This is to certify that the
dissertation entitled

Cooperative Resource Sharing by Integrating
Cellular and Mobile Ad Hoc Networks

presented by

Danyu Zhu

has been accepted towards fulfillment
of the requirements for the

Doctoral degree in Computer Science and Engineering Department

Matt W. Mutka

Major Professor's Signature

14 April 2005

Date

Cooperative Resource Sharing by Integrating Cellular and Mobile Ad Hoc Networks

By

Danyu Zhu

A Dissertation

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Department of Computer Science and Engineering

2005

Abstract

Cooperative Resource Sharing by Integrating Cellular and Mobile Ad Hoc Networks

By

Danyu Zhu

Mobile users are demanding for "anywhere, anytime, always-on and high-speed" wireless services. However, this requirement can not be meet by current wireless technologies. The popular WLANs and WWANs are limited in either coverage range or transmission data rates. On the other hand, the wireless resources are not fully utilized by the mobile users.

In this document, a novel *Cooperative Integrated Wireless Network Architecture (CIWNA)* is proposed to take the advantages of both the high-speed WLANs and the cellular data networks for a better utilization of the costly cellular resource. The mobile devices cooperate and share their idle cellular links to the WWANs via a mobile ad hoc network (MANET) formed from WLAN technologies. Some of the peers may act as servers or proxies, providing different kinds of services for the other members in the same MANET. The WLAN and WWAN technologies are combined to leverage their advantages and share the resources of mobile devices without any changes to

the underlying network infrastructure. The performance improvement in the cellular data networks comes from the efficient peer-to-peer sharing of the idle resources among the mobile users.

Four different applications are proposed aiming to reduce the power consumption of the cellular interfaces, to improve the QoS of the cellular links, and to reduce the download latency of large files from Internet servers. A new set of network protocols are developed for CIWNA to address on the network formation, group management, proxy rotation, message routing, service discovery, failure recovery, and security and privacy. A distributed trust model and a mechanism for using credits are presented to promote the incentive of cooperation among a group of strangers in CIWNA. Peers benefit cooperatively when they follow the scheme. Malicious peers are excluded from the system. Game theoretical analysis is provided to justify the proposed trust model and the credit system.

To *my family*

ACKNOWLEDGMENTS

Writing this dissertation has certainly been hard, but it has not been the debilitating agony as I expected. At the end of this expedition, I am indebted to many people for their inspirations, ideas, encouragement, support, and love.

Foremost, I am deeply grateful to my kind advisor, Professor Matt Mutka for introducing me to this brilliant idea of CHUM. He is grateful acknowledged for his years of encouragement, his scientific contribution, his infinite patience, his insights in our numerous discussions, his financial support and his careful review of this manuscript. Without these generous assistance, this thesis could not have come into light. This opportunity to be one of his students is my great honor. I also owe my thanks to my previous supervisors in China, Professor Shilin Wu and Zhengkang Zhou, for guiding me into this exciting area of computer networking and providing me with valuable chances for working on many projects.

I also wish to express my gratitude to my dissertation committee, Professor Lionel Ni, Professor Abdol-Hossein Esfahanian, Professor Li Xiao and Professor Sarat Dass for sparing their precious time to serve on my committee and giving valuable comments and suggestions.

I am very happy to have worked together with an exciting research group, the ELANS lab at CSE department. I owe my special thanks to Zhiwei Cen for helping me to

finish the idea of QAWBA and parallel downloading, to Hongbo Zhou and Feng Zhu for many interesting discussions.

Heartfelt thanks go out to my parents and my other family members, for supporting me in all possible means during many years. I would like to thank my mother for her years of forever love, inspiration and patience, and also for my father, a professor in Fujian Normal University, for inspiring me to the scientific research. They have been always proud of their son for each progress he has made in study and research.

Finally, I am especially indebted to my wife, Minhua Pan, for her deep love and understanding for my research arcoss two continent. I feel lucky enough to have such a wonderful woman to share the rest of my life. I also would like to thank my lovely daughter, Mindy Zhu, for bring me so much pleasure since she has born.

Table of Contents

LIST OF TABLES . . . . . **xiv**

LIST OF FIGURES . . . . . **xv**

LIST OF ABBREVIATIONS . . . . . **xviii**

1 Introduction and Motivation **1**

1.1 Trend 1

1.2 Motivation 3

1.3 Cooperative Resource Sharing by Integrating Cellular and MANET . . . 5

1.3.1 Cooperative Integrated Wireless Network Architecture (CIWNA) . . . 5

1.3.2 Peer-to-Peer Resource Sharing 7

1.4 Structure of the Content 8

2 Overview of CIWNA Applications **10**

2.1 Assumptions 10

2.2 Applications 12

2.2.1 Sharing Energy Efficient Instant Messaging Channel 12

2.2.2 Sharing General Message/Event Notification Channel 15

2.2.3 QAWBA: QoS Aware Wireless Bandwidth Aggregation 17

2.2.4 Cooperative Multipath Parallel File Downloading 19

3 Literature Review 23

3.1 Integrating Cellular Data Networks and Wireless LANs 24

3.1.1 Overview of Wireless Technologies 24

3.1.2 MANET: Mobile Ad Hoc Network 35

3.1.3 Hybrid Network Architectures 45

3.2 Peer to Peer Computing 50

3.2.1 Peer-to-Peer Applications 51

3.2.2 Manage User Behavior in Peer-to-Peer Systems 54

3.3 Peer-to-Peer Trust Management Systems 57

3.3.1 Policy-based Trust Systems 59

3.3.2 Recommendation-based Trust Systems 60

3.3.3 Social Networks-based Trust Systems 61

4 Sharing Energy Efficient Instant Messaging Channel 64

4.1 Overview 64

4.2 Message Notification Protocol (MNP) 66

4.2.1 Basic Idea 67

4.3 Using Bloom Filters to support Message Notification 71

4.3.1 Bloom filter Background 72

4.3.2 Bloom filters as message notification in CHUM 73

4.4 Implementation 76

4.5 Performance Evaluation 79
4.5.1 Evaluation Settings 80
4.5.2 Energy Analysis Model 81
4.5.3 Summary of parameter settings 84
4.5.4 Evaluation Result 85
4.6 Discussion 89
4.7 Summary 93

5 Power Efficient General Message Notification Service 94
5.1 Introduction 94
5.2 Overview 96
5.2.1 Basic scheme 96
5.2.2 Bloom filter representation of notifications 98
5.3 Group Management 101
5.3.1 Group formation 101
5.3.2 Message/Event notification 103
5.3.3 Proxy scheduling 104
5.3.4 Failure recovery 106
5.4 Discussion 107
5.5 Summary 108

6 QAWBA: QoS Aware Wireless Bandwidth Aggregation 109
6.1 Introduction 109
6.2 QoS Aware On-demand Routing 110

6.2.1 Neighborhood Maintenance 111
6.2.2 Bandwidth Reservation Tables 112
6.2.3 K-path proxy discovery algorithm 114
6.2.4 Computation of Available Bandwidth 119
6.2.5 Failure Recovery and Automatic Resource Release 120
6.3 Performance Evaluation 121
6.3.1 Simulation Model 121
6.3.2 Simulation Results 123
6.4 Summary 126

7 Cooperative Multiple Paths to Reduce File Download Latency . . 129

7.1 Introduction 129
7.2 Proxy Discovery and File Download 130
7.2.1 On-demand Proxy Discovery 131
7.2.2 Pipelining Requests 134
7.2.3 Failure Recovery 136
7.3 Trust Model and Feedback Report 137
7.3.1 Distributed Trust Model 137
7.3.2 Proxy and Forwarder Decision 142
7.4 Performance Evaluation 143
7.4.1 Experiments 143
7.4.2 Simulation Model 145
7.4.3 Single File Downloading Scenario 146

7.4.4 Multiple File Downloading 149

7.5 Summary 151

8 Promoting Fairness Among Strangers in MANET 153

8.1 Introduction 153

8.2 Developing Trust Knowledge 155

8.2.1 Trust Among Peers 156

8.2.2 Trust Evaluation and Update 157

8.2.3 The System of Credits 160

8.3 Promoting Fairness in CHUM 162

8.3.1 Game Theoretic Analysis 163

8.3.2 A Peer's Strategy 166

8.3.3 Discussion 169

8.4 Performance Evaluation 172

8.4.1 Simulation Setup 173

8.4.2 Simulation Results of Honest Peers 175

8.4.3 Simulation Results of Unpredictable Behaviors 180

8.4.4 Cheating and Attacks 181

8.5 Summary 183

9 Conclusion 185

9.1 Summary 185

9.2 Future Work 187

9.2.1 Potential Applications 187

9.2.2 Centralized Infrastructure Support 189

BIBLIOGRAPHY 190

List of Tables

3.1 Key WWAN Standards 27

4.1 Power consumption parameters used in the evaluation 84

4.2 Other parameter settings for the evaluation 85

4.3 Energy consumed in a peer 87

7.1 Download Latency and Throughput Gain w/o Parallel Downloading . . . 145

8.1 Notation of Trust Model 162

8.2 Parameters for Simulation 172

List of Figures

1.1 Example of cooperative integrated wireless network architecture. 7

2.1 Example of CHUM sharing 14

2.2 Example of QAWBA 19

2.3 Cooperative Parallel File Downloading 21

3.1 Cellular Cell Structure 29

3.2 An example of 802.11 ad hoc power saving protocol. 33

4.1 Sharing group creation and message notification service subscription. . . 67

4.2 Message notification. 69

4.3 Proxy migration. 70

4.4 Algorithm for constructing compressed Bloom filter representation . . . 73

4.5 Compressed Bloom filter. 74

4.6 Example of the message notification represented by a Bloom filter. . . . 76

4.7 Snapshots of the CHUM proxy and client implementation 77

4.8 Total energy consumed by the whole group 86

4.9 Total energy saving in the whole group 87

4.10 Energy consumed by a mobile device 88

4.11 Total energy consumed by the whole group 89

4.12 Total energy saving in the whole group 90

4.13 Total energy consumed by a mobile device 91

5.1 The framework of CHUM notification 95

5.2 An example of a subscription list 100

5.3 Group formation 102

5.4 Proxy scheduling 104

5.5 The procedure of message notification 105

6.1 Example of the QoS session 113

6.2 QoS Session Table 114

6.3 Cellular and MANET flow reservation tables 114

6.4 QoS Aware Proxy Discovery Algorithm 116

6.5 Processing of QoS Request Message 117

6.6 Processing of QoS Reply Message 118

6.7 QoS request admission rate with different request bandwidths, $\delta = 40$ . . 123

6.8 QoS request admission rate with different request bandwidths, $\delta = 20$ . . 124

6.9 QoS request admission rate with different request bandwidths, $\delta = 10$ . . 125

6.10 Cellular link utilization with different request bandwidths, $\delta = 40$ 126

6.11 Cellular link utilization with different request bandwidths, $\delta = 20$ 127

6.12 Cellular link utilization with different request bandwidths, $\delta = 10$ 127

6.13 Average number of proxies in QAWBA 128

6.14 Average delay of proxy discovery in QAWBA 128

7.1 On-demand Proxy Discovery Example 133

7.2 Block Request without Pipeline 136

7.3 Feedback Report Protocol Diagram 139

7.4 Performance with different node density 147

7.5 Performance with different cellular link bandwidth 148

7.6 Performance with different file size 149

7.7 Effect of Pipeline Technique 150

7.8 Performance Gain by 5 Downloads 151

7.9 Performance Gain by 10 Downloads 152

8.1 Graph to find equilibrium 165

8.2 State machine for a peer in the group 167

8.3 Proxy rotation in one CHUM group 175

8.4 State transition in one CHUM group 176

8.5 Fairness vs. Peer lifetime 176

8.6 Effect of Credit/Rotate Cycle 177

8.7 Effect of Average Group Size 179

8.8 Effect of Peer/Proxy Leaving Group without Notice 180

8.9 Effect of Free Riders in the System 182

8.10 Trust about Peer 2 in Peer 1 over Time 183

List of Abbreviations

- QoS: Quality of Service
- IM: Instant Messaging
- WWAN: Wireless Wide Area Network
- WPAN: Wireless Personal Area Network
- WLAN: Wireless Local Area Network
- QAWBA: QoS Aware Wireless Bandwidth Aggregation
- CIWNA: Cooperative Intergrated Wireless Network Architecture
- MANET: Mobile Ad Hoc Network
- CHUM: Cooperating ad Hoc network to sUpport Messaging
- MNP: Message Notification Protocol

Chapter 1

Introduction and Motivation

1.1 Trend

Over the past decade, the world has become increasingly mobile. The demand for wireless services has experienced major growth with both the worldwide upgrade of cellular networks and widespread deployment of wireless local area networks and hotspot services. Many wireless access technologies have been developed and standardized by industry companies and academic researchers. These technologies include the wide area cellular data networks to support wireless Internet data access, IEEE 802.11 based wireless local area networks for high speed wireless network connections, and Bluetooth/IEEE 802.15 wireless personal area network technologies for pervasive computing environment. An important reason for the existing of heterogeneous wireless access technologies is behind the performance tradeoffs they exhibit in terms of mobility support, network capacity, coverage area and transmission power. One technology that is most suitable for a scenario may become useless in the changing

environment. It is widely perceived that the future wireless networks will exhibit a similar trend with coexisting of different wireless access technologies.

