyan L. and Capkun S. (2001) "The Quest for Security in Mobile Ad Hoc Networks" Proc. Of the ACM MobiHoc 2001, Long Beach, CA, USA, Oct. 2001.

[HUIP08] Hui P, Crowcroft J., Yoneki E.,(2008) “Bubblerap: social-based forwarding in delay tolerant networks”, Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing SIGMOBILE’08 , pp.241-250, 2008

[HUNY04] Huynh, T., Jennings, N., Shadbolt, N. (2004), “Developing an integrated trust and reputation model for open multi-agent systems”, In proceedings of the 7*th* International Workshop on Trust in Agent Societies, New York, USA (2004) pp. 65–74.

[HUNY06] Huynh T., Jennings N., and Shadbolt N.(2006), “An integrated trust and reputation model for open multi-agent systems,” Autonomous Agents and Multi-Agent Systems, Vol. 13, pp. 119–154, Sept 2006.

[HUY02] Hu Y., Perrig A., and Johnson D. (2002), “Packet Leashes: A Defense Against Wormhole Attacks inWireless Ad Hoc Networks”. *Proc. of IEEE INFORCOM,* 2002.

[HUY04] Hu Y. and Perrig A. (2004), “A Survey of Secure Wireless Ad Hoc Routing”. *IEEE Security & Privacy,* pp. 28-39, 2004.

[HYYT06] Hyytiä E., Lassila P., and Virtamo J. (2006)“Spatial Node Distribution of the Random Waypoint Mobility Model with Applications”. IEEE Transactions on Mobile Computing, Vol. 5, No. 6, pp. 680-692.

[IBMPCC] IBM’s PCC http://www.zurich.ibm.com/pcc/

[IBNK04] Ibnkahla M. et.al. (2004), “High-Speed Satellite Mobile Communications: Technologies And Challenges”, Proceedings of the IEEE, Vol. 92, No. 2, February 2004.

[IEEE 802.11] WLAN web site. http:// www.ieee802.org/11/

[IEEE 802.15] WPAN web site. www.ieee802.org/15/pub/TG4.html

[IEEE 802.16] WMAN web site. ieee802.org/16/

[ILYA03] Ilyas M. (2003), *The Handbook of Ad Hoc Wireless Networks,* CRC Press, 2003.

[JACQ01] Jacquet P., Muhlethaler P., Clausen T., Laouiti A., Qayyum A., Viennot L., (2001) “Optimized Link State Routing Protocol for Ad Hoc Networks”, Multi Topic Conference, 2001. IEEE INMIC 2001.

[JAIN04] Jain S., Fall K., Patra R.,(2004) “Routing in a delay tolerant network”, in SIGCOMM’04 Computer Communication Review Vol.34 No.4 (2004) pp.145–158. October 2004.

[JEAD05] Jea D., Somasundara A., and Srivastava M (2005). “Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks”. In Proceedings of DCOSS'05, pp. 244-257, April 2005.

[JOHN96] Johnson, D., and Maltz, D. 1996. “Dynamic Source Routing in ad-hoc wireless networks”, *Mobile Computing* (1996), pp.152–181.

[JURC03] Jurca, R., & Faltings, B. (2003). Towards incentive-compatible reputation management. In R. Falcone, S. Barber, L. Korba, & M. Singh, (Eds.), *Trust, reputation and security: theories and practice*. Vol. 2631 of Lecture Notes in AI (pp. 138–147). Springer-Verlag, Berlin, Heidelberg.

[KANT08] Kantere V. et.al. (2008), "A Simulator for a Mobile P2P Database Environment," The Ninth International Conference on Mobile Data Management (MDM08), 2008. pp.223-224.

[KARY02]Karygiannis T. and Owens L. (2002), “Wireless Network Security-802.11, Bluetooth and Handheld Devices. National Institute of Standards and Technology”. Technology Administration, U.S Department of Commerce, *Special Publication pp.*800-848, 2002.

[KERR09] Kerr R. and Cohen R. (2009), “Smart cheaters do prosper: Defeating trust and reputation systems,” in AAMAS ’09. Budapest, Hungary: ACM, 2009.

[KLEM03] Klemm A. et.al.(2003) “A special-purpose peer-to-peer file sharing system for mobile ad hoc networks”. Proc. of IEEE VTC, October 2003.

[KONG02] Kong J. et al (2002), “Adaptive Security for Multi-layer Ad-hoc Networks”, Special Issue of Wireless Communications and Mobile Computing, John Wiley InterScience Press, 2002.

[LEEM07] Lee M. and Yoe H. (2007), “Comparative Analysis and Design of Wired and Wireless Integrated Networks for Wireless Sensor Networks,” Software Engineering Research, Management & Applications, 2007. SERA 2007. 5th ACIS International Conference on, 2007, pp. 518-522.

[LEGU06] Leguay, J., Friedman, T. and Conan, V., 2006. “Evaluating Mobility Pattern Space Routing for DTNs”, *Proceedings of the IEEE INFOCOM 2006,* April 2006, Vol. 5, pp 2540-2549.

[LEHR02] Lehr W., Mcknight L. (2002), “Wireless internet access: 3G vs. WIFI”, Telecommunication Policy Vol.27 (2002) pp.351–370.

[LEWI85] Lewis (1985), “Trust as a social reality,” Social Forces, Vol. 63, pp. 967–985, June 1985.

[LIAN07] Lian, Q., Zhang, Z., Yang, M., Zhao, B.Y., Dai, Y., Li, X. (2007) An empirical study of collusion behavior in the Maze P2P file-sharing system. In: proc of the 27th International Conference on Distributed Computing Systems (ICDCS 2007), IEEE Computer Society (2007) pp.56.

[LIFE] Lifeguard Monitoring system, http://lifeguard.stanford.edu

[LIH07] Li H.and Singhal M. (2007), “Trust Management in Distributed Systems,” Computers, Vol. 40, No.2, Feb. 2007, pp. 45-53.

[LIJ08] Li J., Li R., and Kato J. (2008), “Future Trust Management Framework for Mobile Ad Hoc Networks: Security in Mobile Ad Hoc Networks,” IEEE Communications Magazine, Vol. 46, No. 4, Apr. 2008, pp. 108-114.

[LIMC08] Lim Choi Keung S., Griffiths, N.(2008) “Towards improved partner selection using recommendations and trust”. In Falcone, R., et al., eds.: Trust in Agent Societies

(TRUST 2008). Volume 5396 of Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg (2008) pp.43–64.

[LIMC09] Lim Choi Keung S. and Griffiths N. (2009) "Building a Trust-based Social Agent Network", in *Proceedings of the 12th International Workshop on Trust in Agent Societies*, pp. 68-79, 2009.

[LIND03] Lindgren A., Doria A., and Schelen O. (2003), "Probabilistic routing in intermittently connected networks," Mobile Computing and Communications Review, Vol. 7, No. 3, July 2003.

[LIUB08] Liu B., Chou C., Misra A. and Jha S., "Rate-diversity and resource-aware broadcast and multicast in multi-rate wireless mesh networks", Journal of Mobile Networks and Applications, Vol. 13, No.1-2, pp.38-53, 2008.

[LOUW03] Lou W. and Fang Y. (2003)," A Survey of Wireless Security in Mobile Ad Hoc Networks: Challenges and Available Solutions". *Ad Hoc Wireless Networks,* edited by X. Chen, X. Huang and D. Du. Kluwer academic Publishers, pp. 319-364, 2003.

[LUGA07] Lugano G. and Saariluoma P., (2007). "To Share or not to share: Supporting the user decision in Mobile Social Software applications," *Proceedings of the International User Modelling conference (UM 2007; Corfu, Greece, 25–29 July)*. Lecture Notes in Computer Science, volume 4511. Berlin: Springer. pp. 440–444

[LUOA09] Luoa J. ,Liub X., Fana M. (2009), "A trust model based on fuzzy recommendation for mobile ad-hoc networks", Computer Networks, Volume 53, Issue 14, 18 September 2009, pp. 2396-2407

[LUOY08] Luo Y., Wolfson O., and Xu B. (2008), "Mobile Local Search via P2P Databases", Proc. of Portable 2008: the 2nd IEEE International Interdisciplinary Intersociety Conference on Portable Information Devices (PIDs), Garmisch-Partenkirchen, Germany, Aug. 2008.

[MAHE08] Maheo Y., Said R. and Guidec F., "Middleware Support for Delay-Tolerant Service Provision in Disconnected Mobile Ad Hoc Networks", In proc. of IEEE International Symposium on Parallel and Distributed Processing, (IPDPS 2008). 14-18 April 2008.

[MARI06] Marias S., Georgiadis P., Flitzanis D., and Mandalas K.(2006), "Cooperation enforcement schemes for manets: a survey: Research articles," Wireless Communications and Mobile Computing, Vol. 6, No. 3, pp. 319–332, 2006.

[MARS94] Marsh, S. (1994), "Formalizing trust as computational concept", PhD Thesis, department of computer science, University of Sterling, 1994.

[MART03] Martin J., Jones M. and Shenoy R. (2003). "Towards a design framework for wearable electronic textiles", 7th IEEE International Symposium on Wearable Computers, pp. 190– 199, 2003.

[MASU09] Masoudifar M., 2009. "A review and performance comparison of QoS multicast routing protocols for MANETs", *Elsevier Ad Hoc Networks*, Vol. 7, No. 6, August 2009, pp. 1150-1155.

[MERU07] Merugu S., Ammar M, and Zegura E. (2007), "Routing in space and time in networks with predictable mobility," *Technical report, GIT-CC-04-07, College of Computing, Georgia Tech. Georgia USA.*

[MERW07] Merwe J., Dawoud D., and McDonald S. (2007), "A survey on peer-to-peer key management for mobile ad hoc networks," ACM Comput. Surv.,Vol. 39, No. 1, pp. 1-11, 2007.

[MIKL07] Miklas A., Gollu K., Chan K., Saroiu G., Gummadi K., and de Lara E. (2007) "Exploiting social linteractions in mobile systems". Proceedings of UbiComp 2007.

[MOTT08] Mottola L., Cugola G., Picco G., "A Self-Repairing Tree Topology Enabling Content-Based Routing in Mobile Ad Hoc Networks", IEEE Transactions on Mobile Computing, Vol.7, No.8, August 2008, pp. 946-960.

[MUIL02] Mui L., Mohtashemi M., and Halberstadt A.(2002), "A computational model of trust and reputation for e-businesses," in HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 7. IEEE Computer Society, 2002, pp. 188.

[MUSO08] Musolesi M. and Mascolo C., (2008) "A Framework for Multiregion Delay Tolerant Networking", Proceedings of the 2008 ACM workshop on Wireless networks and systems for developing regions (WiNSDR'08) in SIGMOBILE 2008. pp 37-42.

[MUSO09] Musolesi M. and Mascolo C.(2009), "CAR: Context-aware Adaptive Routing for Delay Tolerant Mobile Networks". IEEE Transactions on Mobile, Vol 8. No 2. Feb 2009. pp 246-260.

[MUTH05] Muthusamy V., Petrovic M., Jacobsen H., "Effects of Routing Computations in Content Based Routing Networks with Mobile Data Sources", in proc. of ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM05), pp 103-116, August 28–September 2, 2005.

[NICO06] Nicolai T., Yoneki E., Behrens N., and Kenn H.. (2006) "Exploring social context with the wireless rope". In Proceedings of the OTM Workshop MONET, 2006.

[OLIV03] Oliveira L., Siqueira I., and Loureiro A. (2003). "Evaluation of ad-hoc routing protocols under a peer to peer application", Proc. of IEEE WCNC, March 2003.

[OPER] Opera Mini website http://www.opera.com/mini/

[PAPA03] Papadimitratos P. and Haas Z. (2003), "Secure Data Transmission in Mobile Ad Hoc Networks". Proc. of the 2003 ACM Workshop on Wireless Security, pp. 41-50, 2003.

[PATW05] Patwardhan A. et al.(2005) "Active Collaborations for Trustworthy Data Management in Ad Hoc Networks", Proceedings of the 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems. November, 2005.

[PEIG00] Pei G., Gerla M. and Hong X. (2000), "LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility", Proceedings of IEEE/ACM MobiHOC 2000, Boston, MA, Aug. 2000, pp. 11-18.

[PELU06] Pelusi L., Passarella A., and Conti M. (2006). "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks" IEEE Communications Magazine, Vol.44, No.11 pp.134–141, November 2006.

[PERK01] Perkins C. (2001), *Ad Hoc Networks,* Addison-Wesley, 2001.

[PERK94] Perkins, C. E., and Bhagwat, P. (1994), "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers". Proc. of ACM SIGCOMM (October 1994), Vol. 2, ACM Press, pp. 234–244.

[PIET09] Pietilainen A. and Diot C.,(2009) "Experimenting with opportunistic communications" in MobiArch'09: The 4th ACM International Workshop on Mobility in the Evolving Internet Architecture, 2009.

[PIYA08] Piyatumrong A. and Bouvry P., Guinand F., Lavangnananda K. (2008), "Trusted Spanning Trees for Delay Tolerant Mobile Ad Hoc Networks", Proceedings of IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2008. EUC '08. Shanghai, China, Vol.2, pp. 293-299, 17-20 Dec. 2008

[PUBK] Public-Key Infrastructure (X.509), http://www.ietf.org/html.charters/pkix-charter.html

[PUCH06] Pucha H., Das S., and Hu Y.(2006). "Imposed route reuse in ad hoc network routing protocols using structured peer-to-peer overlay routing". IEEE Transactions on Parallel and Distributed Systems, Vol.17 No.12 pp.1452–1467, 2006.

[QURE10] Qureshi B., Min G., Kouvatsos D., "Collusion prevention in FIRE+ Trust & Reputation model" in the proceedings of the 10th IEEE International Conference on Scalable Computing and Communications (Scalcom2010), Bradford UK, pp.2548-2555, 29 June – 1 July 2010.