For example, the wireless local area networks (WLANs) are able to provide nearly Ethernet-equivalent data rates within limited transmission range. They can be used for wireless extension of the traditional wired networks or hotspot services in defined environments, such as libraries, hotels and airports. The goal of wireless wide area networks (WWANs), on the other hand, is to provide "anywhere, anytime, always on" services since they have a wide transmission range in dozens or hundreds of miles. To cope with the heterogeneous wireless technologies, a mobile user today can potentially be equipped with multiple wireless interfaces that have access to different wireless networks. The convergence of mobile phones and computers also foresees the popularity of such devices. New smartphones and pocket PC phones that support both cellular data networks and IEEE 802.11-based WLANs are available. Some vendors are beginning to consider combing 2.5/3G and Wi-Fi silicon into one device. For example, Avaya, Motorola and Proxim are co-developing a dual-mode smartphone and infrastructure to enable mobile phone users roam between cellular networks and WLANs. In the near future, it will not be unusual that a mobile device may have an integrated IEEE 802.11-based wireless interface to access the local access point, a Bluetooth adapter to connect peripherals, and a 3G interface card for Internet connection. The new GSM/GPRS-enabled iPaq H6315 developed by HP and T-Mobile is an example that combines mobile phone technology with Wi-Fi and Bluetooth wireless networking.

With all the wireless technologies available, the wireless networks are beginning to become an integral part of the global communication architecture. Eventually wireless users may demand for the same Quality of Service (QoS) and features that are currently available on today's wired networks. "Anywhere, anytime, always on and high-speed" connections are preferred by the mobile users. However, none of the existing wireless networks can meet this requirement. The 3G cellular data networks are aimed to provide "anywhere, anytime" services. However, the low data rates that they can support greatly limit the potential applications. Existing 2.5G networks support bandwidth to the tune of a few tens of Kbps per user, while the next generation 3G wireless networks support bandwidth of a mere 384Kbps per channel outdoors and 2Mbps per channel indoors. The WLANs (such as IEEE 802.11a/b/g) can up to 54Mbps bandwidth, however, with very limited transmission range. The future generation of wireless networks (4G and beyond) are addressed for an open architecture across different wireless network technologies and providing high data rates. The research of 4G is in the beginning stage. The evolution may take tens of years and cost millions of dollars to deploy. New techniques and architectures are still needed to overcome the limitations of current wireless technologies.

1.2 Motivation

A considerable body of research has been done to improve performance of each existing wireless technology in isolation. Achievements of such research include smarter radio transmission [1], better channel access schemes [2], more efficient scheduling

schemes [3], fast and intelligent hand-offs [4], and transport protocols that are wireless-aware [5]. Most of them involve a significant modification in the existing wireless network infrastructure and the wireless interface design. They are not practical for immediate implementation.

A recent class of approaches for enhancing the performance of cellular wireless data networks has focused on improving the underlying network model. It has been shown that using the peer-to-peer network model, a mode of communication typically seen in mobile ad hoc networks, can result in performance improvements such as increased data rate, reduced transmission power, better load balancing, and enhanced network coverage [6]. However, all of them still demand some modifications in the existing network infrastructure, deployment of new hardware, or the involvement of cellular network operators [7, 8, 9].

On the other hand, the available mobile resources are not fully utilized by the mobile users. When connecting to a cellular data network with a 3G wireless interface, the high-speed WLAN interface is generally idle. The requirement of always-on Internet connections results as idle cellular links most of the time due to the bursty nature of Internet usage pattern. A good example is the instant messaging (IM) service. The near real-time interactive functions of IM require a continuous connection to the Internet IM server. However, the real traffic for data transmission is very small. The costly cellular link is idle most of the time, consuming valuable battery power of the mobile device.

One open question that remains is whether we can utilize and combine the current wireless technologies to leverage their advantages and share the resources of mobile devices without any changes to the underlying network infrastructure.

1.3 Cooperative Resource Sharing by Integrating Cellular and MANET

1.3.1 Cooperative Integrated Wireless Network Architecture (CIWNA)

In this document, we develop a novel *Cooperative Integrated Wireless Network Architecture (CIWNA)* that could take advantage of both the high-speed WLANs and the cellular data networks by utilizing two wireless network interfaces simultaneously to provide a better utilization of the costly cellular resource. CIWNA will significantly improve the performance of the low-throughput, high-power-consumption cellular data links. No modifications in the underlying wireless infrastructure and special hardware in the mobile devices are needed. The improvement in the cellular data networks comes from the efficient peer-to-peer sharing of the idle resources among the mobile users. Four different applications are proposed aiming to reduce the power consumption of the cellular interfaces, to improve the QoS of the cellular links, and to reduce the download latency of large files from Internet servers.

The basic idea is that the mobile devices cooperate and share their idle cellular links to the WWAN via a mobile ad hoc network (MANET) formed from WLAN technologies [1]. Some of the peers may act as servers or proxies, providing different kinds of services for the other members in the same MANET by contributing their idle cellular connections to the WWANs. For example, a mobile device may act as a proxy to provide message notification service for other nearby peers in order to relief their burden of the power costly continuous cellular connections to the WWANs. In another example, a mobile device may contribute its idle cellular bandwidth for another mobile device in the same MANET to improve its throughput in file downloading. The mobile devices take turns to be proxies or clients in order to achieve fairness. Therefore all the mobile users may benefit from the new integrated wireless network architecture. Figure 1.1 shows an example of the cooperative integrated wireless network architecture. Five mobile devices (A, B, C, D and E) form a MANET via their IEEE 802.11 network interfaces. Node A and node D are acting as servers or proxies by contributing their cellular data links to node B, C and E.

In developing such an integrated wireless network architecture, we need to develop a new set of network protocols to address on the network formation, group management, proxy rotation, message routing, service discovery, failure recovery, and security and privacy. The details of the protocols may depend on the specific applications which we will explore further in the following chapters.

[1] IEEE 802.11 based wireless interfaces can be used to form the MANET when operating under an ad hoc mode. Alternatives are wireless personal area network (WPAN) technologies, such as Bluetooth and IEEE 802.15

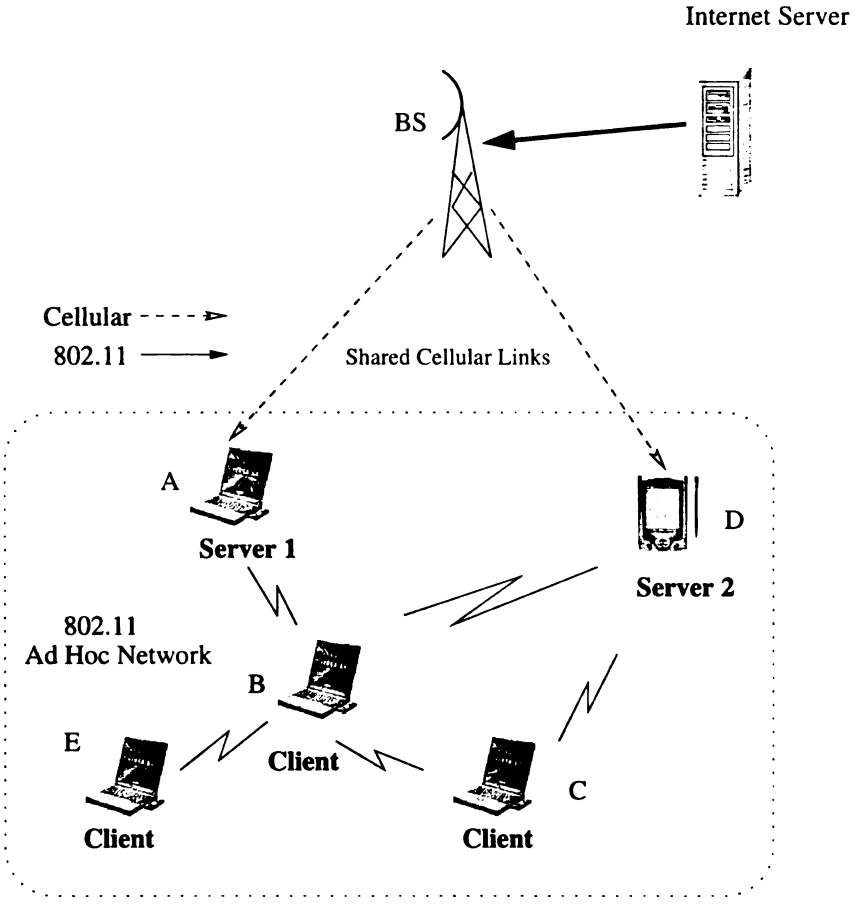


Figure 1.1: Example of cooperative integrated wireless network architecture.

1.3.2 Peer-to-Peer Resource Sharing

The idea of CIWNA is inspired from the popular concept of peer-to-peer computing, which has been widely used by many peer-to-peer applications [10]. In these applications, the free resources in the edge of the network, such as CPU computation power and disk storage, are shared among peers in order to bring benefit for the whole community. The peers are both acting as servers and clients by providing services to others while receiving services from others. The community is formed without any centralized authority.

The CIWNA tries to utilize the idle cellular data links in the mobile devices. The costly cellular connections are shared among a group of mobile users to improve the performance of all members in the same cooperative network. By joining the cooperative network, the mobile devices may reduce their power consumption, improve the QoS, or reduce the latency of large file downloading. The mobile device is acting as a server while contributing its idle cellular connection to WWAN and acting as a client while using the cellular connection in another mobile node. The whole network is formed without support from centralized servers or network operators. No modification is needed in the existing network infrastructure.

Such network architecture requires cooperation among a group of users to gain benefit for the whole community. However, when the users do not belong to the same authority, they lack the motivation to cooperate when selfish behavior may provide more benefit. In order to promote the incentive of cooperation among a group of strangers in the highly mobile environment, such as CIWNA, we develop a distributed trust model and a credit system in chapter 8. In this model, peers evaluate the trust they have for other peers. A peer that obeys the established protocols trusts other peers who are perceived to follow the protocols. Trustworthy peers reap the benefit of service provided by others.

1.4 Structure of the Content

The remainder of this document is structured as follows. In chapter 2, an overview of the applications are presented. Background information and a literature review is

given in chapter 3. In chapter 4, the application of sharing instant messaging channel is explored in order to reduce the power consumption and telecommunication cost. Chapter 5 extends this idea into a more general event notification channel sharing. In chapter 6, QoS aware bandwidth aggregation is presented. The framework is further applied in chapter 7 for a faster file downloading. The solution to promote fairness among mobile devices is presented in chapter 8. Finally, a summary and possible future work are outlined in chapter 9.

Chapter 2

Overview of CIWNA Applications

2.1 Assumptions

Before discussing several applications of the cooperative integrated wireless network architecture, we make several assumptions throughout the whole document.

- **Two wireless interfaces:** We suppose that the participating mobile devices are equipped with both cellular-based and IEEE 802.11-based wireless network interfaces, and have the ability to access cellular data networks and IEEE 802.11 based WLANs simultaneously.

- **Radio interference:** There is no radio interference between the IEEE 802.11-based WLANs and the cellular data networks since they operate under different frequency band ranges. The IEEE 802.11b and 802.11g standards operate on the 2.4-2.483GHz band range and the 802.11a standard operates on 5.15-5.25GHz, 5.25-5.35GHz and 5.725-5.825GHz band ranges. The 2.5G of cellular networks

(e.g., GPRS, EDGE) operate on 800-900MHz (cellular band) and 1.5-1.8GHz (PCS band). Future 3G cellular networks will operate on 1.885-2.023GHz and 2.210-2.200GHz band ranges.

- **Communication cost:** Mobile users may be charged by the number of bytes transmitted, by the length of time connected, or have unlimited access within the cellular data network. They have more incentive to cooperate if there is unlimited access. However, they are more sensitive to fairness when there is a per-byte service charge or per-minute service charge in some applications. A carefully designed trust model and credit system is proposed to promote the incentive of cooperation. On the other hand, the access to the IEEE 802.11 based MANET is free of charge to all the mobile nodes.

- **Routing Protocol:** Cellular networks operate as a single hop (mobile nodes to the base station) topology. The IEEE 802.11 based wireless interfaces can operate both under infrastructure mode with an access point and ad hoc mode without an access point. In this document, we suppose all the IEEE 802.11 based wireless network interfaces are working under an ad hoc mode and forming a MANET.

2.2 Applications

With the basic network architecture and the concept of peer-to-peer resource sharing, there are many possible applications. In this document, we have explored four those applications.

1. **Sharing Energy Efficient Instant Messaging Channel**
2. **Sharing General Message/Event Notification Channel**
3. **QAWBA: QoS Aware Wireless Bandwidth Aggregation**
4. **Cooperative Multipath Parallel File Downloading**

In the rest of this section, we will introduce some background information and the basic ideas of these applications. Although all of the four applications share the same network architecture and peer-to-peer sharing concept, the different features of the applications require different network protocols and algorithms. The details of application specific protocols, algorithms and performance evaluations will be shown in the following chapters.

2.2.1 Sharing Energy Efficient Instant Messaging Channel

An increasingly important application that may thrive in an even greater manner when powerful mobile devices and WWAN networks merge is *instant messaging (IM).* IM services have grown enormously in the past few years. Their popularity are well known among youth to contact and *chat* with *on-line friends.* Its use among corporate

members is also expected to increase significantly. Many commercial instant messaging services are available and gaining soaring popularity. These service providers include Microsoft [11], AOL [12], ICQ [13] or Yahoo! [14]. IM services require knowledge of *presence*, or availability, of a user to respond to requests to conduct IM message exchanges. Short Message Service (SMS), which is related to IM, is available on mobile telephones and is very popular in many parts of Europe and Asia. However, SMS does not support presence and the "near" real-time interactive functions of IM. In spite of the improved technologies to enable mobile devices to have continual access to Internet services, battery power consumption and network access costs play a role in inhibiting their widespread deployment for use in new and existing services such as mobile IM[1]. In the IM service usage, users need to maintain their presence with the IM service. Continuous presence means continuous energy consumption by the devices. Although some power saving features may be deployed, the devices need to be ready continually to receive IM requests. If a device is repeatedly powered off, and on, to conserve energy, then presence information of the user is not continuous, which means the user may miss attempts to exchange IM interactions with others. For wide deployment of pervasive computing systems that required continual Internet access, such as IM, new methods may be needed for reducing the power consumption.

Users of IM services may be in the proximity of each other when they visit shopping malls, attend sporting events, wait within airports, or congregate in other places in large numbers. Mobile devices may connect to the Internet via their ISP via a 3G connection, and then connect to their favorite IM service provider, such as Microsoft

[1]The Compaq iPAQ 3670 can sustain no more than 3 hours of continuous usage.

MSN Messenger, AOL, ICQ, or Yahoo!. Some may even connect to less well-known services such as Jabber [15], which provides plug-in compatibility with most of the well known IM service providers.

To save battery power, each mobile device may periodically connect and disconnect to the ISP. While disconnected, the IM server may cache messages that would be missed. If disconnected for an extended period, then the IM server provides presence information that indicates that the PDA user is not available. Reconnection incurs network costs. Frequent reconnections increase the network costs with the hope of reducing missed IM messages.

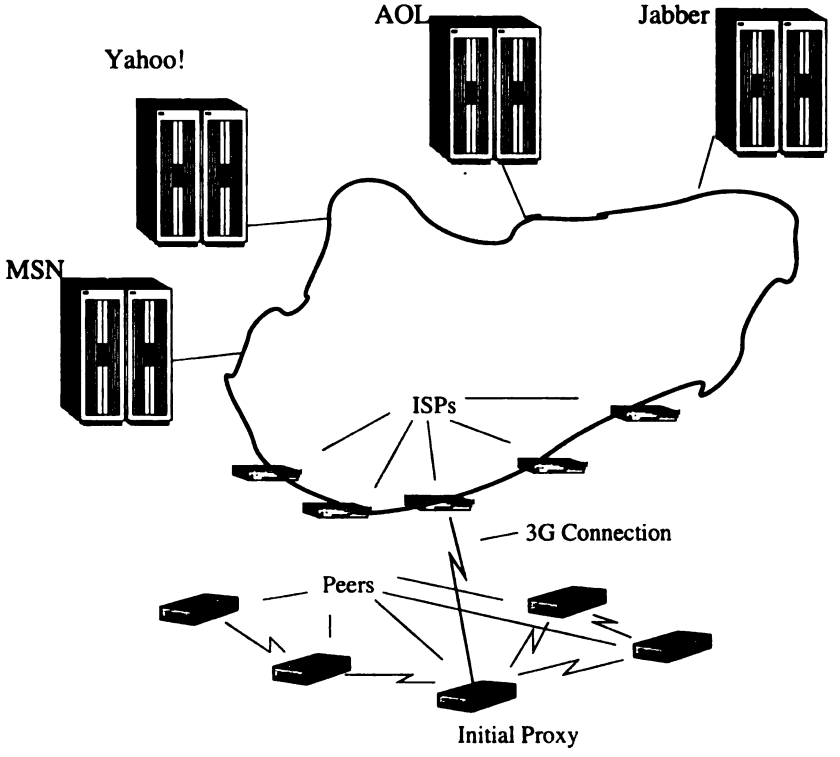


Figure 2.1: Example of CHUM sharing

Our effort is to reduce the network and power consumption costs by forming a Cooperating ad Hoc network to sUpport Messaging (CHUM) by means of WLAN technology

with a set of nearby peers, only one peer at a time is required to serve as a proxy to be connected to the Internet via a cellular link. The peers may share the cost of maintaining presence by taking turns serving as the proxy and accessing the cellular network. We develop protocols in which mobile devices form groups of peers within ad hoc networks for the purpose of maintaining an IM presence. Fig. 2.1 illustrates a case when a group of five peers use a proxy to share the IM message notification via a single connection. A single peer maintains a continuous wireless Internet connection for the purpose of maintaining presence information and providing message notification for a set of peers. This proxy server may migrate to a new peer, yet the same IM message notification capability is provided to the group. CHUM allows users to have virtual continuous IM presence capability while reducing telecommunication costs and power consumption.

Details of CHUM network architecture and notification protocols are presented in the chapter 4.

2.2.2 Sharing General Message/Event Notification Channel

Instant messaging service is a typical *"always on, anywhere, mostly idle"* application, which requires "always on, anywhere" Internet access, and yet may incur little network activity. It requires a continuous connection to the IM server to conduct the exchange of IM messages and *presence* information. However, little data is transmitted.

There are many other similar services available in the area of wireless communication. For example, advertisements that are based on a user's geographic location (LBA -

Location Based Advertising) are becoming an important mobile service in telecommunications. This service enables an advertiser to provide leisure and entertainment information, traffic reports, maps and directions and promotions to a customer when he/she is most likely to buy. The subscriber of the LBA should maintain an Internet connection in order to have "real-time" tracking of the location of the customer and to receive the advertisements. Other possible services include "always on" e-mail, stock quote update notifications that are time sensitive, on-line auction, corporate event/alert notifications, and weather updates. In general, event notification services, such as Elvin [16], could benefit from the "always on, anywhere, mostly idle" Internet access.

As we have discussed in the previous section, battery power consumption and network access costs inhibit the widespread deployment of those "always on, anywhere, mostly idle" mobile services. Powerful PDAs may only have battery power for a few hours of continual usage. WWAN costs are expected to charge users on the basis of data traffic generated or the time connected to Internet, which may be significant. Many mobile applications need to maintain continuous active WWAN connections, which incur continuous high energy consumption[2]. Since the amount of network traffic may be small, most of the energy is used to maintain idle connections.

In order to reduce power consumption while maintaining continuous network access, we extend the idea of CHUM to more efficiently support a broader set of "always on, anywhere, mostly idle" mobile services. Similar to the instant messaging service, the

[2]The Sierra Wireless AirCard 555 CDMA2000 1x consumes 450mW in IDLE mode and 3400mW in TRANSMIT mode [17].

cooperating devices form a peer-to-peer network to share a single continuous cellular link. Only one peer at a time is required to serve as a *proxy* to be connected to the Internet via a cellular interface. The other peers may receive information from Internet via their WLAN interface from the peer serving as a proxy. Cooperating peers take turns to be the proxy.

The details of group formation, message notification and proxy migration protocols and algorithms are described in the chapter 5.

2.2.3 QAWBA: QoS Aware Wireless Bandwidth Aggregation

Wireless users may demand for the same Quality of Service (QoS) for applications that are currently available on today's wired networks. This requirement can not be met by the cellular network and the IEEE 802.11 based network alone. The cellular network provides relatively low throughput and cannot meet the bandwidth requirements of many multimedia applications. The latest commercial deployment of 1xEV-DO offers only 38.6Kbps to 2.4Mbps depending on the signal strength, while the IEEE 802.11b standard can provide 1-11Mbps and the IEEE 802.11a/g standards can provide up to 54Mbps. However, the IEEE 802.11 based network can cover very limited areas, while the network access provided by the cellular network is virtually "anytime, anywhere."

In order to meet the high availability and high bandwidth QoS requirements at the same time, we present a novel integrated network architecture for *QoS Aware Wireless*

Bandwidth Aggregation (QAWBA) that utilizes the cellular interface and the IEEE 802.11 based interface to take the advantage of both networks. The basic idea of QAWBA is that cooperating mobile nodes form a MANET using the IEEE 802.11 based interface operating in an ad hoc mode in order to share their cellular network connections. Several low bandwidth cellular network connections are aggregated to meet the high bandwidth QoS requirement of multimedia applications for which a single cellular connection is insufficient. For a mobile node (*client*) that requires bandwidth higher than its own available cellular capacity, several other nodes in the same MANET may function as "*proxies*" by contributing their idle cellular connections to the client. The traffic is forwarded by the proxies to the client in the MANET via IEEE 802.11 interfaces.

Since the cellular and the IEEE 802.11 networks utilize different frequencies, the traffic on the two networks will not interfere with each other. For a single node, as many proxies can be used as the IEEE 802.11 based network is able to support. Thus with QAWBA, it is possible to provide similar high bandwidth as the IEEE 802.11 based network while keeping the high availability of the cellular network.

Figure 2.2 shows an example of QAWBA, in which five mobile nodes form a MANET. The client node C executes an application requiring 500Kbps bandwidth, which can obtain only 300Kbps from its cellular link. A and D act as proxies to forward a portion of the total traffic to C. The 500Kbps traffic flow is split into three flows in the base station, and forwarded to C via different paths. Thus, with the help of nodes A and D, C is able to receive the required 500Kbps bandwidth by aggregating three flows, which would not be possible under one single cellular connection.

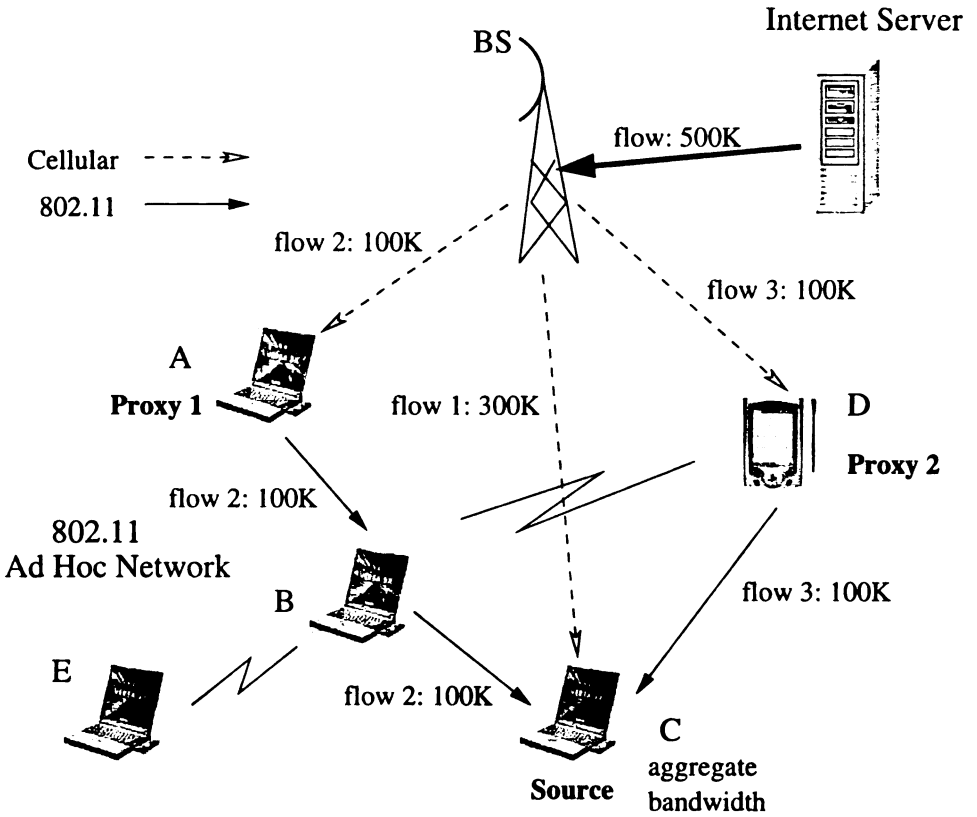


Figure 2.2: Example of QAWBA

In the chapter 6, we present the details of the routing protocols, algorithms and the results of the performance evaluation.