[RAEN05] Raento M. and Oulasvirta A., (2005). "Privacy management for social awareness applications," Proceedings of 1st Workshop on Context Awareness for Proactive Systems — CAPS 2005, Helsinki, Finland, pp. 105–114.

[RAMC04] Ramchurn S.D., Huynh D. and Jennings N. R. (2004) "Trust in multi-agent systems," Knowledge Engineering. Review, Vol. 19,No. 1, pp. 1–25, 2004.

[REDI06] Redi J., et.al. (2006) "JAVeLEN C An Ultra-Low Energy Ad hoc Wireless Network" BBN Technologies, Cambridge, MA 2006

[REPA05] Repantis T., Kalogeraki V. (2005) "Data Dissemination in Mobile PeertoPeer Networks" Proceedings of MDM 2005 Ayia Napa Cyprus.

[REPA06] Repantis T. and Kalogeraki V. (2006), “Decentralized Trust Management for Ad-hoc Peer-to-peer Networks”. In Proceedings of the 4th International Workshop on Middleware for Pervasive and Ad- Hoc Computing (MPAC’06) in conjunction with MIDDLEWARE 2006. November, 2006.

[RESN02] Resnick, P., & Zeckhauser, R. (2002). Trust among strangers in internet transactions: Empirical analysis of eBay’s reputation system. InM. R. Baye, (Ed.), *The economics of the internet and e-commerce*. Vol. 11 of Advances in Applied Microeconomics. Elsevier Science.

[RIPE02] Ripeanu M., Iamnitchi A., and Foster I.(2002). “Mapping the gnutella network”. IEEE Internet Computing, Vol.6 No.1 pp.50–57, 2002.

[ROWS01] Rowstron A. and Druschel P. (2001). “Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems”. In IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), pp. 329–350, November 2001.

[ROYE99] Royer E. and Toh C. K., (1999), “A review of current routing protocols for Ad Hoc mobile wireless networks”, IEEE Personal Communications, Vol. 6, No. 2, pp. 46-55.

[SABA01] Sabater J. and Sierra C. (2001), “Regret: A reputation model for gregarious societies,” in Fourth Workshop on Deception Fraud and Trust in Agent Societies, 2001, pp. 61–70.

[SABA05] Sabater J., Sierra C.(2005), “Review on computational trust and reputation models,” Artiicial. Inteligence review., Vol. 24, No. 1, pp. 33–60, 2005.

[SALE09] Salehi-Abari A. and White T. (2009), “Towards con-resistant trust models for distributed agent systems,” in IJCAI ’09: Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence, 2009.

[SARA06] Sarafijanovic-Djukic N. and Grossglauser M. (2006), “Island Hopping: Efficient Mobility Assisted Forwarding in Partitioned Networks,” in Proceedings of IEEE SECON’06, September 2006.

[SAZI02] Sanzgiri K., Dahill B., Levine B., Shields C., and Belding-Royer E. (2002), A Secure Routing Protocol for Ad Hoc Networks. Proc. of IEEE International Conference on Network Protocols (ICNP), pp. 78-87, 2002

[SCHI00] Schillo, M., Funk, P., Rovatsos, M. (2000), “Using trust for detecting deceitful agents in artificial societies”, Applied Artificial Intelligence, Special Issue on Trust, Deception, and Fraud in Agent Societies Vol.14 No.8 (2000) pp.825–848.

[SELC08] Selcuk A., Uzun E., and Pariente M. (2008), “A reputation-based trust management system for p2p networks”, International Journal of Network Security, Vol. 6, No. 3, pp. 235–245, 2008.

[SERE07] Seredynski M, Bouvry P., and Klopotek M. (2007), “Preventing selfish behavior in ad hoc networks,” Congress on Evolutionary Computation (CEC 2007), pp. 3554 – 3560, IEEE Computer Society, September 2007.

[SHAH03] Shah R., Roy S., Jain S., and Brunette W. (2003), “Data MULEs: Modelling a Three-Tier Architecture for Sparse Sensor Networks”. Technical Report IRS-TR-03-001, Intel Corporation, January 2003

[SINH01] Sinha P., Sivakumar R., Bharghavan V. (2001), “Enhancing Ad Hoc Routing with Dynamic Virtual Infrastructures” Proceeding of INFOCOM 2001 pp.1763-1772.

[SIVA99] Sivakumar R., Sinha P., Bharghavan V.” CEDAR: a Core-Extraction Distributed Ad Hoc Routing Algorithm” Proceedings of INFOCOM 1999, pp.202-209.

[SMH] http://www.rochester.edu/pr/Review/V64N3/feature2.html

[SONG05] Song S., Hwang K., and Zhou R. (2005), “Trusted p2p transactions with fuzzy reputation aggregation”, IEEE Internet Computing, Vol. 9, No. 6, pp. 24–34, Dec 2005.

[SPKI] Simple Public Key Infrastructure (SPKI), http://www.ietf.org/html.charters/spkicharter.html.

[SPYR05] Spyropoulos T., Psounis K., and Raghavendra C., “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” Proceeding of WDTN’05. New York, NY, USA: ACM Press, 2005, pp. 252–259.

[SRIV06] Srivatsa, M., Liu, L. (2006) “Securing decentralized reputation management using TrustGuard”. Journal of Parallel and Distributed Computing Vol.66 No.9 (2006) pp.1217–1232

[STAL02] Stallings W. (2002), *Wireless Communication and Networks*, Pearson Education, 2002.

[SUJ07] Su J., Scott J., Hui P., Crowcroft J., de Lara E., Diot C. , Goel A., Lim M., and Upton E. (2007), "Haggle: Seamless networking for mobile applications," in proceedings of Ubiquitous Computing (UbiComp 2007), pp. 391-408. 2007.

[SUKU06] Sukumaran S. and Blessing R.(2006), “Reputation based localized access control for mobile ad-hoc networks,” in ADHOCNOW, Lecture Notes in Computer Science, pp. 197–210, Springer, 2006.

[TAKE08] Takeshita K., Sasabe M., and Nakano H.(2008). “Mobile p2p networks for highly dynamic environments”. In PERCOM’08: Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications, pp. 453–457, 2008.

[TANG04] Tang C., Xu Z., and Dwarkadas S. (2004) “Peer-to-peer information retrieval using self-organizing semantic overlay networks”. In Proc. of ACM SIGCOMM, August 2004.

[TEAC05] Teacy W., Patel J., Jennings N., and Luck M. (2005), “Coping with inaccurate reputation sources: experimental analysis of a probabilistic trust model,” in AAMAS ’05. New York, NY, USA: ACM, 2005, pp. 997–1004.

[TOMI06] Tomiyasu H., Maekawa T., Hara T., Nishio S., (2006) “Profile-based Query Routing in a Mobile Social Network”, Proceedings of the 7th International Conference on Mobile Data Management (MDM’06), pp-105-109.

[TWITT] Twitter website, Internet http://www.twitter.com

[VAHD00] Vahdat A. and Becker D. (2000), “Epidemic Routing for Partially Connected Ad Hoc Networks,” Department of Computer Science, Duke University, Tech. Rep. CS-2000-06, 2000.

[VAHD02] Vahdat A. and Becker D. (2002), “Epidemic routing for partially-connected ad hoc networks,” *Technical report, Duke University*, 2002.

[WAN04] Wan P., Alzoubi K. and Frieder O. (2004). “Distributed construction of connected dominating set in wireless ad hoc networks”. Mobile Network. Applications, Vol.9 No.2 pp.141–149, 2004.

[WANG08] Wang, J.C., Chiu, C.C.(2008) “Recommending trusted online auction sellers using social network analysis”. Expert Systems with Applications Vol.34 No.3 (2008) pp.1666–1679.

[WIES00] Wieselthier, J., Nguyen, G., Ephremides, A., “On the construction of energy-efficient broadcast and multicast trees in wireless networks", Proceedings of 19th IEEE INFOCOM, Vol.2, pp. 585 - 594. 2000.

[WOLF09] Wolfson O. (2009) "MOBI-DIK: An Approach to Querying Data in a Mobile Ad Hoc Network", Proc. of the Workshop on Research Directions in Situational-aware Self-managed Proactive Computing in Wireless Adhoc Networks,, St. Louis, MO, Mar. 2009.

[WUJ05] Wu J. (2005), “Peer-to-Peer Overlay Abstractions in MANETs”, Handbook on Theoretical and Algorithmic Aspects of Sensor Ad Hoc Wireless, and Peer-to-Peer Networks, Auerbach Publications, pp 857-874, 2005.

[XUB04] Xu B. and Wolfson O. (2004), "Data Management in Mobile P2P Networks", Springer Verlag Lecture Notes in Computer Science, Proc. of the 2nd International Workshop on Databases, Information Systems, and P2P Computing (DBISP2P'04), Toronto, Canada, Aug 2004.

[XUE09] Xue J., Li J., Cao Y., and Ji F. (2009), “Advanced PROPHET Routing in Delay Tolerant Network”, In Proc. of International Conference on Communication Software and Networks, China, Feb 27-28 2009, pp 411-413.

[XUK03] Xu K., Hong X., Gerla M. (2003) “Landmark routing in ad hoc networks with mobile backbones” Journal of Parallel and Distributed Computing archive Volume 63 , Issue

2, Special issue on Routing in mobile and wireless ad hoc networks, pp. 110 - 122, 2003

[YANG04] Yang H., Luo H., Ye F., Lu S., and Zhang L. (2004), “Security in Mobile Ad Hoc Networks: Challenges and Solutions”. *IEEE Wireless Communications*, pp. 38- 47, 2004.

[YANG06] Yang G. (Ed.), *Body Sensor Networks*, Springer-Verlag, March 2006.

[YAOJ07] Yao-Jen C., Hung-Huan L., Li-Der C., Yen-Wen C., Haw-Yun S. (2007), “A General Architecture of Mobile Social Network Services”, in proceedings of the International Conference on Convergence Information Technology, pp- 151 - 156, 21-23 Nov. 2007

[YIS04] Yi S. and Kravets R. (2004), “Composite Key Management for Ad Hoc Networks”. *Proc. of the* 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous’04), pp. 52-61, 2004.

[YONE04] Yoneki E. and Bacon J., (2004) “An adaptive approach to content-based subscription in mobile ad hoc networks”, Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW04), pp. 92-97.

[YUB00] Yu B. and Singh M. (2000), “A social mechanism of reputation management in electronic communities,” in CIA ’00: Proceedings of the 4th International Workshop on Cooperative Information Agents IV, The Future of Information Agents in Cyberspace. Springer-Verlag, 2000, pp. 154–165.

[YUB03] Yu B. and Singh M. (2003), “Detecting deception in reputation management,” in AAMAS ’03. New York, NY, USA: ACM, 2003, pp. 73–80

[YUNF07] Yunfang F. (2007), “Adaptive Trust Management in MANETs,” Proc. 2007 Int’l Conf. on Computational Intelligence and Security, Harbin, China, 15-19 Dec. 2007, pp. 804-808.

[ZACH00] Zacharia, G., Maes, P.(2000), “Trust management through reputation mechanisms”, Applied Artificial Intelligence Vol.14 No.9 (2000) pp.881–907.

[ZHAN07] Zhang Z., Zhang Q. (2007), “Delay/disruption tolerant mobile *ad hoc* networks: latest developments”, Wireless Communications and Mobile Computing, Vol. 7 No.10, pp.1219 - 1232, 2007.

[ZHAO04] Zhao V., Ammar M., and Zegura E. (2004), “A message ferrying approach for data delivery in sparse mobile ad hoc networks,” in Proceedings of MobiHoc’04, May 2004.

[ZHAO09] Zhao H., Li X. (2009), "VectorTrust: Trust Vector Aggregation Scheme for Trust Management in Peer-to-Peer Networks", Proceedings of ICCCN 2009. San Francisco, CA, pp 1 – 6, 3-6 August 2009.

[ZHOK07] Zhou R. and Hwang K. (2007), “Gossip-based reputation management for unstructured peer-to-peer networks”, IEEE Trans. on Knowledge and Data Engineering, Jan 2007.

[ZHOU00] Zhou H., Singh S. (2000), “Content based multicast (CBM) in ad hoc networks”, Proceedings of the First Annual Workshop on Mobile and Ad Hoc Networking and Computing, MobiHOC 2000, Boston, MA, USA. pp 51-60.

[ZHOU07] Zhou R. (2007), “Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing”, IEEE Transactions on Parallel and Distributed Systems, Vol. 18, No. 5, May 2007.

[ZHOU08] Zhao H. and Li X. (2008), “H-trust: A robust and lightweight group reputation system for p2p desktop grid”, Proc. of the 2nd Intl Workshop on Trust and Reputation Management in Massively Distributed Computing Systems (TRAM 2008) in conjunction with the 28th IEEE ICDCS, Beijing, China, pp. 235-240, 2008.

[ZHOU99] Zhou L. and Haas Z. (1999), "Securing Ad Hoc Networks," IEEE Network Magazine, Vol. 13, No. 6, Nov 1999.

[ZHU07] Zhu H. and Lu K..(2007) “Resilient opportunistic forwarding: Issues and challenges”. Military communications Conference, 2007. MILCOM 2007. *IEEE*, pp. 1–7, Oct. 2007.

[ZIMM95] Zimmermann P. (1995), “The Official PGP User’s Guide”, MIT Press, June 1995.

[ZIVN06] Ziv, N.D., Mulloth B., (2006). “An Exploration on Mobile Social Networking: Dodgeball as a Case in Point, Proceedings of the International Conference on Mobile Business, ICMB '06. 26-27 June 2006 Page(s):21 – 29.

[ZOUR05] Zouridaki C., Mark B., Hejmo M., and Thomas R. (2005), “A Quantitative Trust Establishment Framework for Reliable Data Packet Delivery in MANETs,(2005)” in In Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN’05), pp. 1–10, November 2005

[ZOUR06] Zouridaki C., Mark B., Hejmo M., Thomas R. (2006), “Robust cooperative trust establishment for MANETs”, Proceedings of the fourth ACM workshop on security of ad hoc and sensor networks, October 30-30, 2006, Alexandria, Virginia, USA.