2.2.4 Cooperative Multipath Parallel File Downloading

Although IEEE 802.11-based WLANs provide much higher bandwidth in comparison to a cellular network, its small coverage area (up to 250m) makes it impossible for an "always on" Internet connection. When moving outside the range of WLANs, mobile users experience significant performance degradation with the bandwidth limitation on cellular data networks. Significant delay is expected when downloading a large file, such as a MP3 music file from an Internet server via the slow cellular link, while

the high speed IEEE 802.11 interfaces are idle. On the other hand, bursty Internet usage patterns result as idle cellular links most of the time.

We present a novel cooperative parallel file downloading scheme, which integrates the cellular data network and the IEEE 802.11-based ad hoc network. The scheme provides higher bandwidth and reduces file download latency of the cellular data networks. The basic idea is that the mobile nodes cooperate and share their idle cellular links via a MANET formed from their IEEE 802.11-based interfaces. In the scheme, a file is split into small portions. Several neighbor nodes act as proxies to share the burden of file downloading with the destination node by downloading portions of the file from the Internet server via their idle cellular links and forwarding them to the destination via the IEEE 802.11-based MANET. Thus, the nearby idle cellular links and the high speed IEEE 802.11-based MANET may be utilized to create multiple paths from the Internet server to the destination. Multiple portions of the file are downloaded in parallel via different paths to the destination node, which can significantly reduce file download latency.

Figure 2.3 shows an example of cooperative parallel file downloading. In this example, five mobile users A, B, C, D and E form a MANET. C wants to download a file from the Internet server, and is referred to as a *client*. A and D act as *proxies* to contribute their idle cellular links to support C's file downloading. B is a relay node (*forwarder*) in the MANET, contributing to the system by forwarding packets from the proxy A to the client C. The file is split into small portions and downloaded in parallel via three different paths: from the base station to C directly; from the base station to

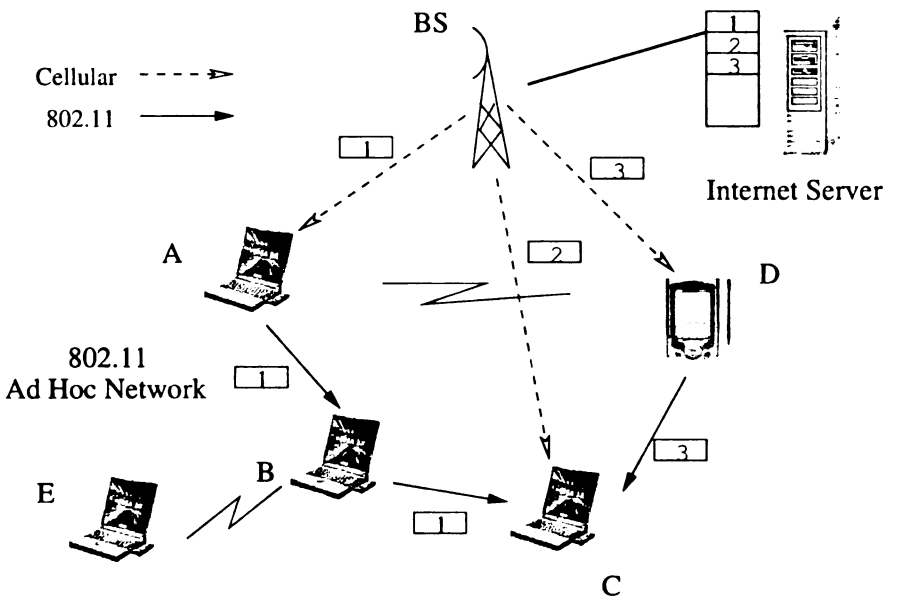


Figure 2.3: Cooperative Parallel File Downloading

proxy A, forwarded to forwarder B, which is further forwarded to client C; from the base station to proxy D, then forwarded to client C.

Parallel downloading has been used as a technique to improve the performance of downloading large web files. In [18], Rodriguez and Biersack study a dynamic parallel-access scheme to accelerate web downloads by connecting to multiple mirror servers and downloading different parts of the file from each of them. Using a dynamic parallel-access scheme, a client experiences dramatic speedup in downloading documents, and the load is shared among servers without the need for a server selection mechanism. The effects of aggressive implementation and poor scalability are studied in [19]. Recent development of peer-to-peer networks also adopt parallel download techniques to accelerate file download from different users sharing the same document, such as BitTorrent [20]. Some special erasure codes, such as Tornado Codes, can be used in parallel downloading [21, 22]. With erasure codes, a receiver can gather an

encoded file in parallel from multiple sources. As soon as enough packets arrive from any combination of the sources, the receiver can recover the original document. The goal of the cooperative parallel downloading scheme is independent of the parallel-access algorithm used. The techniques and algorithms developed in these research activities can be used to enhance cooperative parallel file downloading.

In the chapter 7, we present the details of the network protocols, algorithms and the results of performance evaluation.

Chapter 3

Literature Review

The effort of CIWNA is aimed to utilize the peer-to-peer MANET formed by the WLAN interfaces to share the resource provided in the cellular data networks. The idle cellular data links are shared among a number of mobile users in the spirit of peer-to-peer resource sharing by forming a mobile ad hoc network. Such network architecture is built on top of existing wireless technologies and the concept of the peer-to-peer sharing. Trust and reputation management system is used to promote cooperation in the system and prevent free-riding of selfish users. In this chapter, we will present a literature review of technologies and research in the related areas. In section 3.1, we review three important wireless technologies and discuss some efforts in the integration of cellular WWANs and WLANs. The literature review of peer-to-peer (P2P) computing is presented in section 3.2. The research of trust and reputation management system in P2P networks is discussed in section 3.3.

3.1 Integrating Cellular Data Networks and Wireless LANs

3.1.1 Overview of Wireless Technologies

Three different wireless technologies are widely used in the current mobile systems: wireless wide area networks (WWAN), wireless local area networks (WLAN) and wireless personal area networks (WPAN). Each has different wireless features and potential applications.

The goal of WWANs is to provide service anywhere in a metro area, state, country, or even continent. The majority of the traffic on WWAN networks today is voice oriented but the demand for data and Internet services is becoming more pronounced. They give the user the ability to move around anywhere while still remaining connected to the Internet or company intranet. The WWANs require an extensive infrastructure support, which may cost the carriers hundreds of billions of dollars to deploy on a broad scale. The range of a WWAN is typically measured in dozen or hundreds of miles. Communication over such distances requires high-power transmissions and a costly license for a specific frequency band. Therefore the WWAN connection to the mobile telephone is expensive. On the other hand, high-power transmission leads to trade-offs between power consumption and data rates in WWANs. Typical data rates for today's cellular networks are relatively slow in comparison to wired LAN connections. In summary, the WWAN connections are costly, high power consumption, low throughput and wide range.

The WLANs extend the traditional LANs with wireless interfaces. The trend toward wireless data services also drive the deployment of location-specific hotspot services. In comparison to 2.5G and 3G cellular systems, the WLANs and hot spot services are relatively inexpensive to deploy. The services are often available in variable locations with limited ranges. The WLANs use unlicensed spectrum, which is free of charge to users. The advent of the IEEE 802.11 standards have achieved nearly Ethernet-equivalent speeds. Some WLAN systems may work under two different modes: infrastructure mode and peer-to-peer ad hoc mode. This feature can be used to create a peer-to-peer mobile ad hoc network. The Wireless Ethernet Compatibility Alliance (WECA) is created to focus on wireless-fidelity (Wi-Fi) interoperability among equipment vendors. Today, the WLAN interface has been integrated into mobile PCs by major notebook PC makers for the mass market. In the near future, it will become a standard interface for other mobile devices, such as PDAs and smartphones. In summary, the WLAN connections are popular, flexible, inexpensive, relative low power consumption, high speed and limited transmission range.

A WPAN is a personal area network for interconnecting devices that operate within a short range of an individual person's workspace. WPANs can be used to replace cables between computers and their peripherals, or to establish location aware services with an extremely low power consumption. They may support traditional computing devices as well as new IP appliances, including "wearable" computers. The WPANs also use unlicensed spectrum, for example the 2.4GHZ frequency band. Like WLAN, WPAN interfaces may also be used to form mobile ad hoc networks within a short

range. In summary, the WPANs connections are extremely low-power, short range and relatively high throughput.

WWAN: Cellular Data Networks

WWANs have been around for a number years, but data support was rather limited and expensive until the late 1990s. Major advances were seen in the area of standard efforts with regards to 2.5G and 3G WWAN services during the mid-1990s and early 2000s. Today's WWAN technologies are based on infrastructure in common use for cellular communications, such as GSM, TDMA, and CDMA. Existing 2G systems offer low data rates, such as 14.4Kbps circuit-switched services. Improved solutions (called 2.5G) are emerging. As carriers enhance their networks to prepare for 3G, they can offer early adopters upgraded data services. (2.5G because they fall between current 2G and 3G technology.) These intermediary technologies, such as GPRS and CDMA2000 1X, deliver data at rates between 115 and 307 Kbps. When 3G is finally deployed, 2.5G systems can be replaced.

3G mobile communication is a concept outlined in a set of proposals called *International Mobile Telecommunications-2000* (IMT-2000) to define an anywhere-anytime standard for the future of universal personal communications. It seeks to provide up to 2Mbps for indoor communication and 144Kbps for outdoor communication. The ITU has given support to two 3G technologies: W-CDMA and CDMA2000. North American providers have leaned in favor of CDMA2000 and EDGE. Multiple groups are involved in standards development, such as ETSI, IETF, ANSI T1, 3GPP,

Table 3.1: Key WWAN Standards

Technology	Generation	Description
TDMA	2G	The standard used by AT&T services
GSM	2G	The most widely used wireless standard in Europe, based on TDMA
CDMA	2G	The leading air interface in North America
GPRS	2.5G	Supports midrange data service to TDMA and GSM devices. Maximum bit rate is 115Kbps.
CDMA2000 1X	2.5G	Provides CDMA users with data rates as fast as 307Kbps.
EDGE	3G	Enhanced TDMA for data rates between 384Kbps to 2Mbps
CDMA2000 3X	3G	Provides data services to CDMA devices at a bit rate as fast as 2Mbps
W-CDMA	3G	ITU's official 3G migration path for TDMA networks

3GPP2 and so on. Four systems for 3G mobile communications have been proposed after major industry involvement:

- W-CDMA UMTS FDD (frequency-division duplexing);
- CDMA2000;
- W-CDMA UMTS TDD (time-division dulpexing);
- UMC-136 (ITM-SC single-carrier EDGE);

Table 3.1 lists major WWAN standards and their belonging generation.

The cellular system replaces a large zone with a number of small cells, with a single base station (BS) covering a fraction of the area. Figure 3.1 shows an example of cellular system consisting of small cell zones with all mobile hosts located in a cell

being served by a BS. The BS is located at the center of the cell. The cell area is determined by the signal strength of within the region. In the ideal radio environment, the shape of a cell can be circle around the BS. For all practical purpose, the cell is approximated by a hexagon (see Figure 3.1). The hexagon is a good approximation of a circular region, and it allows a larger region to be divided into non-overlapping hexagonal subregions of equal size, with each one representing a cell area.

A mobile station (MS) needs to communicate with the BS of the cell where the MS is currently allocated, and the BS acts as a gateway to the rest of the world. In any cellular scheme, four simplex channels are needed to exchange synchronization and data between the BS and MS: the forward (downlink) control channel; the reverse (uplink) control channel; the forward (downlink) traffic channel; and the reverse (uplink) traffic channel. Only a limited amount of bandwidth is allocated for the wireless service. Three basic multiplexing techniques are employed to increase the effectiveness of the overall system: frequency division multiple access (FDMA), time division multiple access (TDMA) and code division multiple access (CDMA).

Wireless LAN technologies

The most well known representatives for WLANs are based on the standards of IEEE 802.11 [23] and HiperLAN [24] with all their different variations.

HiperLAN stands for high-performance LAN. It is derived from traditional LAN environments and can support multimedia data and asynchronous data effectively at high rates (23.5 Mbps) with a coverage range of 50m and mobility less than 10 m/s.