Appendix A

Random Waypoint Mobility Model (RWP) and Madhoc Simulator

The modelling of the movement behaviour of the stations is an important building block in simulation–based studies of mobile ad hoc networks. Mobility models are needed in the evaluation of protocols for medium access, power management, leader election, routing, and so on. The choice of the mobility model and its parameters has a significant influence on the obtained simulation results.

A very popular and frequently used mobility model in ad hoc networking research is the random waypoint mobility model (RWP). It is a simple and straightforward stochastic model that describes the movement behaviour of a mobile network node in a two–dimensional system area as follows:

- The initial positioning of the nodes is typically taken from a uniform distribution. The nodes are typically placed in a square or a circular (disc) area.
- A node randomly chooses a destination point in the area and moves with constant speed to this point.
- After waiting a certain pause time, it chooses a new destination, moves to this destination, and so on.
- The pause time durations are independent and identically distributed random variables.

The most common problem with simulation studies using random waypoint model is a poor choice of velocity distribution. e.g., uniform distribution U (0,V_{max}). Such velocity distributions lead to a situation where at the stationary state each node stops moving. In order to avoid this, the velocity distribution should be such that $\frac{1}{E[\frac{1}{V}]} > 0$.

A variant of RWP is the Random waypoint on the border (RWPB). In RWPB, the waypoints can be uniformly distributed on the border of the domain. The spatial node density resulting from RWPB model is quite different from the RWP model. Stochastic properties of the random waypoint mobility model can be found in [BETT02].

Madhoc [HOGI] is a discrete-time mobile ad hoc network (MANET) simulator targeting the investigation of mobile ad hoc networks in metropolitan environments. It offers the possibility to implement new environments, new mobility schemes and new applications. Madhoc is freely distributed under GPL License and is written in Java programming language. For the purpose of simulation of various techniques presented in this thesis, Madhoc was extensively used. Many modules were modified and / or re-written in Java to cope with various parameters of experiments carried out in this study. Here we briefly present the basic overview of the simulation tool.

Simulation Model

Mobile phones and Personal Digital Assistants (PDA) nowadays are capable of storing data, processing information as well as establish communication using popular ad hoc communication channels such as Bluetooth and Wi-Fi. The compact size and various embedded functionalities of these devices permit connectivity while a user is on the move. Comparatively, laptop computers are much more powerful but when switched on, these tend to be used in a stationary mode (sitting down). Madhoc supports Wi-Fi (IEEE802.11b) and Bluetooth protocol. Although these protocols are not modelled in detail on the physical and MAC layers of a network, these are represented in terms of the following.

- *Bandwidth*: The bandwidth is shared by all communicating devices operating on a common media. All devices have the same chance to send/receive data.
- *Range of coverage:* Defines the maximum distance to/from which the devices can receive data;
- *Packet size*: Transmitted data is organized into packets. Packets can be transmitted over the network, for a given protocol.
- *Data transfer cost*: Defines the price for transmitting one byte over the network. Madhoc defines several basic cost models.

Communication model and changes made to Madhoc

Madhoc models the physical and MAC layers only in terms of available bandwidth, signal power and packet size. No clear difference is made in the implementation of these two layers. Moreover the current implementation of Madhoc does not consider networking layer. As a consequence multi-hop networking is unachievable. Furthermore trust management methods studied in this thesis are not implemented in this simulator. Due to these limitations in the Madhoc implementation, the original classes were modified to include further details necessary for implementation of the following:

- Routing information: Each node now carries a table containing paths to the next hop for a destination. These tables are updated and dead routes are removed periodically.
- Node's memory/repository: Each node also carries a limited size of repository (storage area). This is currently implemented with variables identifying the file names, file types, file size, total size, size left etc. If a file cannot be stored due to limited storage availability it can be replaced by a file with the least time to live identifier (ttl) as described in chapter 4.
- Content generation (files with profile types): Files (file identifiers) are generated globally or locally with respective parameters having different file types and sizes.
- History module: Each node maintains history of interactions with other nodes. This is stored as variables with fixed values including trust rating values, ratings received, positive and negative interactions and confidence values.
- Interaction values: Every transmission increases or decreases interaction values. These are used to implement trust ratings.
- Ratings lists: Nodes maintain ratings lists containing trust ratings for other nodes. These lists are used to generate direct and witness trust and reputation values.
- Direct and witness trust values: Direct and witness trust values are passed to other nodes as messages.
- Policies: Various policies described in chapter 7 are implemented and embedded in nodes.
- Messages: Madhoc does not define or implement messages as such. However the delay in transmission of message is calculated by the communication medium and size of message. A module was written to interact with Madhoc communication class to include further parameters in a message.
- Random number generator: This module was modified to generate random number based on various distributions such as normal, uniform, exponential, hyper-exponential distributions.

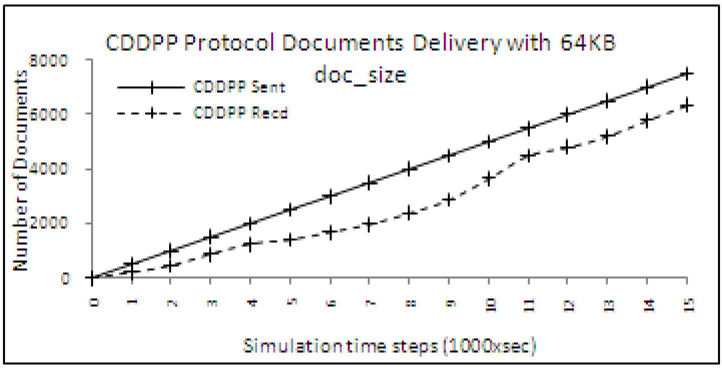
Mobility Scenarios

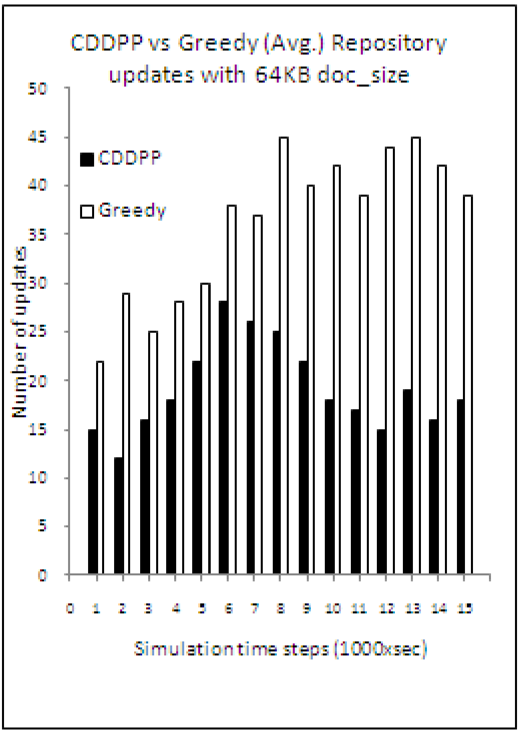
Madhoc defines various mobility models. Random waypoint mobility model (RWP) was extensively used in this thesis. Various mobility scenarios are implemented in the tool. The mobility scenarios defined in chapter 6 include Campus, Shopping Mall and City Street.

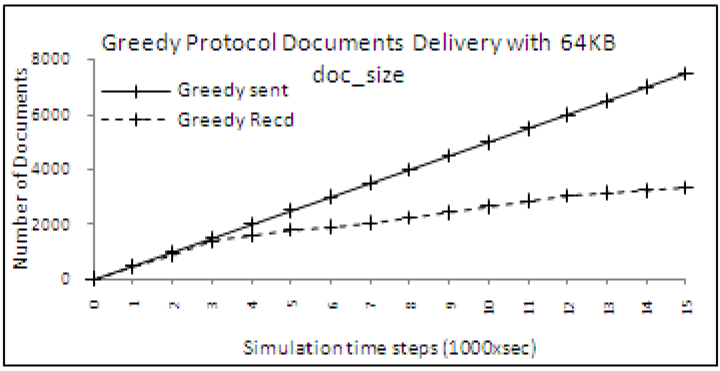
Chapter 4 Results for CDDPP Protocol

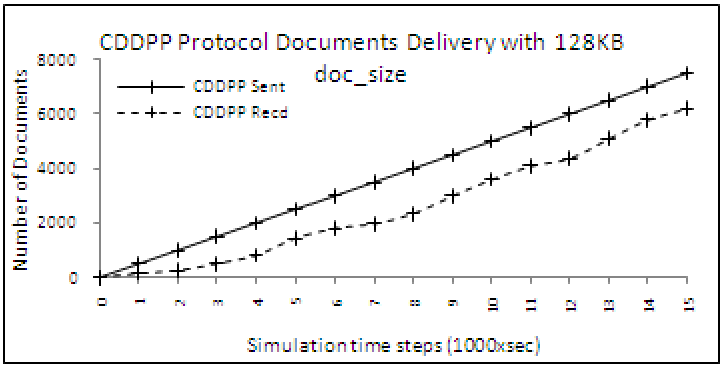
List of Interest Profiles (32 Matching keywords) used in simulations

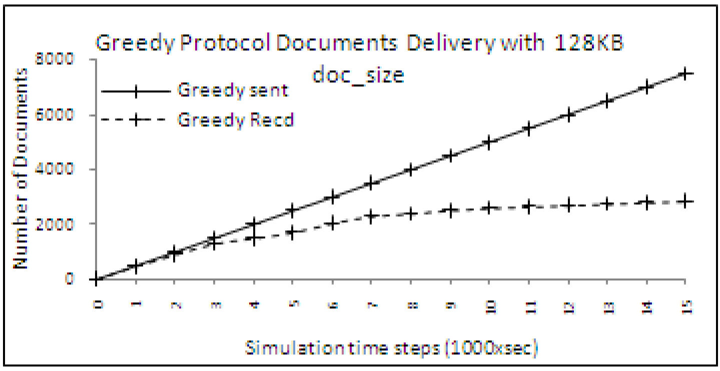
A0	B0	C0	D0
A1	B1	C1	D1
A2	B2	C2	D2
A3	B3	C3	D3
A4	B4	C4	D4
A5	B5	C5	D5
A6	B6	C6	D6
A7	B7	C7	D7

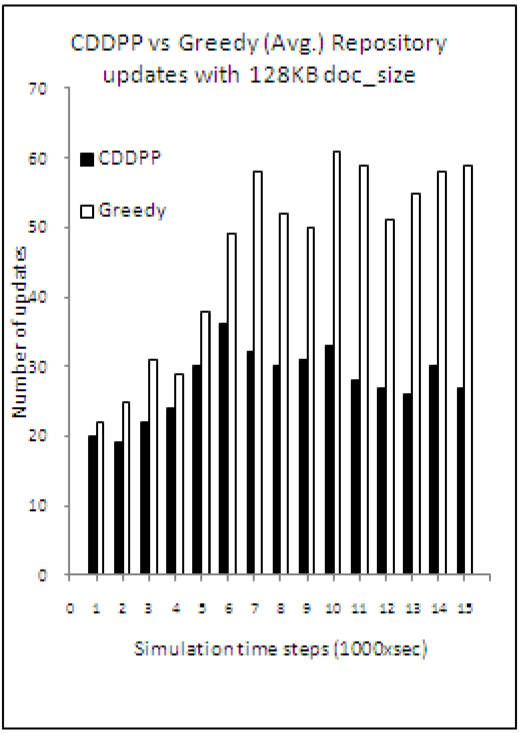

CDDPP Protocol Documents Delivery with 64KB doc_size

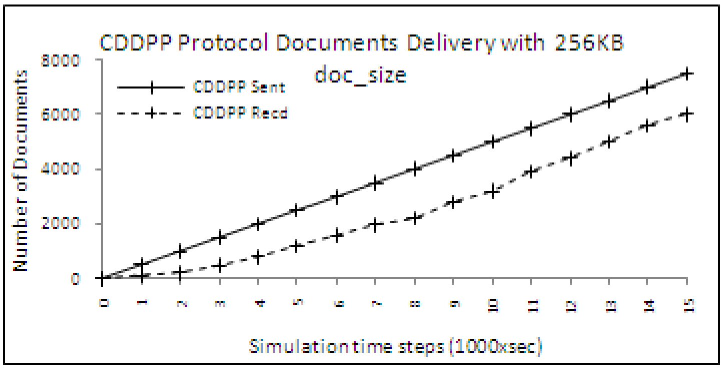

CDDPP vs Greedy (Avg.) Repository updates with 64KB doc_size

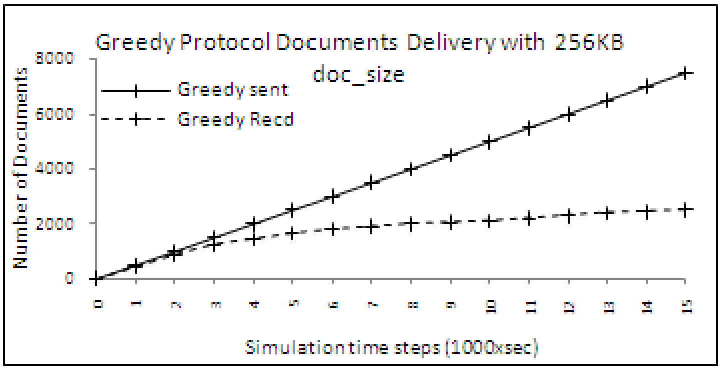

Greedy Protocol Documents Delivery with 64KB doc_size

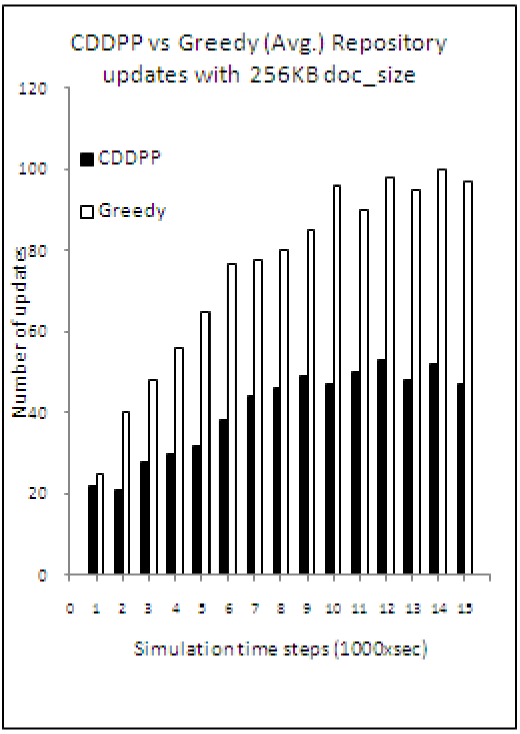

CDDPP Protocol Documents Delivery with 128KB doc_size

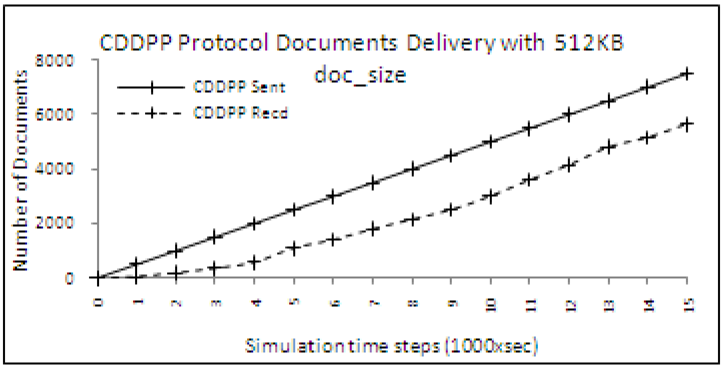

Greedy Protocol Documents Delivery with 128KB doc_size

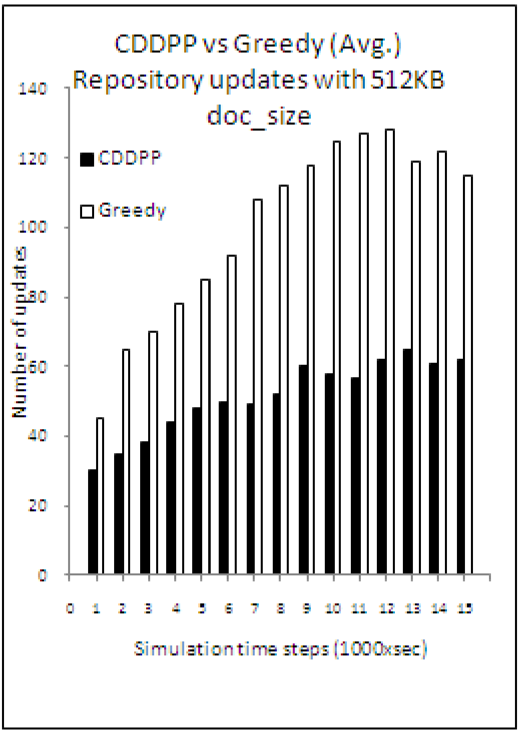

CDDPP vs Greedy (Avg.) Repository updates with 128KB doc_size

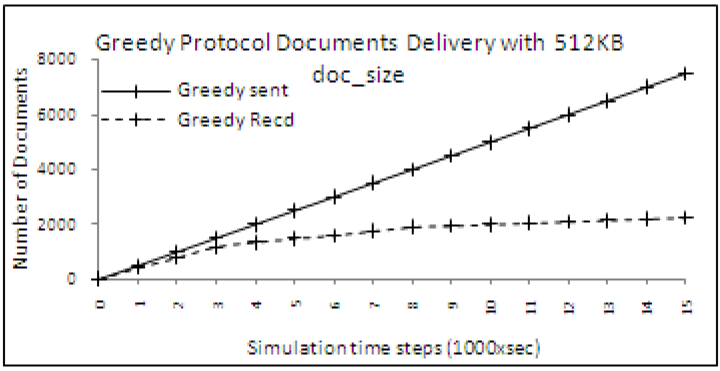

CDDPP Protocol Documents Delivery with 256KB doc_size


Greedy Protocol Documents Delivery with 256KB doc_size


CDDPP vs Greedy (Avg.) Repository updates with 256KB doc_size



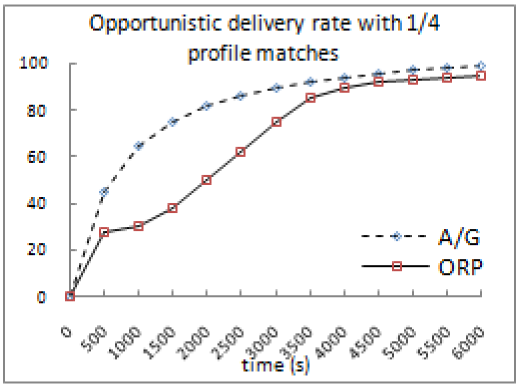


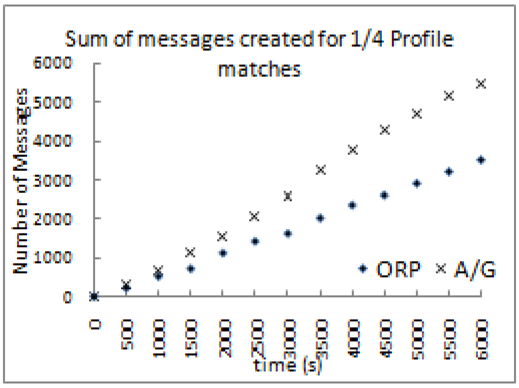


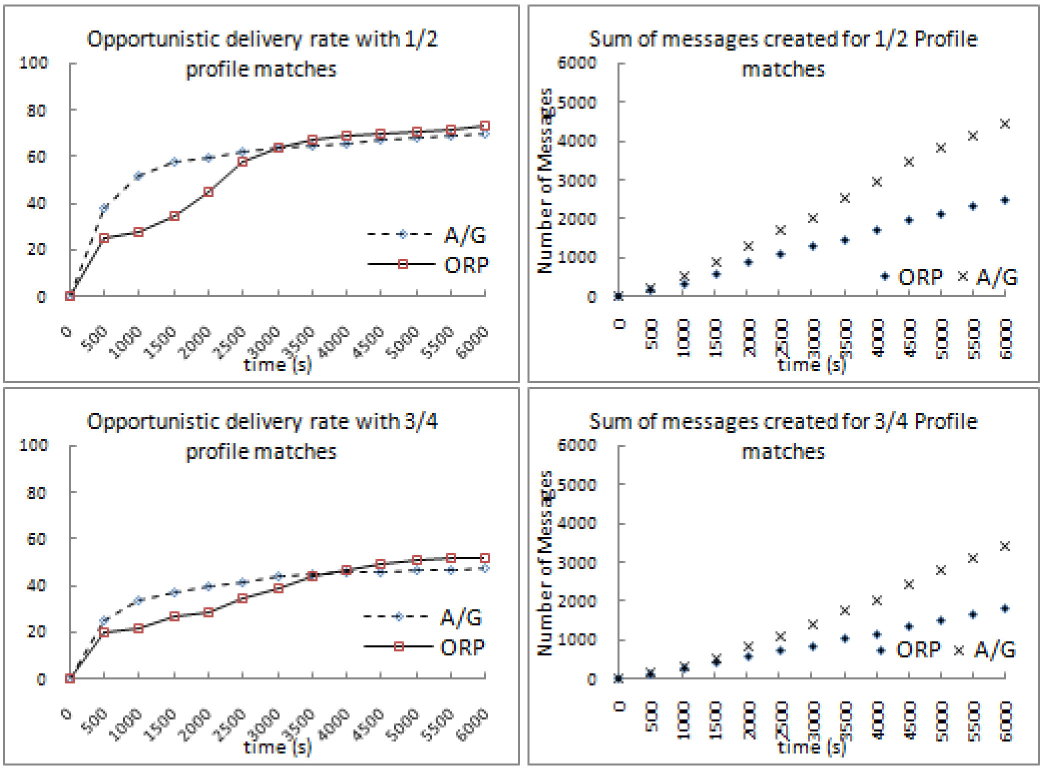
Chapter 5 Results for Opportunistic Routing Protocol

Comparison of ORP protocol and A/G algorithm

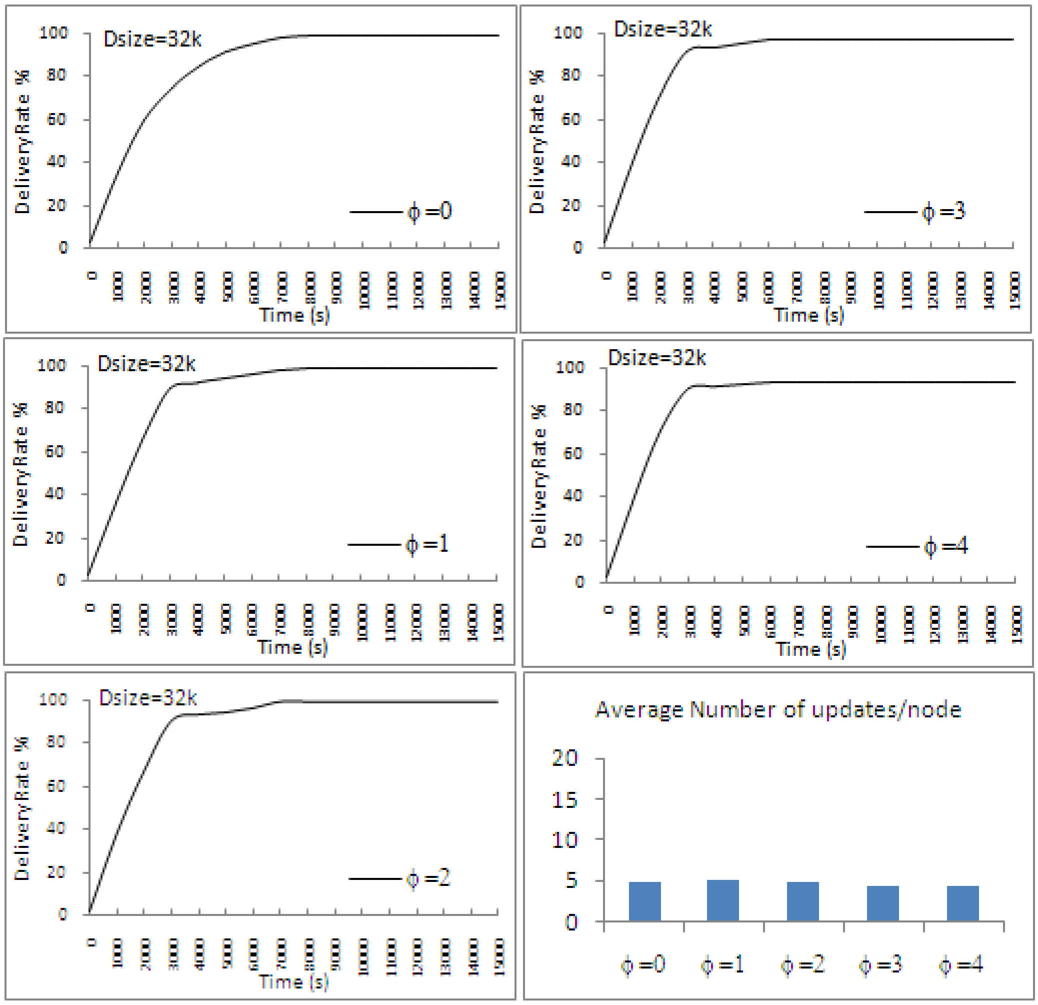
Mobility model	Random Waypoint Mobility model
Number of nodes	100
Number of interest profile (keywords)	32
Repository size	10 MB
Number of hops	$\phi = 0$
Multicasting threshold	50%
Content (document) types used	5
File size used	64KB
Traveler nodes	0
Profiles used for matching	1, 2, 3
Simulation time	6000 s