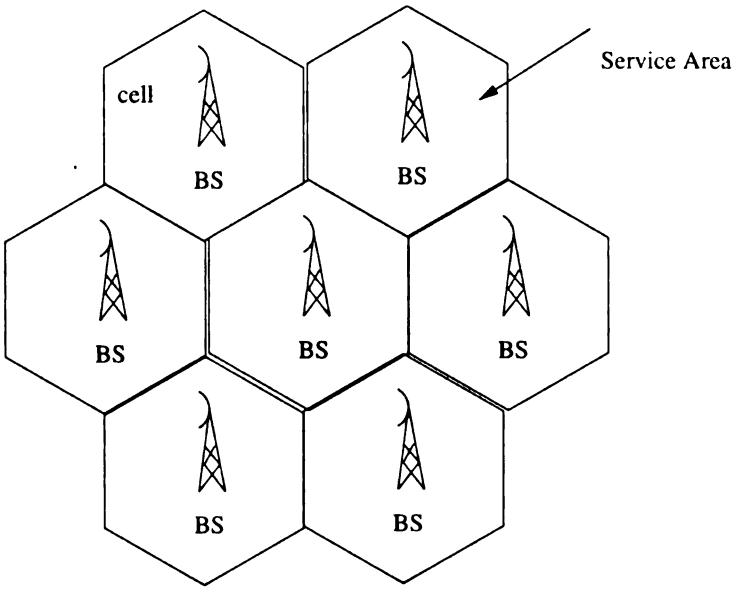


Figure 3.1: Cellular Cell Structure

IEEE 802.11 is an evolving family of standards developed by a IEEE working group. 802.11b is the most popular standard. It specifies a physical (PHY) layer and medium access control (MAC) layer providing a basic rate of 11Mbps and and fall-back rate of 5.5Mbps in the 2.400 GHz to 2.4835 GHz frequency range. The modulation method selected for 802.11b is known as complementary code code keying (CCK), which allows higher data speeds and is less susceptible to multipath-propagation interference in comparison to historical phase-shift keying (PSK) in 802.11. Networks using 802.11a operate at radio frequencies between 5.725 GHz and 5.850 GHz providing data rate as high as 54 Mbps. The specification uses a modulation scheme known as orthogonal frequency-division multiplexing (OFDM). There is less interference with 802.11a than with 802.11b since 802.11a can provide more available channels and the 2.4 GHz band used by 802.11b is shared by various other household appliances and medical devices. 802.11g standard operates at the same radio frequency band as 802.11b, which makes it compatible with 802.11b devices. It employs the modulation

scheme used in 802.11a, OFDM, to obtain higher data speed. Other important 802.11 standards include 802.11d (enhancement for global roaming), 802.11e (enhancement for QoS), 802.11h (enhancement to 802.11a to offer wireless transmission over relative short distances), and 802.11i (enhancement for security).

The IEEE 802.11 MAC protocol is based on Collision Sense Multiple Access/Collision Avoidance (CSMA/CA). A mobile host desiring to transmit data senses the medium first. It is only allowed to transmit when the medium is not busy. But there is always a chance of multiple stations simultaneously sensing the medium as being free and transmitting at the same time, causing a collision. The 802.11 standard uses a Collision Avoidance (CA) mechanism together with a positive acknowledge scheme to reduce the collision probability. It works as follows:

- A mobile host wanting to transmit senses the medium first. If the medium is free for a specified time (Distributed Inter Frame Space (DIFS) in the standard), then the mobile host captures the channel and transmits all pending data packets.

- Otherwise, if the medium is busy, the mobile host defers transmission and enters into the backoff state. The mobile host randomly selects a backoff interval in the range of $[0, cw]$ (cw is the length of the contention window), and waits for this number of idle slots before accessing the medium again. If a collision occurs, the length of the contention window (cw) will be doubled (exponential backoff). After a successful transmission, the value of cw will be restored to its initial value cw_{min}.

- The receiving host checks the CRC of the received packet and sends an acknowledgment packet (ACK). Receipt of the acknowledgment indicates to the transmitter that no collision has occurred. If the sender does not receive the acknowledgment then it retransmits the packet until it receives an acknowledgment or the packet is thrown away after a given number of retransmissions.

Power Saving in IEEE 802.11 The power saving mechanisms adopted by IEEE 802.11 standard enable stations to switch to sleep mode for long periods of time without losing information. The techniques used in infrastructure mode and ad hoc mode are quite different.

In infrastructure mode, there is a base station called an *Access Point* (AP) to monitor the mode of each mobile host. The main idea is that a mobile host may switch to sleep mode and inform the AP of this decision. The AP maintains a continually updated record of the stations currently working in *Power Saving* (PS) mode, and buffers the packets addressed to these stations. The AP periodically transmits a beacon indicating which nodes have packets buffered in the AP. The PS stations will wake up periodically to receive the beacon. If they have packets waiting in the AP, then the PS stations stay awake and request the packets by sending a *PS-Poll* request to the AP. In response to this polling request, the AP will send the buffered data to the stations.

In the ad hoc mode, without the help of the centralized AP, it is much more difficult for energy efficient design. Time is divided into beacon intervals. The power saving (PS) stations wake up periodically in a short interval called the *ATIM window* at

the beginning of the beacon interval. It is assumed that hosts are fully connected and all synchronized, so the ATIM windows of all PS hosts will start at about same time. At the beginning of the ATIM window each station competes to transmit a beacon frame using the standard backoff algorithm. Any successful beacon frame will be used for synchronization. After the beacon frame, the hosts with buffered packets can send a ATIM frame to their intended PS receivers and remain awake for the rest of the beacon interval. On reception of the ATIM request, the station will reply by sending an ATIM ACK and also stay up for the remaining period. The packets will be transmitted based on the normal DCF access procedure after the ATIM window finishes. The station that does not receive an ATIM request during an ATIM window, and has no pending packets to transmit may switch to doze mode during the rest of the beacon interval. Fig. 3.2 shows an example of 802.11 ad hoc power saving mechanism, where host A wants to transmit a packet to host B. The ATIM frame and the ACK are transmitted during ATIM window. Host A and B stay awake after ATIM window finishes the data transmission. Host C only awakes during the interval of the ATIM window and falls back to doze after the ATIM window.

In the aspect of energy efficiency, there are several disadvantages of the IEEE 802.11 standard. First, energy efficiency is not the design goal of the IEEE 802.11 standard. It does not perform well as those protocols that aim at energy efficiency [25]. Second, the PS mode of the IEEE 802.11 results in delays at the mobile hosts and it may affect the quality of service. Third, the PS mode of IEEE 802.11 is designed for a single-hop or fully connected ad hoc network. It assumes that there is a clock syn-

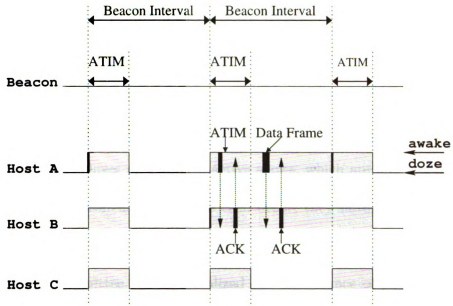


Figure 3.2: An example of 802.11 ad hoc power saving protocol.

chronization among all stations. This clock synchronization is complex for multihop ad hoc networks.

Wireless PAN technologies

The key characteristics of a WPAN are short range, low power, low cost and small network. The best example representing WPANs in the recent industry standard is Bluetooth [26]. Companies like Ericsson, Intel, IBM, Nokia, and Toshiba started this in 1998 by establishing a Bluetooth Special Interest Group. It is optimized by design for WPANs. Low-cost, low-power, radio-based wireless links eliminate the need for short cables between small personal devices. The concept of Bluetooth has evolved to provide a universal standard for short-range RF communication of both voice and data. Bluetooth utilizes the unlicensed ISM band at 2.4GHz. A typical Bluetooth device has a range of about 10 meters. The communication channel supports data

and voice with a total bandwidth of 1Mbps. Bluetooth devices can interact with other Bluetooth devices in several ways. One of the devices may act as the master and (up to) seven others as slaves. The master and the slaves form a *piconet*, which is a network of devices connected in an ad hoc fashion. The master regulates channel access for all active slaves and other inactive slaves, which are referred to as parked nodes. When two piconets are close to each other, they have overlapping coverage areas. This scenario, in which nodes of two piconets intermingle, is called a scatternet.

The IEEE 802 committee has also realized the importance of short-range wireless networking and initiated the establishment of the IEEE 802.15 working group to standardize protocols and interfaces for wireless personal area networking. The IEEE 802.15 working group is formed by four task groups (TGs):

- The IEEE 802.15 WPAN/Bluetooth TG1: Support applications which require medium-rate WPANs, such as Bluetooth.
- The IEEE 802.15 Coexistence TG2: Develop specifications and recommended practices to facilitate the coexistence of WPANs (802.15) and WLANs (802.11).
- The IEEE 802.15 WPAN/High Rate TG3: Charactered to draft a new standard for high-rate (20Mbps or greater) WPANs.
- The IEEE 802.15 WPAN/Low Rate TG4: Provide a standard for ultra-low complexity, cost, and power for a low-data-rate (200kbps or less) wireless connectivity among inexpensive fixed, portable, and moving devices. Location awareness is being considered as a unique capability of the standard.

3.1.2 MANET: Mobile Ad Hoc Network

A mobile ad hoc network (MANET) is the mobile devices that comes together to form a network as needed without any support from the existing infrastructure or fixed stations. It is an autonomous system of MSs (also serving as routers) connected by wireless links, forming a communication network modeled in the form of an arbitrary communication graph. The nodes are free to move and the network topology may change dynamically in an unpredictable manner. In contrast to the well-known single-hop cellular network, in which communication between two mobile nodes relies on the wired backbone and the fixed base stations, MANETs are basically peer-to-peer multihop mobile wireless networks. Information packets are transmitted in a store-and-forward manner from a source to an arbitrary destination, via intermediate nodes. As the nodes move, the resulting change in network topology must be made known to the other nodes so that outdated topology information can be updated or removed [27].

Routing Protocols

Routing protocols in MANETs are quite different from wired ones due to high mobility of hosts and high rates of link failure/repair caused by movement. Many routing protocols have been proposed for MANETs. These protocols may be categorized as pro-active protocols and reactive protocols. In pro-active protocols, mobile hosts periodically exchange routing control packets and update their routing tables. Traditional link-state and distance-vector routing protocols are pro-active. Examples

of MANET pro-active protocols include Optimized Link State Routing (OLSR) [28] and Destination-Sequenced Distance-Vector (DSDV) [29]. Reactive protocols are also called on-demand protocols. The route selection process is initiated by the sender only when needed. The Ad hoc On-Demand Distance Vector (AODV) [30] and Dynamic Source Routing (DSR) [31] are two good examples of on-demand protocols. The Zone Routing Protocol (ZRP) [32] is an example of a hybrid protocol. ZRP pro-actively maintains state information for links within a short distance from any given node and on demand protocol is used for determining routes to far away nodes.

The on-demand protocols result in less control packets and are more adaptive to topology changes, but lead to longer route set up delay before a packet may be sent. On the other hand, the pro-active protocols require more control packet exchanges, but do not incur the additional route set-up delay. However, it is possible that the pre-computed route is incorrect due to host mobility and link failure, thus leading to potential lost packets. Performance comparisons of routing protocols for MANET are available in [33, 34, 35, 36].

Some routing protocols use cluster-based schemes. In these protocols, a leader is elected for each cluster of nodes with some special responsibilities. The advantage of cluster-based protocols is that several heuristic methods can be employed to improve the protocols' performance. However, there is always a high overhead for cluster maintenance and the leader has to handle additional traffic.

Different protocols may differ in the way clusters are determined, the way cluster head (leader) is chosen, and the duties assigned to the cluster head. Two examples of cluster based routing scheme are Clusterhead Gateway Switch Routing (CGSR) [37]

and Core-Extraction Distributed Ad Hoc Routing (CEDAR) [38]. In the CGSR protocol, all nodes within a cluster is communicated via a clusterhead. Routing between different clusters uses a clusterhead-to-gateway approach. The cluster head is chosen by a Least Cluster Change (LCC) clustering algorithm, which changes cluster head only when two cluster heads come into contact or when a node moves out of contact of all other cluster heads. In the CEDAR protocol, a subset of nodes in the network is identified as the core. Each node in the network must be adjacent to at least one core node. Each core node determines paths to nearby core nodes by means of a localized broadcasting. Thus the link state propagation occurs among core nodes.

QoS aware routing protocols Much research has been dedicated to solve Quality of Service issues for MANETs [39, 40]. INSIGNIA [41] is an effort to design a cross-layer framework to support QoS routing in ad hoc networks. INSIGNIA uses an in-band and soft-state based signaling protocol to support fast reservation, restoration and end-to-end adaptation of QoS parameters.

Several protocols have been proposed to address on the QoS aware routing in MANETs. The CEDAR algorithm [38] uses a set of ad hoc nodes called the *core* to establish a QoS aware route from the source to the destination. Information regarding the availability of bandwidth propagates among core nodes using a link state protocol. In AQDR [42], the source uses limited flooding for route establishment. The destination of the route is responsible for QoS violation detection and the destination-initiated recovery process begins when a QoS violation is detected.

Ticket based probing [43] is one of the flooding based QoS routing discovery algorithms. It assumes an imprecise state model and tries to reduce the amount of flooding routing messages by issuing logical tickets. When a probe arrives at a node, the tickets contained in the probe can be split to its neighbors. When one or more probes arrive at the destination node, the routing path is known and the networking information can be used to establish a quality aware path. The ticket based routing is again extended by Liao, et al. to find a multi-path QoS routing scheme between the source and the destination [44].