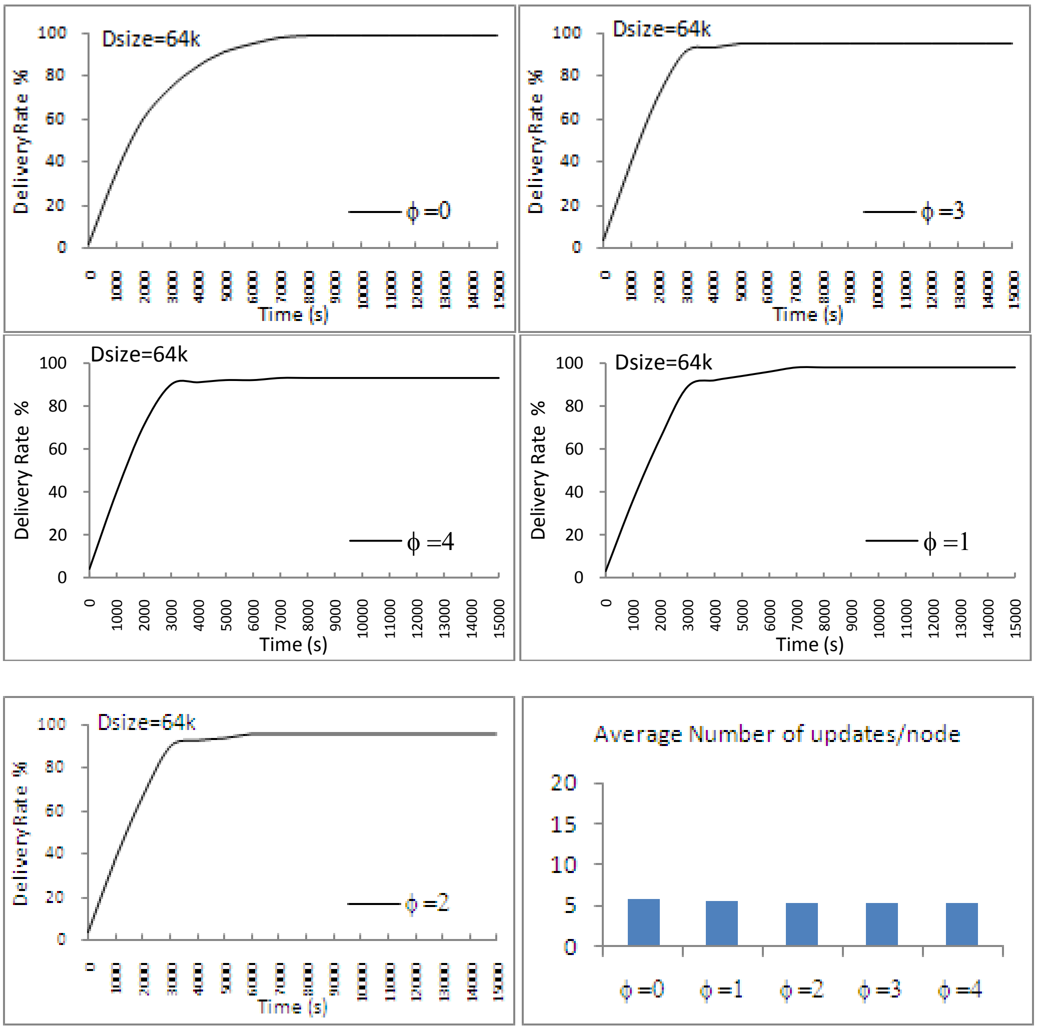




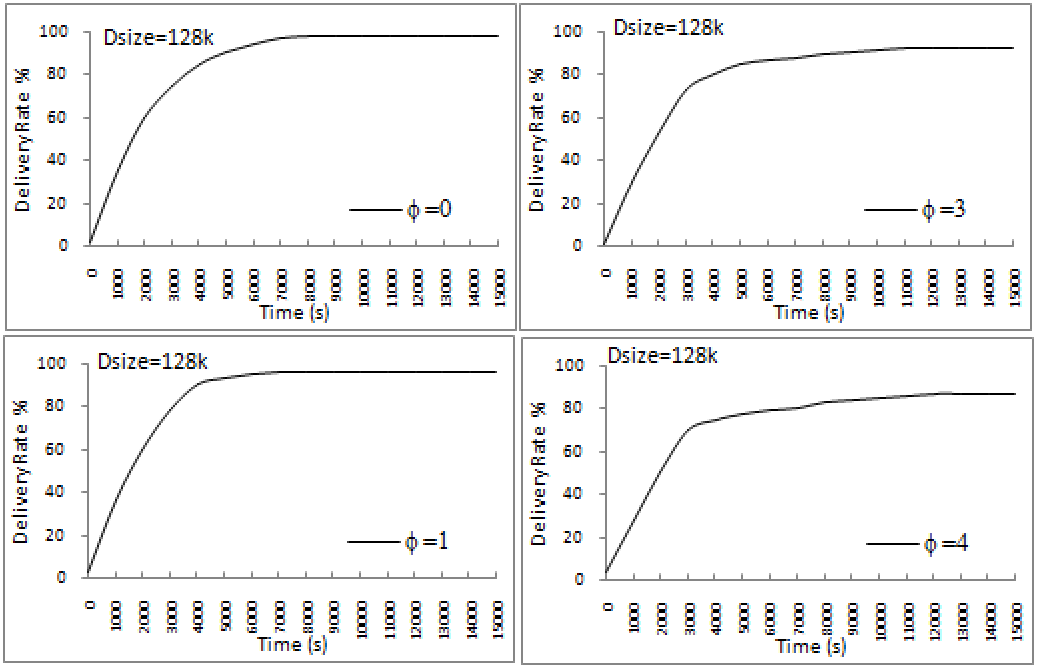
Performance graphs for ORP with document size = 32KB

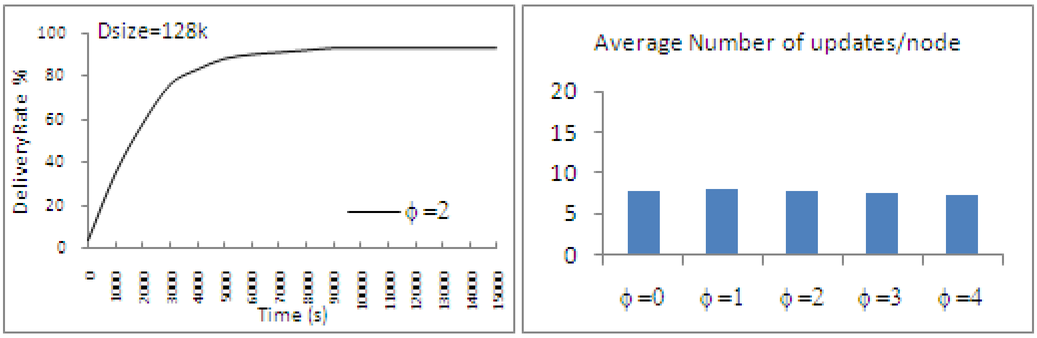


Performance graphs for ORP with document size = 64KB

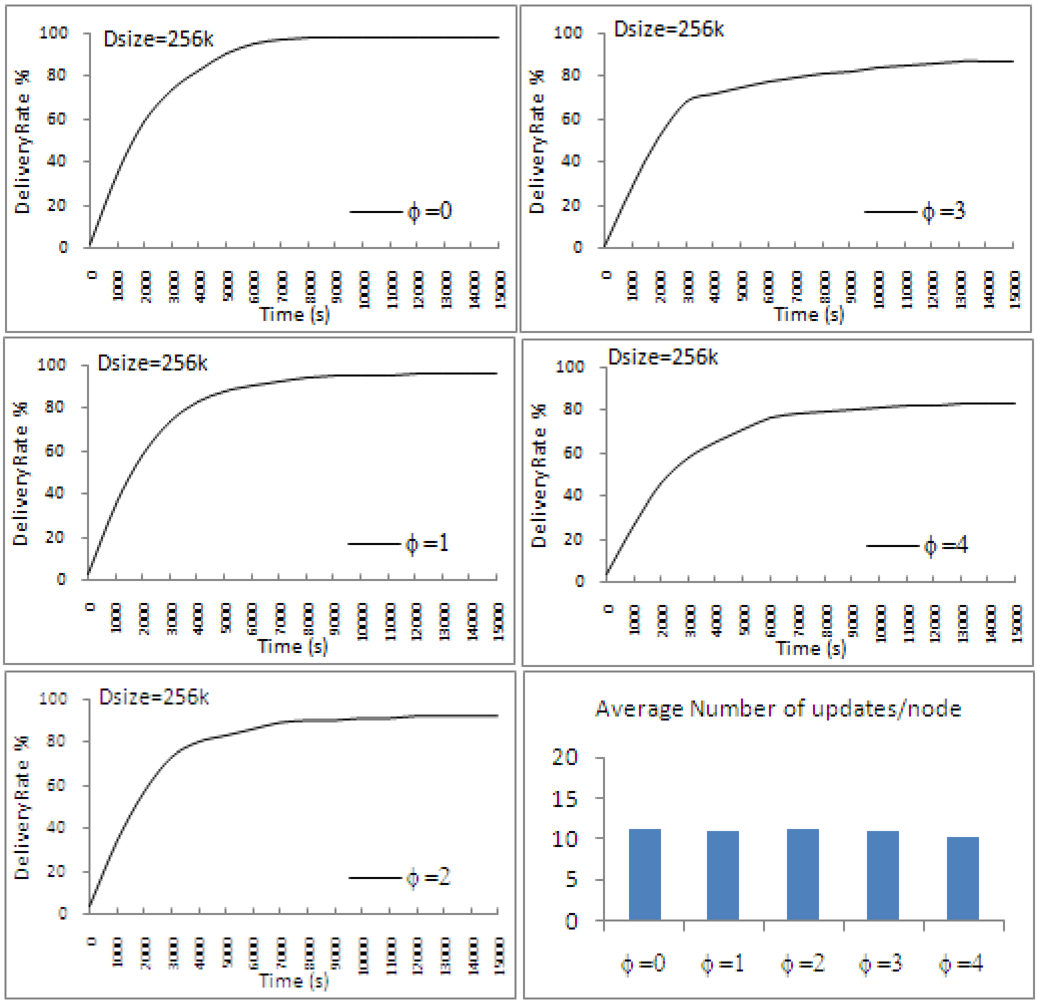


Performance graphs for ORP with document size = 128KB

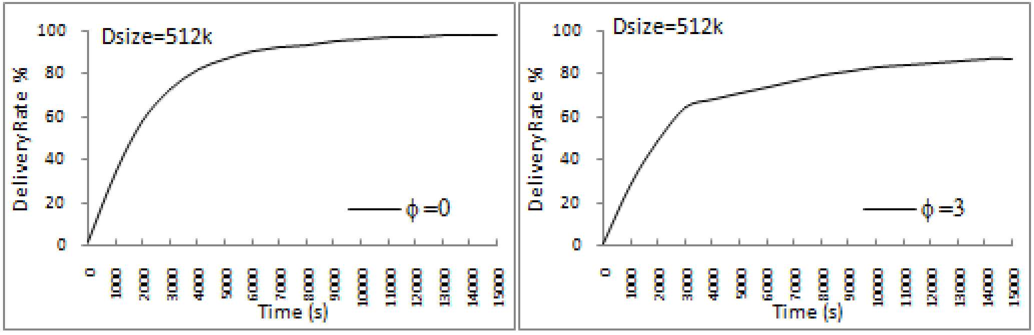


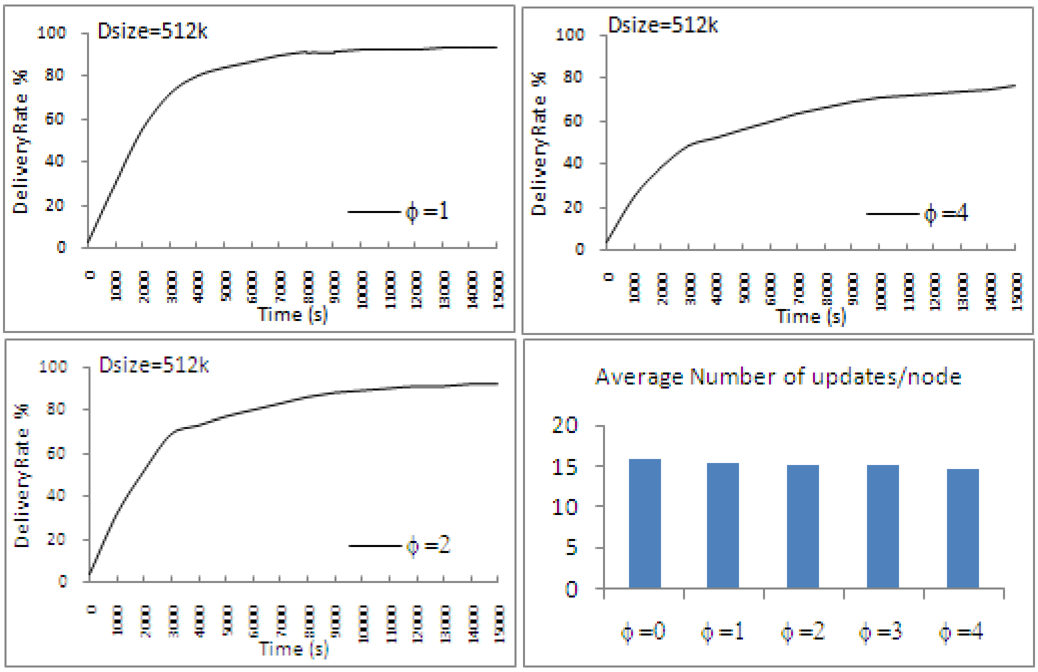


Performance graphs for ORP with document size = 256KB

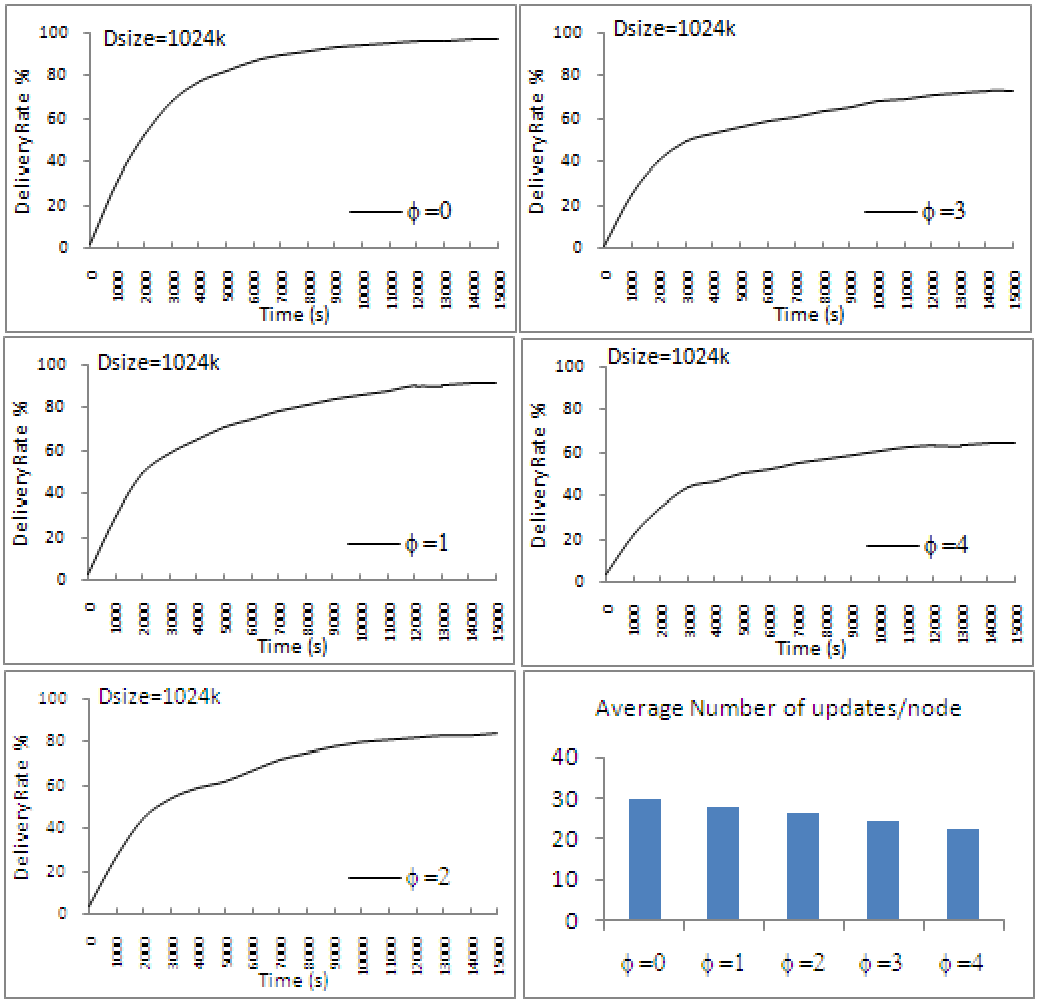


Performance graphs for ORP with document size = 512KB





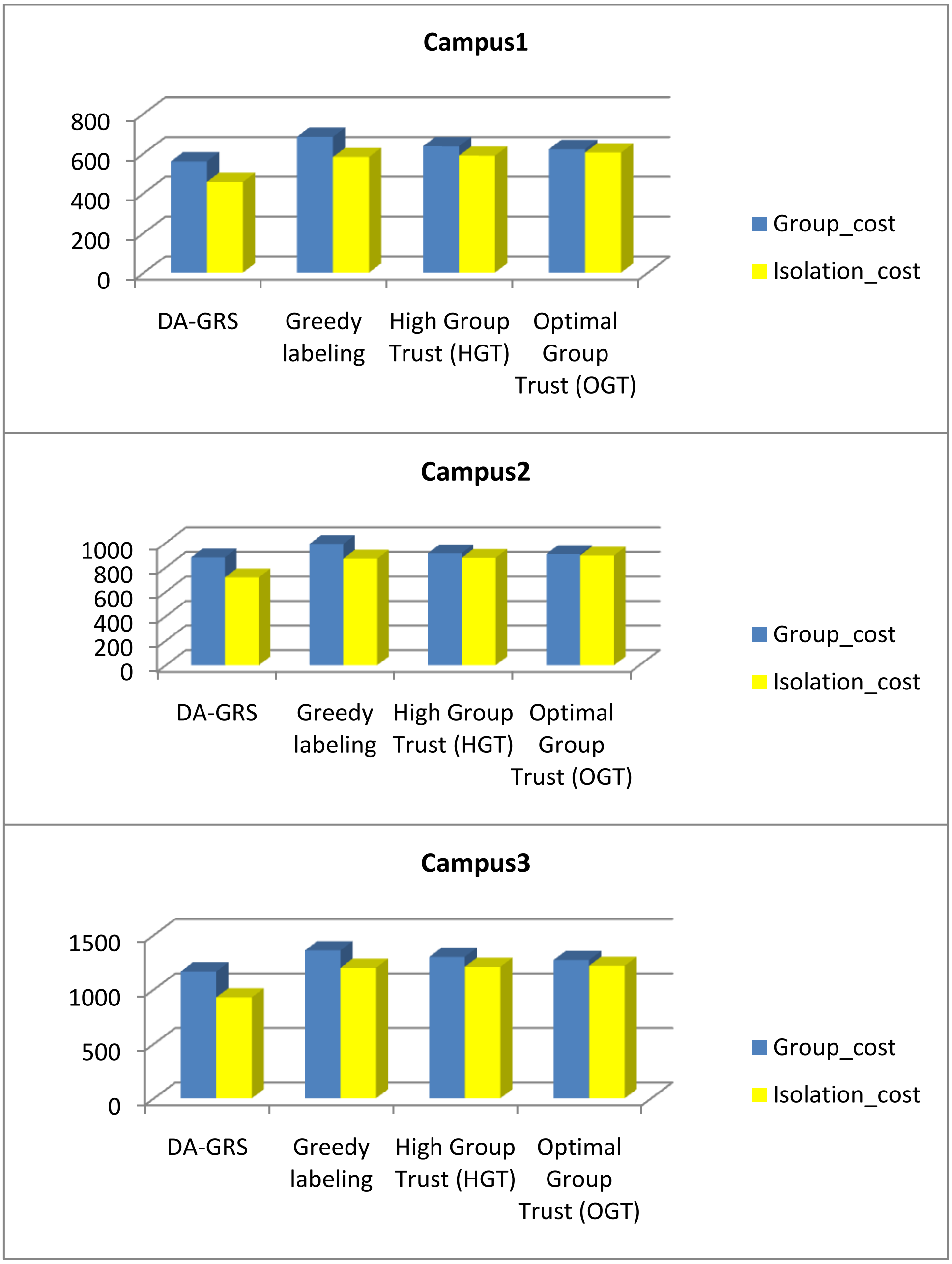
Performance graphs for ORP with document size = 1024KB



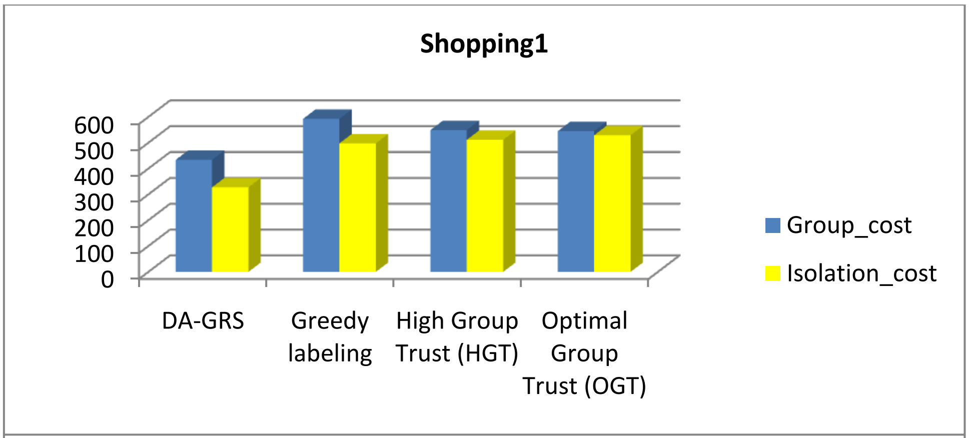
Appendix B

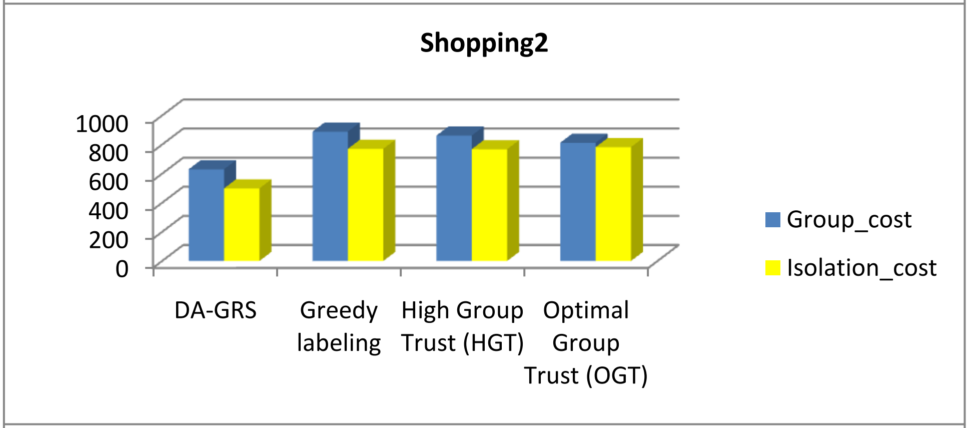
Chapter 6 Results for DAGRS Trust management Technique.

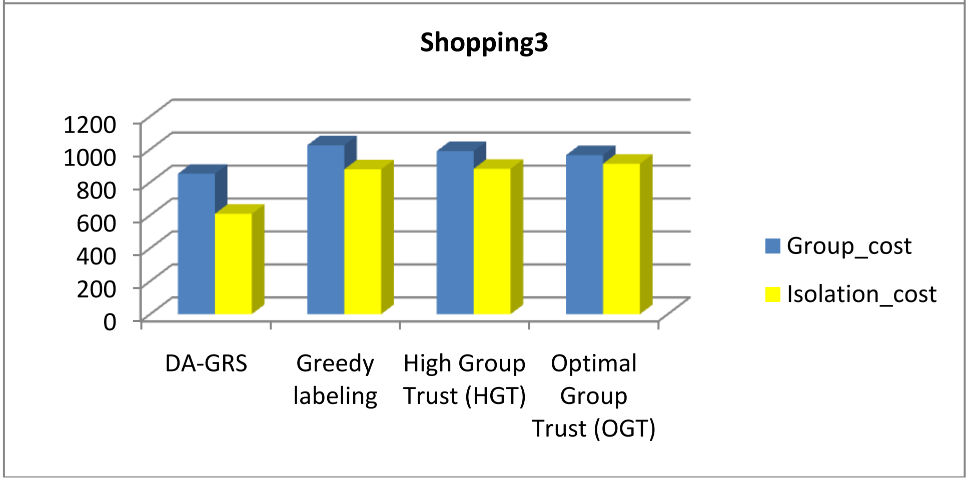
A. DAGRS based Greedy trust management algorithm. A comparison of group_cost and isolation_cost function values for campus networks



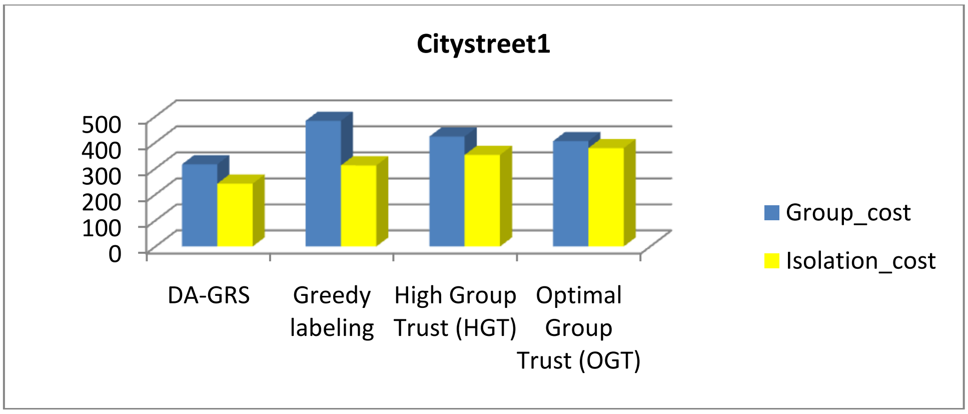
B. Comparison of group_cost and isolation_cost function values for shopping mall networks