Power aware routing protocols Typical metrics used to evaluate ad hoc routing protocols are shortest-hop, shortest-delay and locality stability [45]. However these metrics may have a negative effect in wireless network because they may result in the overuse of energy resources of a small set of mobile hosts, thus decreasing the lifetime of those mobile hosts. The shortest-hop routing protocols are not applicable from the perspective of energy consumption. The routing optimization optimization criteria should also include the metrics of energy consumption. The power-aware routing protocol presented in [45] uses a shortest-cost routing scheme. It tries to minimize energy consumed per packet, maximize time to network partition due to energy depletion, maximize duration before a node fails due to energy depletion, and minimize variance in power levels across mobile hosts. A weight is assigned to each link, which may be a function of energy consumed when transmitting a packet on that link, as well as the residual energy level (low residual energy level may correspond to a high cost). The route is selected with the smallest aggregate weight. Although

the packets may be routed via longer paths, the paths contain mobile hosts that have greater amounts of energy reserves. Also, energy can be conserved by routing traffic through lightly loaded mobile hosts because the energy expended in contention and retransmission is minimized.

Power Saving Techniques in MANET

Wireless devices have maximum utility when they can be used "anywhere at anytime". One of the greatest limitation to that goal is finite power supplies. Power efficiency is considered as one of the most challenging problems in wireless communication. It should be a crucial design consideration through all layers of the protocol stack [46]. The source of power consumption, with regards to network protocols, may be classified into two types: *communication related* and *computation related.* Communication related power consumption involves the usage of transceivers in mobile hosts. Computation related power consumption is concerned with protocol processing. There exists a potential tradeoff between computation and communication costs. Complex protocols tend to have better performance with respect to communication costs. However it may generally result in higher computational requirements. Energy efficient protocols should get balance between these two costs.

Power management system The most direct solution for the shortage of power is increasing the battery capacity. However, battery technology has not experienced significant advancement in the past 30 years. Progress has been slow in comparison to other subsystems in wireless devices and is unable to keep pace with the growth of

power consumption. The other direction is to decrease energy consumption and make the best use of the currently available power. Some power management schemes have been developed in current PC systems. *Advanced Power Management (APM)* [47] is the first widely adopted power management scheme in a PC system. It is a BIOS-based power management system for CPU and devices. *Advanced Configuration and Power Interface (ACPI)* [48] improves APM by moving the power management responsibility to the operating system to overcome the limitation of the BIOS.

General guidelines in energy efficient network protocols Although the operating system provides some power management schemes, significant additional power savings still may result by incorporating low-power strategies into the design of network protocols. A summary of research done on energy efficient networks is available in [49], which presents general guidelines for energy efficient network protocol design: powering off network interfaces whenever possible; eliminating collisions within MAC layer; reducing turnaround between transmit and receive modes [50]; power aware scheduling at the base station [51]; efficient error control strategy [52]; topology control [53, 54]; and power aware routing [45, 55].

Energy efficient MAC protocol design The MAC sublayer is responsible for providing reliability to the upper layer for point-to-point connections established by the physical layer. The shared wireless channel is allocated by MAC protocols among all mobile hosts. Wireless MAC protocols may be classified into two different

categories: centralized protocols for infrastructure based networks and distributed protocols for ad hoc networks.

The *Energy Conserving-Medium Access Control (EC-MAC)* protocol [56] is defined for an infrastructure network with a single base station serving mobile hosts in its coverage area. The main idea of EC-MAC is based on reservation and scheduling for the goal of low energy consumption and QoS provision. A centralized scheduling algorithm is used in the base station to eliminate collisions and provide power efficient bandwidth allocation. The centralized scheduler also optimizes the transmission so that individual hosts will transmit and receive within contiguous transmission slots. A comparison of EC-MAC with IEEE 802.11 MAC and other MAC protocols is done in [25]. The result of simulation shows that reservation and scheduling will avoid collision and thus reduce power consumption. However, the centralized design of EC-MAC also makes it unsuitable for an ad hoc network. Nevertheless, the definition of EC-MAC may be extended to an ad hoc network by letting the mobile hosts elect a coordinator to perform the functions of the base station. The overhead to maintain the group may be very high. Also the coordinator will spend additional power for scheduling.

While the EC-MAC protocol is designed primarily for infrastructure networks, the *Power Aware Multi-Access (PAMAS)* [57] protocol is dedicated for ad hoc networks. This protocol is a combination of the original MACA protocol in [58] and the idea of using a separate signal channel for RTS/CTS control packets. Power conservation is achieved by turning off wireless interfaces for mobile hosts that are not able to receive and send packets. The separate signal channel enables nodes to determine when and

for how long to turn off. Every node makes the decision to power off independently when (i) it has no packet to transmit, but one of its neighbors begins transmitting packets that are not destined for it; or (ii) there is at least one of its neighbor-pair that is communicating. The node determines the length of power off interval by reading the information in RTS/CTS exchange during the packet transmission in its neighborhood. The advantage of the PAMAS protocol is that this protocol is designed for ad hoc networks with energy efficiency as the primary design goal. Simulations show that 10% to 50% power saving may be achieved. However, the PAMAS protocol needs a separate control channel, which may not available in all wireless environments.

Error control The high error rates in wireless networks make it impossible to provide a totally reliable wireless link. A low-power error control protocol should avoid persistence in retransmitting data, trade off number of retransmission attempts for the probability of successful transmission, and inhibit transmission when channel conditions are poor [49]. There are two techniques used for the error control in the logical link control (LLC) sublayer: *Automatic Repeat Request (ARQ)*, where error-detecting codes are used and *Forward Error Correction (FEC)*, where error correcting codes are used. Both ARQ and FEC waste bandwidth and consume power resource due to retransmissions and overhead. With FEC, the node pays a priori battery power consumption overhead and packet delay in computing the FEC code and in transmitting the extra code bits. In return one receives a reduced packet retransmission probability. On the other hand, with ARQ, the node pays for the battery energy later due to the ACK packets and retransmissions of the entire packet.

Based on the above discussion, the FEC scheme is better than ARQ in high error rates or larger packet sizes from the perspective of power consumption. There is always a balance between throughput, reliability, security and energy efficiency in the selection of error control schemes.

An adaptive error control scheme with ARQ is studied in [59, 60]. In these papers, a probing protocol is adopted to slow down data retransmission when the channel conditions are bad. When the channel conditions deteriorate, the transmitter enters into a probing mode, and sends a short probing packet repeatedly. The probing mode will continue until a properly received ACK is encountered. After the channel conditions improve, the transmitter switches back to the normal mode and restarts transmission from the point it was interrupted.

The energy efficient error control scheme presented in [52] provides an adaptive strategy with the combination of ARQ/FEC. The basic idea is that we cannot use a single error control scheme for all traffic types and channel conditions. Each packet stream maintains its own time-adaptive customized error control scheme based on certain set-up parameters and a channel model estimated at run-time. The parameters include packet size and QoS requirements. They are used to select the combination of ARQ (Go-Back-N, Cumulative Acknowledgment or Selective Acknowledgment) and FEC. The error control scheme will change dynamically as channel conditions change over time. The research is extended further to size dynamically the MAC layer frame in [61].

Power control In wireless communication, transmission power has a strong impact on the bit error rate and inter radio interference. These are typically contradicting factors. In [62], power control is adopted to reduce interference and improve throughput on the MAC layer. Topology control techniques are developed to determine transmission power of each mobile host so as to determine the best network topology [63, 54, 53].

From the perspective of power saving, the node should turn off its radio when it is not in use. However, in a MANET, a node must stay awake not only to receive packets addressed to it, but also to participate in the process of routing to forward packets for each other. Span, an energy efficient coordination algorithm, is developed by [63] for topology maintenance in ad hoc wireless networks. In Span, a distributed randomized algorithm is developed for nodes to make local decisions on whether to sleep or to join a forwarding backbone as a coordinator. Span is built on the observation that not all nodes are needed at any time for packet forwarding. Each node bases its decision on an estimation of the number of nodes that can benefit from it being awake and the amount of energy available to it. Periodically, a non-coordinator node determines whether it should become a coordinator or not. The coordinator eligibility rule ensures that the entire network is covered with enough coordinators. On the other hand, a node will withdraw as a coordinator if every pair of its neighbors can communicate directly or via other coordinators. To achieve fairness, the withdrawal will happen if the node has been a coordinator for some period of time and every pair of neighbor nodes may reach each other via some other neighbors, even if those

neighbors are not currently coordinators. This rule gives other neighbors a chance to become coordinators.

Multiple nodes may decide to become coordinators at the same time. The contention is resolved by delaying coordinator announcements with a randomized backoff algorithm. This delay is a function of the number of nodes in the neighborhood that can be bridged using this node, and the amount of energy it has remaining. The possibility of being a coordinator falls as the node uses up its energy. The election algorithm rotates coordinators among all nodes of the network. The simulation results show that Span not only preserves network connectivity, but also preserves capacity, decreases latency, and provides significant energy savings.

3.1.3 Hybrid Network Architectures

The integration of cellular WWANs and WLANs has drawn considerable attention from the research and commercial communities [64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 7].

4G wireless systems

Some researchers and industry companies are interested in integrating different existing wireless network architectures to form the future generation wireless systems (4G and beyond) [76]. The performance of current 2.5G and 3G systems may not be sufficient to meet needs of future high-performance applications like multimedia full-motion video, wireless teleconferencing. Multiple standards for 3G makes it difficult to roam and inter-operate across networks. The 4G wireless networks will move be-

yond the limitations and problems of 3G which is having trouble getting deployed and meeting its promised performance and throughput. They are envisioned to provide ubiquitous high-speed access over heterogeneous radio technologies.

4G wireless networks will be all digital IP packet networks, covering both data and voice over IP. In comparison to wide-area cell-based 3G wireless networks, 4G systems will be an integration across different network topologies, including wireless WANs, wireless LANs (IEEE 802.11a/b/g, IEEE 802.15 and IEEE 802.16, Bluetooth), and fiber-based Internet backbones. Broadband wireless networks will be a part of this integrated network architecture. Currently, there are two groups in ITU defining the 4G: one is on high data rates (up to 100Mbps), the other is on open architecture.

Niebert et al. proposed a novel networking concept called the ambient network for a future wireless network [77]. The approach is to embrace the heterogeneity arising from the different network control technologies and to allow agreement for cooperation between networks on demand. A new end-to-end transport layer approach called *pTCP* is proposed in [78] to effectively perform bandwidth aggregation on multi-homed mobile hosts.

Multihop cellular networks

As mobile ad hoc network architecture has emerged as an important wireless and mobile communication paradigm, some research has been focus on developing multihop cellular networks by adding multihop mobile relays into the cellular networks [64, 66, 67, 68, 72]. From the cellular network model perspective, adding mobile multihop relay capability to a cellular network can increase system service coverage and may

also increase system capacity. Mobile terminals with poor signal reception can benefit from the alternative multihop relay path, this could provide greater throughput or better QoS. Within these projects, we can classify them into two categories. The first one involves the use of a single wireless interface for both the relay and infrastructure mode. Examples are ad hoc GSM cellular system (A-GSM) [79], Opportunity Driven Multiple Access [80] and Mobile Assisted Data Forwarding (MADF) [8]. The second one takes the advantages of two wireless interfaces, connecting to the cellular data network and the mobile ad hoc network simultaneously. The Unified Cellular and Ad-Hoc Network Architecture (UCAN) [67] and iCAR [9] fall into this category. The CIWNA system presented in this document can also be classified into this category.

The A-GSM and ODMA are two proposed cellular systems that support multihop wireless relay. A-GSM adds the relay capability to a second generation GSM network to enhance system coverage. The ODMA proposal to the Third Generation Partnership Project (3GPP) provides a relaying protocol to enhance cellular coverage and reduces radio transmission power in UMTS Terrestrial Radio Access (UTRA) time-division duplex (TDD). ODMA is used to maintain high data rates at the boundaries of a cell. Relaying seeds or terminals are deployed to relay traffic for mobile stations in the low-data-rate area (boundary) of the cell via multi-hop transmissions.

In MADF, an ad-hoc overlay is added into the fixed cellular infrastructure. The channel pool is divided into a set of fixed channels and a set of forwarding channels so that data packets can hop from "hot" cells to "cold" cells using the forwarding channels without going through the "hot" cell's base station in order to reduce delay

and increase capacity. The authors in [66] investigate a hybrid IEEE 802.11 network architecture with both DCF and PCF modes, again using one wireless interface.

In [71], the authors propose a new wireless network model called Multi-hop Cellular Network (MCN). The model involves mobile stations farther away from the base station communicating with the base station using a multihop path consisting of other mobile stations. There are two possible architectures of MCN: MCN-p and MCN-b. In MCN-p, the transmission power of the base station and mobile stations are reduced to achieve throughput increase and power reduction. In MCN-b, the transmission range remains the same, the number of base stations is reduced.