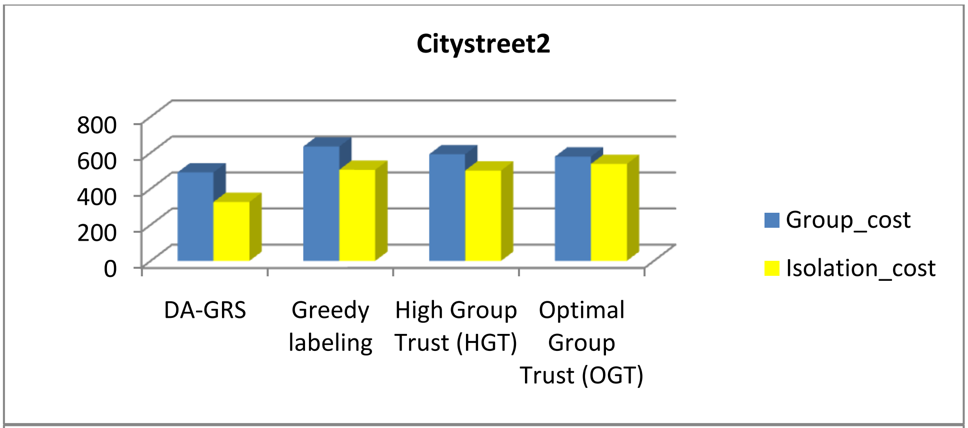


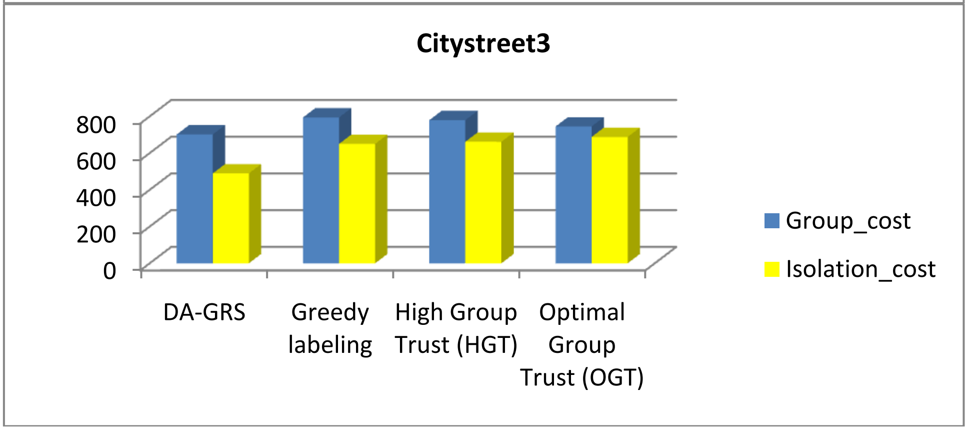




C. Comparison of group_cost and isolation_cost function values for city street networks

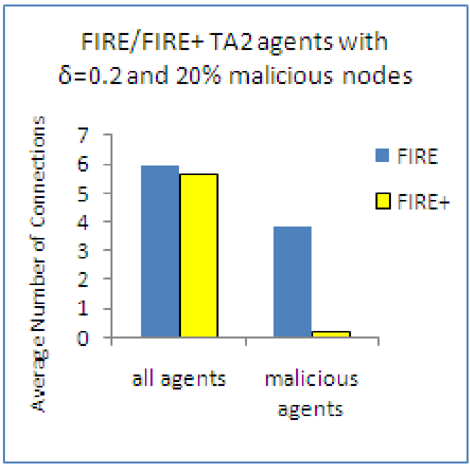


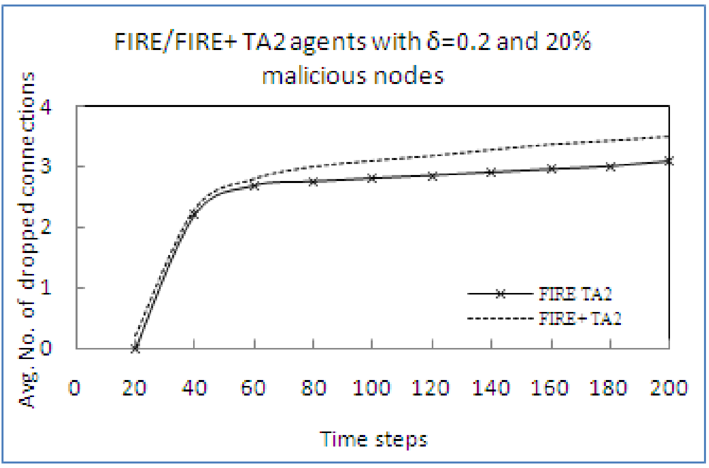


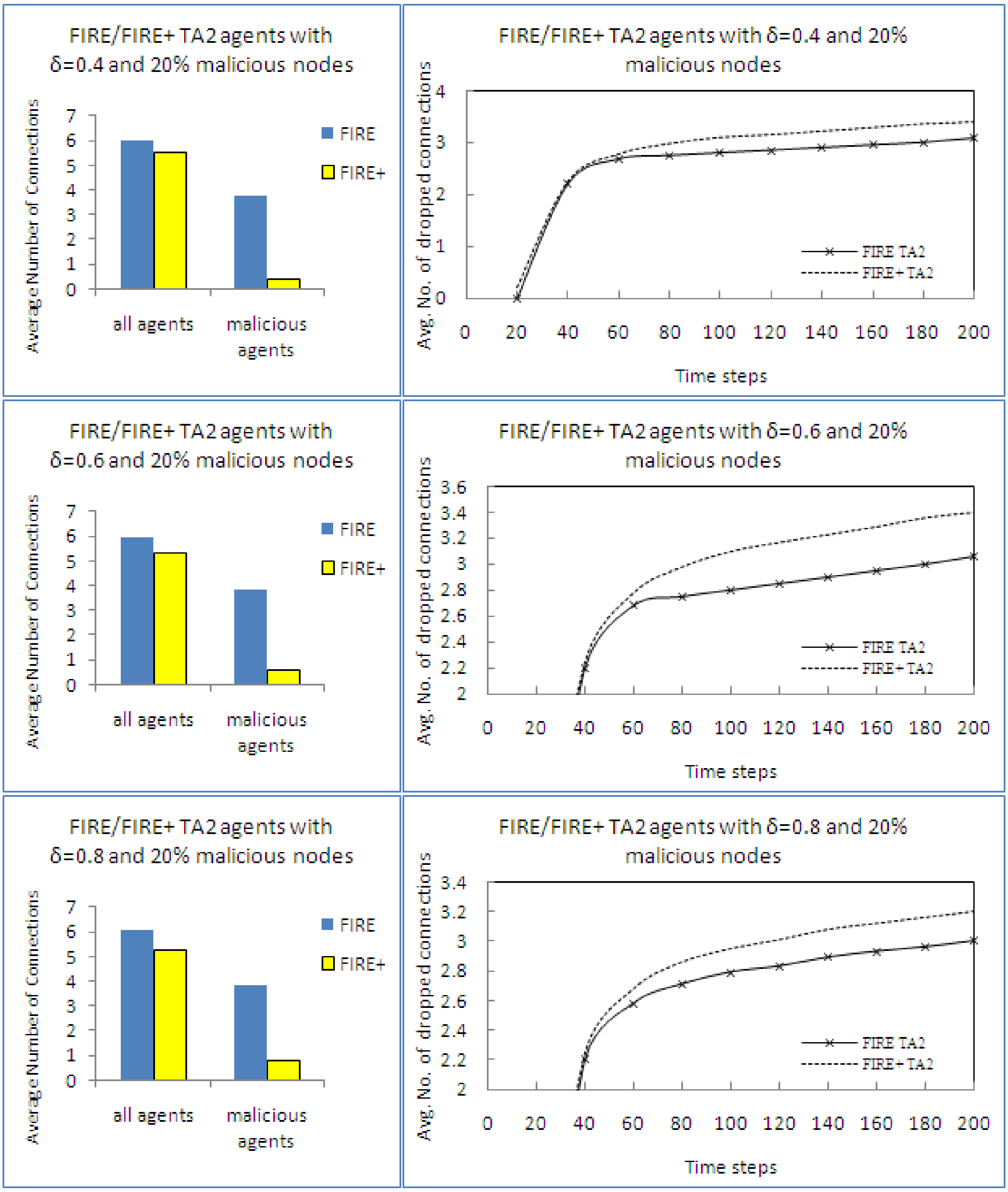


Chapter 7 Results for FIRE+ and FIRE trust models.

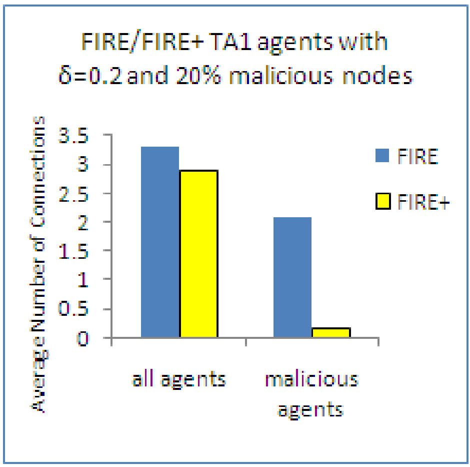
A. Comparison of FIRE and FIRE+ model with TA2 agents and 20% malicious nodes with various values of δ.

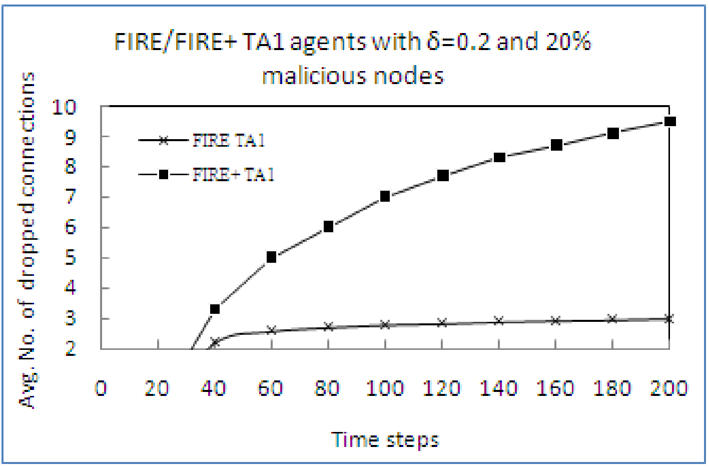


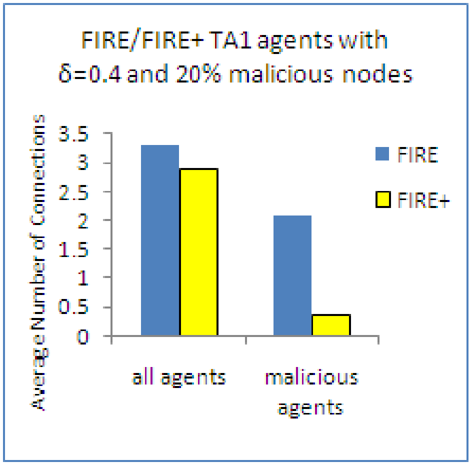


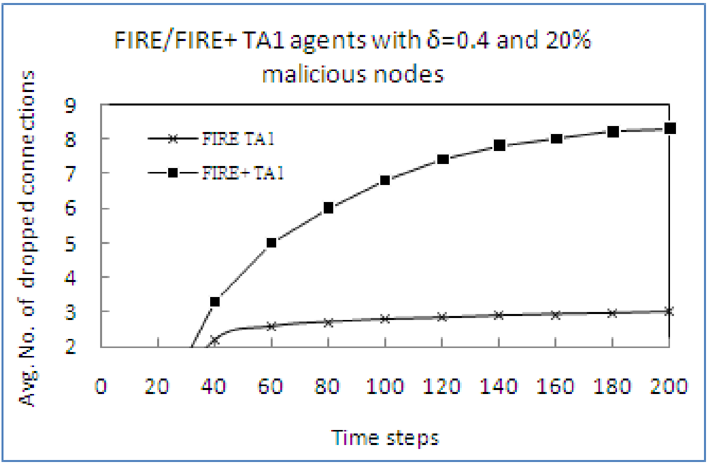


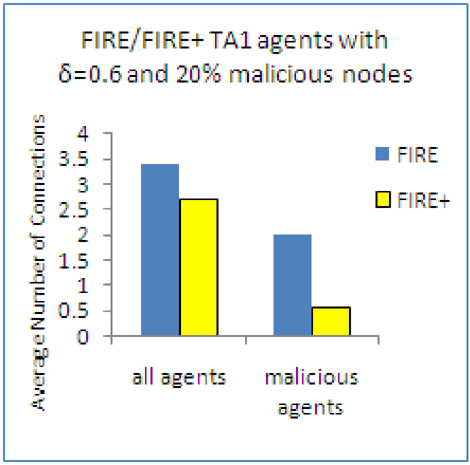
B. Comparison of FIRE and FIRE+ model with TA1 agents and 20% malicious nodes with various values of δ.

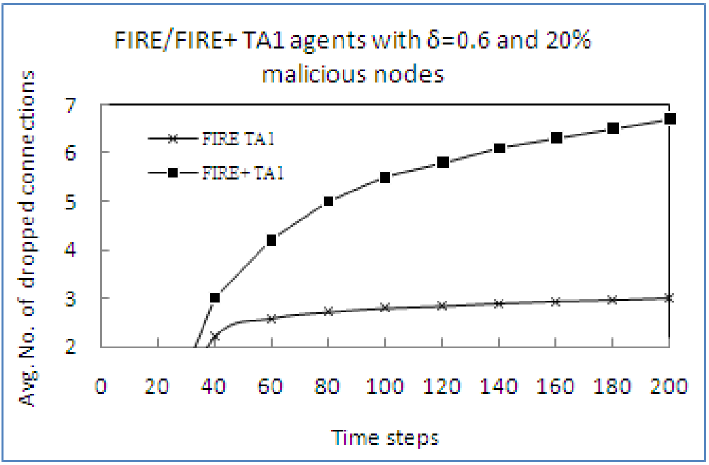

FIRE/FIRE+ TA1 agents with δ=0.2 and 20% malicious nodes

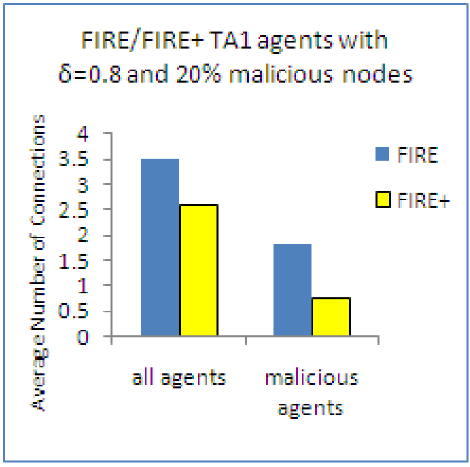

FIRE/FIRE+ TA1 agents with δ=0.2 and 20% malicious nodes

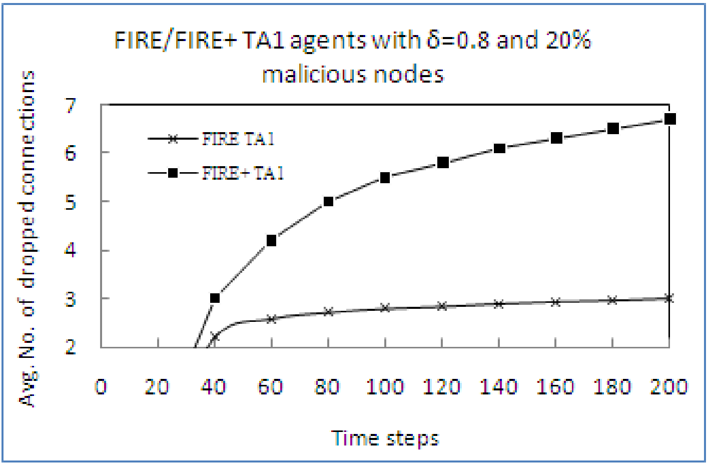

FIRE/FIRE+ TA1 agents with δ=0.4 and 20% malicious nodes


FIRE/FIRE+ TA1 agents with δ=0.4 and 20% malicious nodes

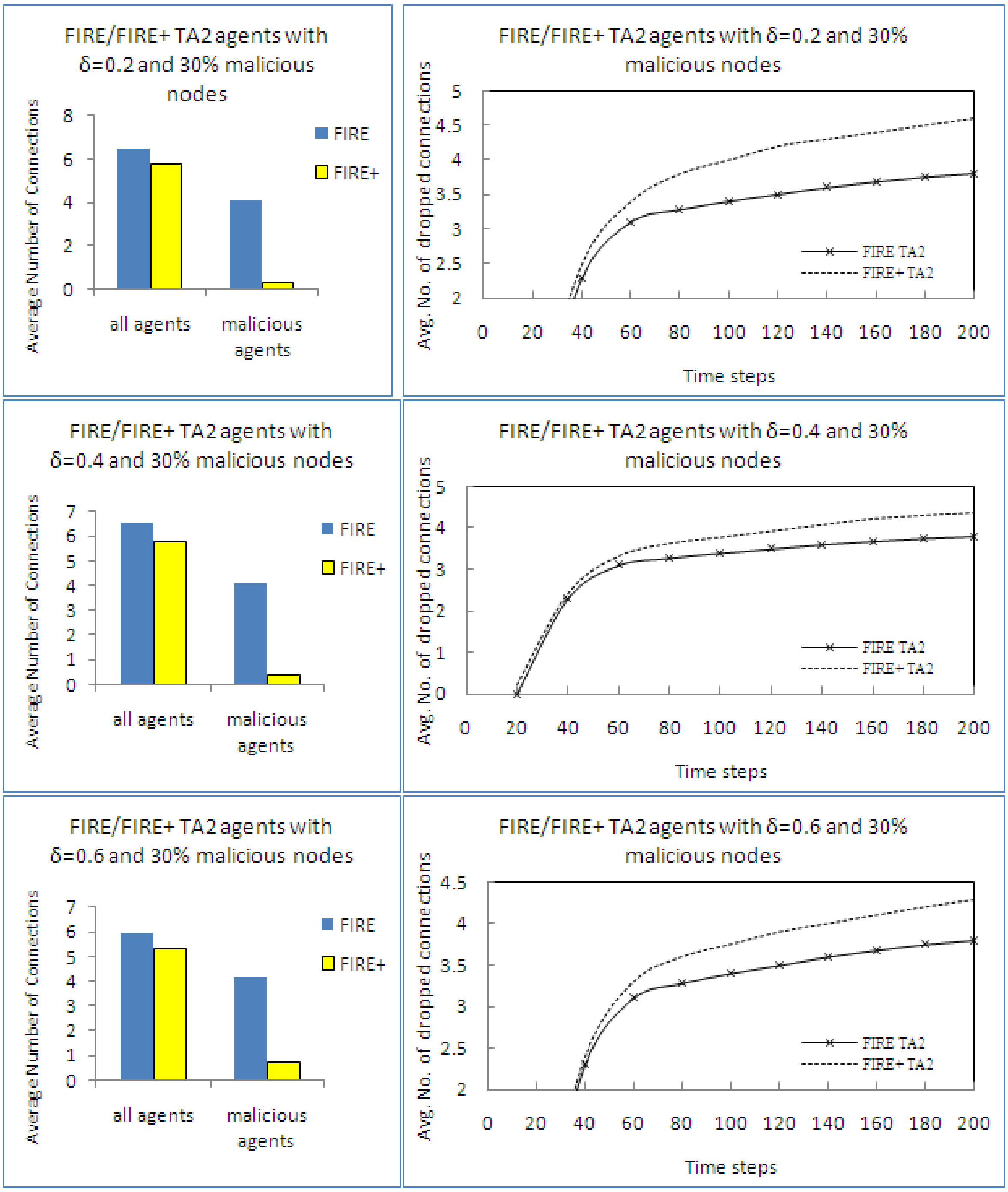

FIRE/FIRE+ TA1 agents with δ=0.6 and 20% malicious nodes

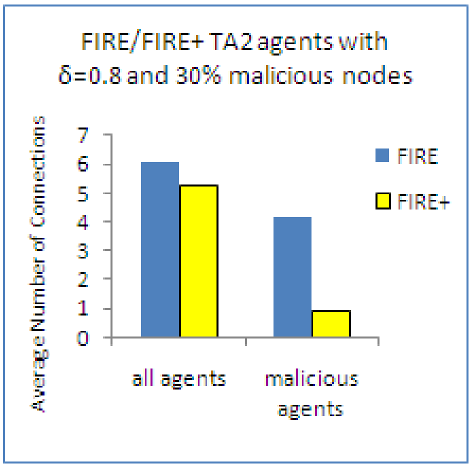

FIRE/FIRE+ TA1 agents with δ=0.6 and 20% malicious nodes

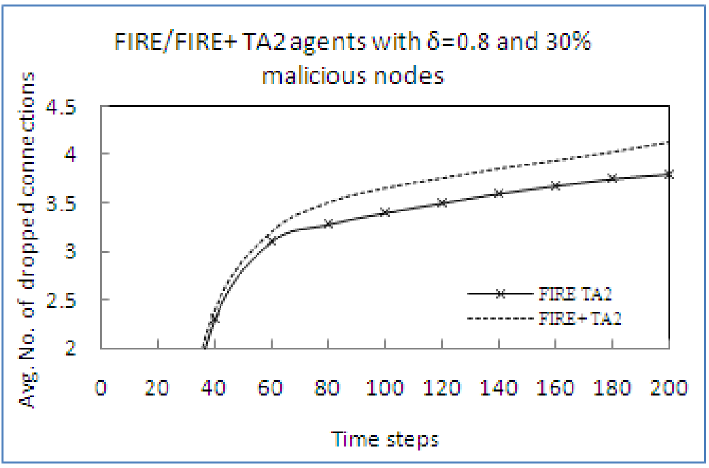

FIRE/FIRE+ TA1 agents with δ=0.8 and 20% malicious nodes


FIRE/FIRE+ TA1 agents with δ=0.8 and 20% malicious nodes

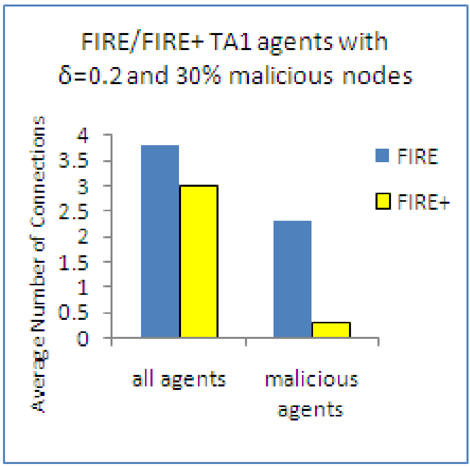
C. Comparison of FIRE and FIRE+ model with TA2 agents and 30% malicious nodes with various values of δ.

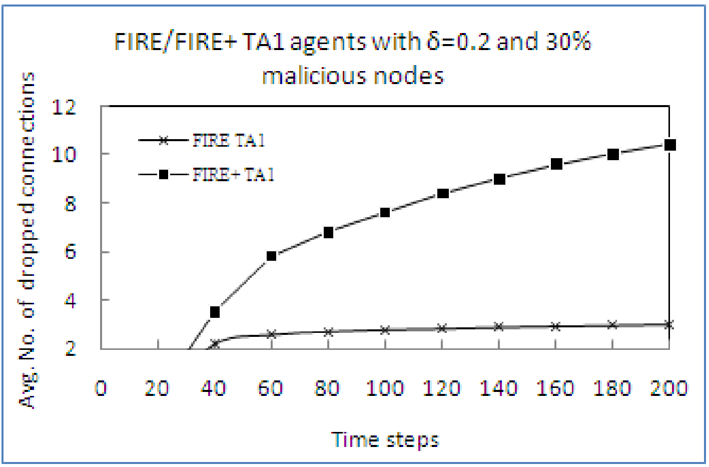


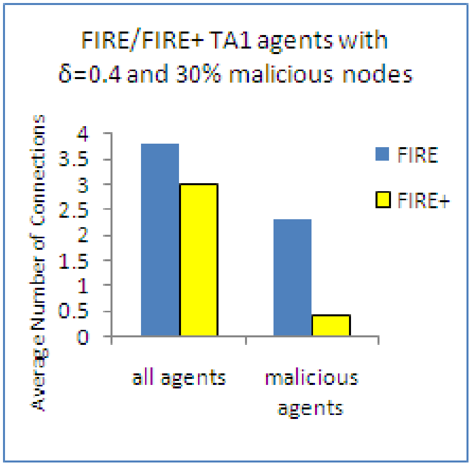


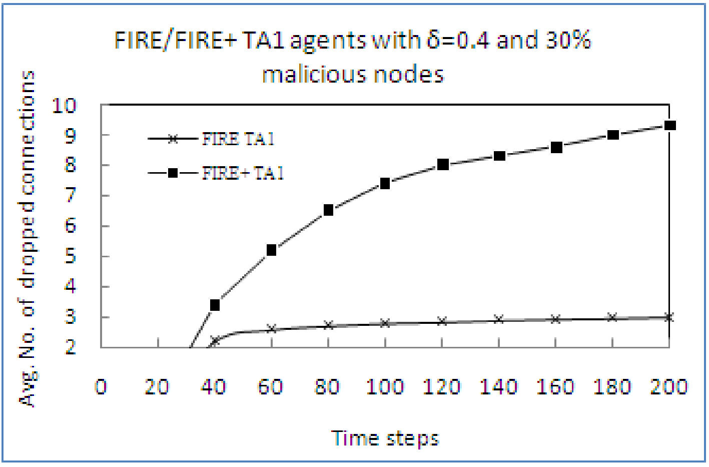


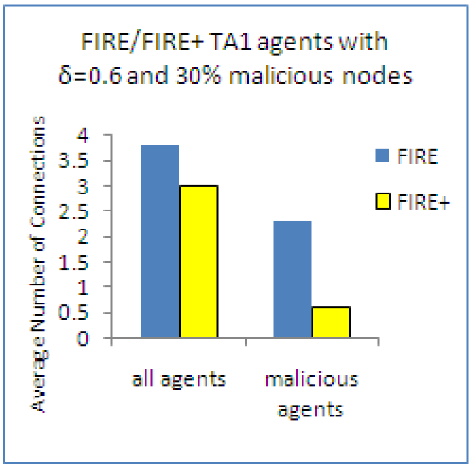
D. Comparison of FIRE and FIRE+ model with TA1 agents and 30% malicious nodes with various values of δ.

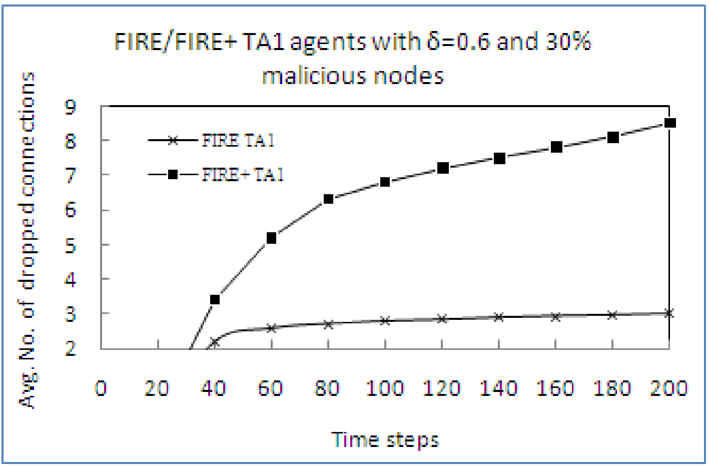


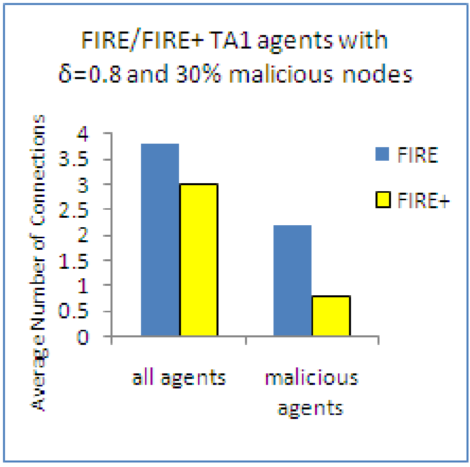


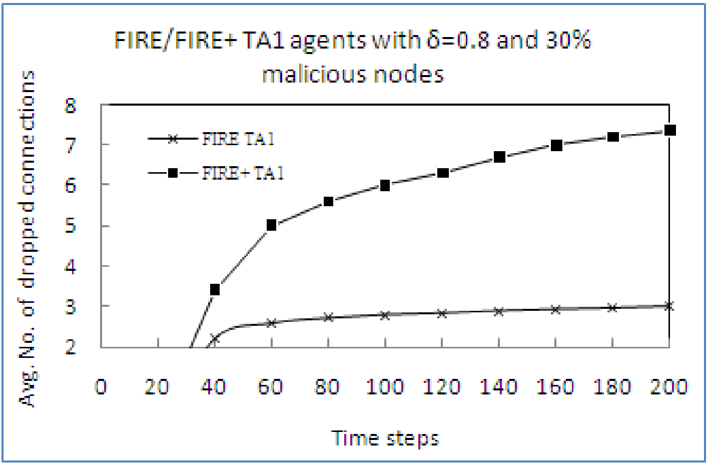





FIRE/FIRE+ TA1 agents with
δ=0.6 and 30% malicious nodes
Average Number of Connections
4
3.5
3
2.5
2
1.5
1
0.5
0
all agents
malicious agents
FIRE
FIRE+


FIRE/FIRE+ TA1 agents with δ=0.6 and 30% malicious nodes
Avg. No. of dropped connections
9
8
7
6
5
4
3
2
0
20
40
60
80
100
120
140
160
180
200
Time steps
FIRE TA1
FIRE+ TA1


FIRE/FIRE+ TA1 agents with
δ=0.8 and 30% malicious nodes
Average Number of Connections
4
3.5
3
2.5
2
1.5
1
0.5
0
all agents
malicious agents
FIRE
FIRE+


FIRE/FIRE+ TA1 agents with δ=0.8 and 30% malicious nodes
Avg. No. of dropped connections
8
7
6
5
4
3
2
0
20
40
60
80
100
120
140
160
180
200
Time steps
FIRE TA1
FIRE+ TA1