In [67], mobile users form an UCAN architecture using their 3G cellular links and IEEE 802.11-based ad hoc links. A relay proxy helps to forward packets from the base station to the clients with poor channel quality via high-bandwidth IEEE 802.11-based ad hoc links to improve throughput of the cellular network. The 3G base station scheduling algorithm is refined so that the throughput gains of active clients are distributed proportional to their average channel rate, thereby maintaining fairness. Two different kinds of proxy discovery protocols are proposed. In greedy proxy discovery protocol, neighboring mobile clients within one-hop IEEE 802.11b transmission range periodically exchange their average downlink channel rates by broadcasting a advertisement message. In on-demand proxy discovery, mobile clients do not proactively maintain their neighborhood information. Instead, the destination client reactively floods a request message within a certain range. A secure crediting mechanism is used to motivate users to participate in relaying packets for others. The results show the individual user's throughput could be improved up to 310% by UCAN. UCAN is

designed specifically for the 1xEV-DO (HDR) 3G cellular network, which limits its application. A similar two-hop-relay architecture is proposed by Hung-Yu and Wei to enhance the system capacity of existing WWAN systems and extend the system coverage of WLAN terminals [81]. This architecture can be considered as a system-level macro diversity techniques that utilizes temporal channel quality variation to achieve increased system capacity. Significant capacity gain is achieved in both a fixed-rate uplink CDMA system and a variable data rate downlink HDR-link system.

The iCAR [9] system addresses two problems for cellular networks: network capacity is limited by the cell boundary; bursty traffic is unevenly distributed among cells. The basic idea is to place a number of ad hoc relay stations (ARSs) at strategic locations, which can be used to relay signals between mobile hosts and base stations. Bursty traffic could be diverted from one congested cell to another one in order to circumvent congestion. East ARS and mobile host has two air interfaces, the cellular interface for communicating with a base station and the relay interface for communicating with a mobile most or another ARS. iCAR requires special kinds of relay stations to be placed by a network operator for packet relaying without utilizing the existing relay ability provided by IEEE 802.11-based network interfaces. Although it is useful for diverting bursty traffic to nearby idle cells, iCAR does not provide a way to improve utilization of the idle cellular link under light traffic load.

3.2 Peer to Peer Computing

With the pervasive deployment of computers, the trend to peer-to-peer systems has renewed interest in both industry and academic [10]. Some big industrial efforts include the P2P Working Group [82], led by many industrial partners such as Intel, HP and Sony; and JXTA [83], an open-source effort led by Sun. Many research projects are in progress at universities, such as Chord [84], OceanStore [85], PAST [86], CAN [87], and FreeNet [88].

"Peer-to-peer" (P2P) refers to "a class of systems and applications that employ distributed resources (such as computing power, data storage and content, and network bandwidth) to perform a critical function in a decentralized manner" [89]. It is an alternative to the centralized and client-server computing model, where a centralized single server or small cluster of servers provide service for many clients. Typical P2P systems reside on the edge of the Internet or in ad-hoc networks. One of the key ideas resides in P2P is about sharing. A peer gives some resources and obtains other resources in return. For example, in Napster [90], the most famous P2P application, a user offers music to the rest of the community and gets other music in return. P2P is also a way of implementing decentralized systems and applications to leverage vast amount of computing power, storage, and connectivity from personal computers distributed around the world.

The P2P approach is often used to reduce cost by eliminating the need for costly infrastructure, to improve scalability and reliability by avoiding dependency on centralized points, to enable resource aggregation by utilizing the otherwise unused resource

in the edge of the network, to increase autonomy, to provide anonymity and privacy protection, to fit for highly dynamic computing environment, and to enable ad-hoc communication and collaboration.

3.2.1 Peer-to-Peer Applications

The P2P systems can be classified into four categories: distributed computing, file sharing, collaborative systems, and P2P platforms [89].

Distributed computing

Distributed computing is very successful by using P2P approach. Most implementations have focused on compute-intensive applications. The Beowulf project from NASA [91] has shown that high performance can be obtained by using a number of standard machines. The general idea is that idle cycles from any computer connected to the Internet can be aggregated to solve difficult problems that require extreme amounts of computation. A large task is split into smaller sub-pieces that can execute in parallel over a number of independent peer nodes. Typically, distributed computing applications require a central controller with long running jobs (months or years) to distribute sub-tasks and collect results.

Examples of implementations include searching for extraterrestrial life [92], code breaking, portfolio pricing, risk hedge calculation, market and credit evaluation, and demographic analysis. Some projects have been raising intensive interest from many users within the Internet community. For example, SETI@home [92] now has a con-

solidated power of about 25 Tflop/s (Thousands of Billions of floating point operation per second), collected from more than three million registered user machines.

Content and file sharing

Content storage and exchange is one of the areas where P2P technology has been most popular. These applications focus on storing information on and retrieving information from various peers in the network. Applications like Napster [90] and Gnutella [93] allow peers to search for and download large files, primarily music files, that other peers have made available. Internet users can use them to circumvent bandwidth limitations that make large multimedia file transfers unacceptable with traditional client-server models. A number of research projects have explored the concept of P2P file systems [87, 94, 85, 95, 86, 84]. Filtering and mining applications such as OpenCOLA [96] and JXTA Search [83] are beginning to emerge. Instead of focusing on sharing information, these applications focus on collaborative filtering techniques that build searchable indices over a peer network.

Distributed storage systems based on P2P technologies has many advantages. First, these systems may provide the user with a potentially unlimited storage area by taking advantage of redundancy. For example, in Freenet [88], a given file is stored on some nodes in the P2P community, but it is made available to any of the peers. A peer requesting a given file just has to know a reference to a file, and is able to retrieve the file from the community by submitting the file reference. Second, the duplication and redundancy policies in some projects, such as OceanStore [85] and Chord [84], offer high availability virtual storage by replicating critical files in many

peer nodes. Last, anonymous storage service can be supported in peer-to-peer file systems. Freenet [88] and Publius [97] are two examples of anonymous P2P storage systems.

Collaboration

The inherently ad-hoc nature of P2P technology makes it a good fit for user-level collaborative applications. It allows real-time collaboration between users, without relying on a central server to collect and relay information. The applications range from instant messaging and chat (such as Yahoo! [14], AOL [12], and Jabber [15]), to shared applications and online games. Shared applications allow people to remotely interact while viewing and editing the same information simultaneously. P2P games are hosted on all peer computers and updates are distributed to all peers without requiring a central server. Example games include NetZ 1.0 by Quazal, Scour Exchange by CenterSpan, Descent, and Cybiko.

Platforms

P2P technology can be used as some sort of platforms and middleware solutions for users and services connected to the web or in an ad-hoc network. There are a number of candidates competing for future P2P platform. .NET [98] is the most ambitious one, going beyond P2P to encompass all service support on the client and server side. JXTA is another attempt, taking a bottom up and strong interoperability approach. Most other systems also have some level of platform support, such as Groove [99] covering enterprise domain and Magi [100], covering handheld devices domain.

3.2.2 Manage User Behavior in Peer-to-Peer Systems

Peer-to-peer networks suffer from the problem of free-riders [101]. Selfish users consume resource on the network without contributing anything in return. Ideally, the users in peer-to-peer systems will be altruistic, "from each according to his abilities, to each according to his needs". In practice, however, altruism breaks down as networks grow larger and include more diverse users. This situation can lead to a "tragedy of the commons", where the selfish behavior causes the system to collapse [102].

Several mechanisms are provided to address the free-riding problem in the peer-to-peer systems and mobile ad hoc network (can be considered as a special kind of peer-to-peer network). These efforts can be classified into five different categories: detection based to identify and isolate misbehavior users [103, 104, 105], incentive based to promote cooperation among peers [106, 107, 108, 109, 110, 111, 112], payment based to limit the access to the free resource [113, 114, 115, 113, 116, 117], fair-exchanging based to provide fairness [118, 119, 120, 121, 122], and trust and reputation based to encourage good behavior [123, 124, 125, 126] (A more detailed discussion of trust and reputation systems are presented in section 3.3).

Misbehavior detection

In [103], the author describes two techniques that improve throughput in an ad hoc network in the presence of misbehaving nodes. The watchdog method detects misbehaving nodes and the pathrater method uses this knowledge of misbehaving nodes to choose the network path that is most likely to deliver packets. However, the selfish

nodes are not punished or made widely known. They could still enjoy the service from others, which discourages the incentive of cooperation. *CONFIDANT* [104] is another approach to detect and isolate misbehaving nodes. However, fairness is not considered as a high priority. In [105], a system based on a counter, called the nuglet counter, is used to address the problem of fairness and cooperation. The counter is decreased when the node sends its own packet and increased when the node forwards a packet. The counter is required to remain positive. Although this approach successfully solves the problem, it requires specially designed hardware to prevent this counter from being modified by the end user.

Incentive based approaches

Blanc and et al. models the problem of peer-to-peer network routing as a random-matching game [107]. A social norm strategy, which is similar to Kandori's work [127], is proposed. In this strategy, a node cooperates with only honest nodes and drop the packets from guilty nodes. Under certain conditions, this strategy is a subgame-perfect equilibrium for this routing problem. A trust-worthy third-party authority is used for a simple reputation system to honestly record the node's behavior and updates its reputation. The simulation results show that the system is robust under malicious nodes and noise. Hidden-actions in ad hoc network routing are discussed in [108]. Unlike the reputation system assumed in [107], it lets the sender to provide incentives, such as payments to encourage the intermediate nodes to forward its packets. The author proves that even without global monitoring, the Nash equilibrium can still exist in which all intermediate nodes cooperate. On the other hand, moni-

toring does provide some benefit by providing a dominant strategy equilibrium. The details of the payment is not discussed in this paper. In [109], an admission system is presented to promote the incentive of cooperation among users. The admission system consists a reputation-based differentiated admission control to accept or deny requests based on the user's reputation or contributions in the past. An eigenvector based method is used to compute the service reputation and usage reputation from the service credit matrix.

Fair exchange of resources

BitTorrent [118] is an example of fair exchange peer-to-peer system. In BitTorrent, the burden of uploading a large file can be distributed to multiple downloaders, which exchange different pieces of the same file among each other. It uses a tit-for-tat algorithm to select the set of uploading peers based on their downloading speed to achieve efficiency and fairness. Thus, fairness is achieved by fair exchanging network bandwidth resource between a pair of peers. The performance of BitTorrent is analyzed in [122].

Payment based systems

Free-riding is profiting to the selfish users because of the free of the public resources. Payment-based system is another way to prevent free-riding by enforcing some kinds of payments or proofs of work to the public resources. Some micropayment protocols are proposed in [115]. A payment based e-mail system is proposed in [114] to prevent

spam. In [113], lightweigted currency is used for the P2P resource market. Processing-bound and memory-bound proof-of-work mechanisms are discussed in [116] and [117].

3.3 Peer-to-Peer Trust Management Systems

Decentralized peer-to-peer applications and wireless ad hoc networks are characterized by the absence of a single centralized authority for controlling and coordinating the behavior of the peers. Each peer makes its local autonomous decisions regarding its behavior. These systems are vulnerable to selfish behavior and malicious actions. Peers need to determine the trustworthiness of other peers in the system. This can be achieved in several ways such as relying on direct experiences or acquiring reputation information from other peers [128].

The concept of trust has been widely studied both in computer science and other fields such as sociology, history, economics, and philosophy [129]. Grandison and Sloman define trust as the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context [130]. Reputation is another important concept closely related to the trust. It is defined by Abdul-Rehman and Hailes as an expectation about an individual's behavior based on information about or observations of its past behavior [131]. The reputation information may be used to determine the trustworthiness of a given peer. An individual who is more reputed is generally considered to be more trustworthy.

Substantial research has been done on the area of trust management in P2P systems [132]. It is considered as a successful approach to maintain overall credibility

level of the system as well as to encourage honest and cooperative behavior among peers. Some of them are focused on developing formal means of expressing trust and reputation relationships [133]. Marsh [129] is among the first to introduce a computational trust model in the distributed artificial intelligence (DAI). A formal framework is introduced in [134] for the analysis and specification of trust evolution and update model, which provides a good way to represent trust evolution and update functions. Carbone and Nielsen proposed a formal model of trust by the Global Computing scenario focusing on trust formation, evolution and propagation [135].

Many different kinds of trust models have been developed for different applications. The PGP model [136], which uses the concept of "web of trust" [137], is the most widely used distributed trust model, primarily in the field of key distribution. The Poblano [138] proposal is designed to perform reputation guided search, so that peers can determine the quality of another peers' content under the JXTA framework. Albul-Rahman and Haies present a distributed trust model based on recommendation to support virtual community [131]. Karl and Zoran present a decentralized trust model in [139], which focuses on both data management and the semantic level. Winslett develops a set of solutions for automated trust building and negotiation [140]. The similar idea is used in the mobile devices for authentication and authorization [141]. The architecture provides an approach for access control and authentication in the highly sensitive interactions between strangers, such as credit card transaction or request of sensitive information. Some trust and reputation systems are developed for online service provision [142].

Based on the way adopted to establish and evaluate trust relationship between peer, the trust management systems can also be classified into three categories: credential and policy-based trust management, reputation-based trust management, and social network-based trust management [132].

3.3.1 Policy-based Trust Systems

In credential and policy-based trust management systems, peers use credential verification to establish trust relationships with other peers [143, 144, 145]. Service providers use credentials and policies to determine the trustworthiness of service requesters and to enable access control. The role of trust management is limited to verify credentials and restrict access to resources according to application-defined policies [130]. The credentials of the requesting peer can be verified either directly or through a web of trust [137]. These systems are only useful when the service providers and their services can be fully trusted. They do not provide the way for the requesting peer to establish trust in the resource-owner and may not be used for all decentralized applications.

PolicyMaker [146] is a trust management system that facilitates the development of security features including privacy and authenticity for different kinds of network applications. Using PolicyMaker a peer may grant another peer access to its service if the providing peer can determine that the requesting peer's credentials satisfy the policies needed to access its service. It provides each peer with local control to specify its policies for the authenticity of credentials presented by other peers, and support

complex trust relationships. A common language is provided by PolicyMaker to program policies, credentials, and relationships. The trust mechanism is separated from the policies in order to keep the authentication mechanism application-independent. The PolicyMaker service acts like a database query service, in which a query is a request to determine whether a public key is permitted to perform a particular action according to a given policy.

KeyNote (RFC 2704) [147] and REFEREE [148] are two other trust management systems based on the same principles as PolicyMaker. Unlike PolicyMaker which placed the task of credential verification upon the application itself, the trust engines in KeyNote and REFEREE are responsible for signature verification.

3.3.2 Recommendation-based Trust Systems

Reputation-based trust management systems provide a mechanism by which a peer requesting a resource may evaluate its trust in the reliability of the resource and the provider [132]. Peers establish trust relationships and assign trust values with each other based on direct or indirect knowledge on earlier interactions. HISTOS [149], XREP [150, 151], NICE [152], DCRC/CORC [153], Beta [154], P-Grid [155] and EigenTrust [156] are examples of such systems.

Distributed trust model based on recommendations

In [157], a distributed trust model is proposed based on a recommendation protocol. The trust relationship is defined as always between exactly two entities, is non-symmetrical (or undirectional) and is conditional transitive. It is further distin-

guished into "direct trust relationship" and "recommender trust relationship". When one peer trusts another, it constitutes a direct trust relationship. But if a peer trusts another peer to give recommendations about another peer's trustworthiness, then there is a recommender trust relationship between the two [131]. Trust information is stored locally, and there is no global centralized map of trust relationships. A simple recommendation protocol is used among entities to facilitate the propagation of trust information. An entity will send its recommended trust information to other entities by request. The recommended trust may propagation via a path to many entities. This approach is focused on decentralization, trust generalization, explicit trust, and recommendations. Decentralization allows each peer to manage its own trust. Trust generalization is concerned with identifying that there are different dimensions to trust called trust categories, and trust in a peer varies depending on these dimensions. Six different trust value (Distrust, Ignorance, Minimal, Average, Good and Complete) are defined for each trust relationship so that trust values can be compared. Finally, in a large distributed system, it is difficult to obtain knowledge about every entity in the network, let alone first hand knowledge and experience of them. Therefore, in order to cope with uncertainty arising due to interaction with unknown peers, a peer has to rely on recommendations from known peers about these unknown peers.

3.3.3 Social Networks-based Trust Systems

Social networks-based trust systems try to utilize social relationships between peers when computing trust and reputation values. In particular, they analyze a social

network which represents the relationships existing within a community and form conclusions about peers' reputations based on different aspects of the social network. Examples of such trust management systems include Regret [158] that identifies groups using the social network, and NodeRanking [159] that identifies experts using the social network.

Community-based Reputation

Yu and Singh are one of the first to explore The effect of social relationships of peers belonging to an online community is first explored by Yu and Singh [160]. A social mechanism is proposed to avoid interaction with undesirable participants. It models an electronic community as a social network. Peers can have reputations for providing good service and referrals. Each user is assigned with a personal agent that assist him to decide whether or how to respond to requests received from other peer agents in the system. The agent also helps to evaluate the services and referrals provided by other peers in order to enable the user to contact the referrals provided by the most reliable peer.

The reputation rating of a agent is based on the direct observation with this agent as well as the the rating of the agent given by the neighbors, and the ratings of those neighbors. This makes this approach a social one and enables information about reputations to propagate through the network. Each agent maintains a set of changing neighbors whom it may directly contact with. How an agent evaluates the reputations of others closely depends on the testimonies of its neighbors, which naturally leads to the idea of a referral chain. Simulation results have shown that

the reputations of selfish and undesirable agent peers decrease rapidly. The initial barrier of entry is low so that the reputation of a new peer will increase steadily by cooperating with others. A negative testimony about a malicious peer is quickly propagated to other peers, which makes the system weed out undesirable players.

REGRET

REGRET [158], as well as TrustNet [161] includes the social dimension of peers and their opinions in its reputation model [162]. Rather than relying only on the corresponding social network as in TrustNet, REGRET model adopts the stance that the overall reputation of a peer is an aggregation of different pieces of information. It takes into account three dimensions of reputation: the *individual dimension*, the *social dimension* and the *ontological dimension.* Individual dimension is used when the peer only depends on its direct interaction with other members in the society to evaluate reputation. Social dimension is defined as the information about another peer provided by other members in the society. In some situations, for example in the electronic marketplace, the points of view of all the members of a community related some specific services are supposed to be unified. The reputation is considered as a global property of an entity and common to all the members of the society, which forms the ontological dimension of the reputation. A single value of reputation is computed by combining these three dimensions.

Chapter 4

Sharing Energy Efficient Instant Messaging Channel

4.1 Overview

Mobility and continual network access are among the needed characteristics to build applications that reside upon emerging pervasive computing systems. However, the services that require the ability of "alway on, anywhere" Internet access are generally restricted by the limited battery power in the mobile devices. In this chapter, we develop a Message Notification Protocol (MNP) that enables reductions in power consumption for applications that convey presence information of mobile users, such as IM services. Nearby mobile devices form groups of peers within ad hoc networks via their WLAN network interfaces for the purpose of maintaining an IM presence through a single shared cellular channel. Peers in the same group are notified when a message or presence information is available for them in the remote server through

a single shared cellular channel maintained by a proxy. Since our MNP enables presence information and the message notification data between the WWAN and the proxy to be significantly compressed, very little data costs are required to serve as a proxy. Battery consumption will only be required continuously by the single peer that momentarily is connected to the WWAN. All cooperating peers will be able to save battery power while maintaining presence at an IM server.

Some difficulties must be solved in order for shared message notification to be successful. First, peers in the CHUM group are dynamic and unreliable, which causes difficulty for group maintenance. If the information about group membership needs to be updated frequently on the peers, there will be a high communication and power overhead. Furthermore, peers are distrusted and may leave the group at any time. This unreliable feature makes it undesirable to let a proxy for a group of peers cache messages for other peers that are in a power-saving mode. Second, security and privacy within a CHUM environment become important. A proxy should not have the opportunity to examine the content of messages intended for other peers or modify them. Some sensitive peers may even be unwilling to let others know their user names within the IM server.

Due to these considerations, we limit the responsibilities of the proxy within the MNP to broadcast message notifications for all other peers in the group. These notifications are generated by IM servers that reside within the wired Internet. The proxy does not need to maintain group information or perform other management duties. Messages are cached by IM servers and fetched by the peers directly from the IM servers via their own cellular connections after they receive message notifications

from the proxy. The proxy is not involved with message exchange, but merely provides message notification in order to reduce the need by PDAs to maintain connections with their IM servers. The proxy will not have the capability to examine or interfere with the content of the IM messages of some other peers.

A description of the responsibility of the proxy and the message notification protocol in the system is given in section 4.2. The compact data representation of the message notification is given in section 4.3. The prototype implementation is described in section 4.4. Section 4.5 provides an energy model to analyze the possible power saving on the mobile device. A discussion is given in section 4.6 and a summary is given in section 4.7.

4.2 Message Notification Protocol (MNP)

In the MNP, the proxy creates a *sharing group* and cooperates with the IM servers to provide the *message notification service* for the other peers in the same CHUM group. After a peer subscribes this service, it will receive a short notification from the current proxy when a message arrives on its IM server. This notification is constructed by the *active server*, which is the IM server of the proxy, after receiving the information from the other IM servers. It contains a *notification set*, which is the set of peers that have IM messages waiting.

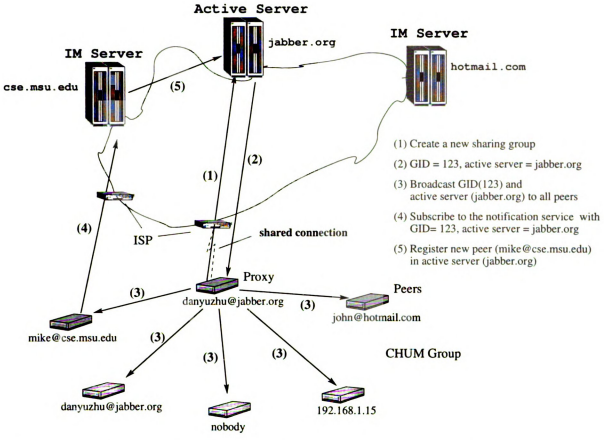


Figure 4.1: Sharing group creation and message notification service subscription.

4.2.1 Basic Idea

The current proxy creates a new sharing group by sending a creation request to its IM server. This IM server replies with a sharing group ID (GID) and its address, and serves as the current active server for this sharing group. The proxy then broadcasts the GID and the active server address to all the other peers. If a peer wants to join the sharing group and subscribe to the message notification service, it sends a subscription request to its IM server along with its peer identification (PID), the GID and the active server address. Its IM server then sends a register request to the active server along with the PID and the GID. This PID is added into the sharing

group in the active server. In the future, when there is a message for this peer, its IM server sends a notification to the active server rather than contacts the peer directly. This peer also has an "available" presence to all its friends at the IM server. After the successful subscription, the peer may safely turn off its WWAN connection and switch to power-saving mode in order to save energy. The peer periodically listens to broadcasts via the WLAN from the proxy, to determine if a message notification is directed to it. As will be discussed in section 4.3, only a few bit positions of a message need to be examined by a peer. Fig. 4.1 shows the procedure for sharing group creation and message notification service subscription. The proxy *danyuzhu@jabber.org* creates a new sharing group *123* in IM server *jabber.org*. The peer *mike@cse.msu.edu* subscribes to the notification service and joins the sharing group *123*.

When there is a message arriving on the IM server for a peer that has subscribed to the message notification service, the IM server sends a notification to the current active server along with the PID and the GID. The active server then constructs a short notification containing the notification set and sends it to the current proxy. The proxy broadcasts this notification to all other peers in the CHUM group. Each peer performs a membership query on the notification set. Based on the result of the query, it decides whether or not to start a new WWAN connection to its IM server. Fig. 4.2 illustrates the procedure of message notification. The peer *mike* has a new message at its IM server *cse.msu.edu*. A short notification is constructed by the current active server *jabber.org* and further broadcasted by the proxy *danyuzhu@jabber.org* to *mike*. A new connection is set up between *mike* and its IM server *cse.msu.edu* for the new message.

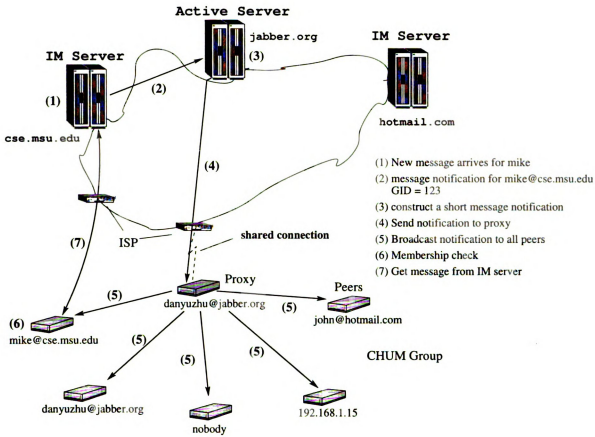


Figure 4.2: Message notification.

All peers in one CHUM group take turns to become the proxy and provide the message notification service to all other peers. The new proxy sends the migration notification to its IM server, which makes this server the new active server. The new active server then contacts the prior server and fetches the membership information for the sharing group. With the group information, the new active server may send proxy migration notifications to all the IM servers of the peers in the sharing group. The old proxy does not need to participate in the procedure of proxy migration. Fig. 4.3 shows the procedure of the proxy migration. The proxy migrates from *danyuzhu@jabber.org* to *mike@cse.msu.edu*. Likewise, the active server migrates from *jabber.org* to *cse.msu.edu*. Sharing group information is transferred from *jabber.org*

to *cse.msu.edu*. All other IM servers have been notified of the migration by the new active server *cse.msu.edu*.

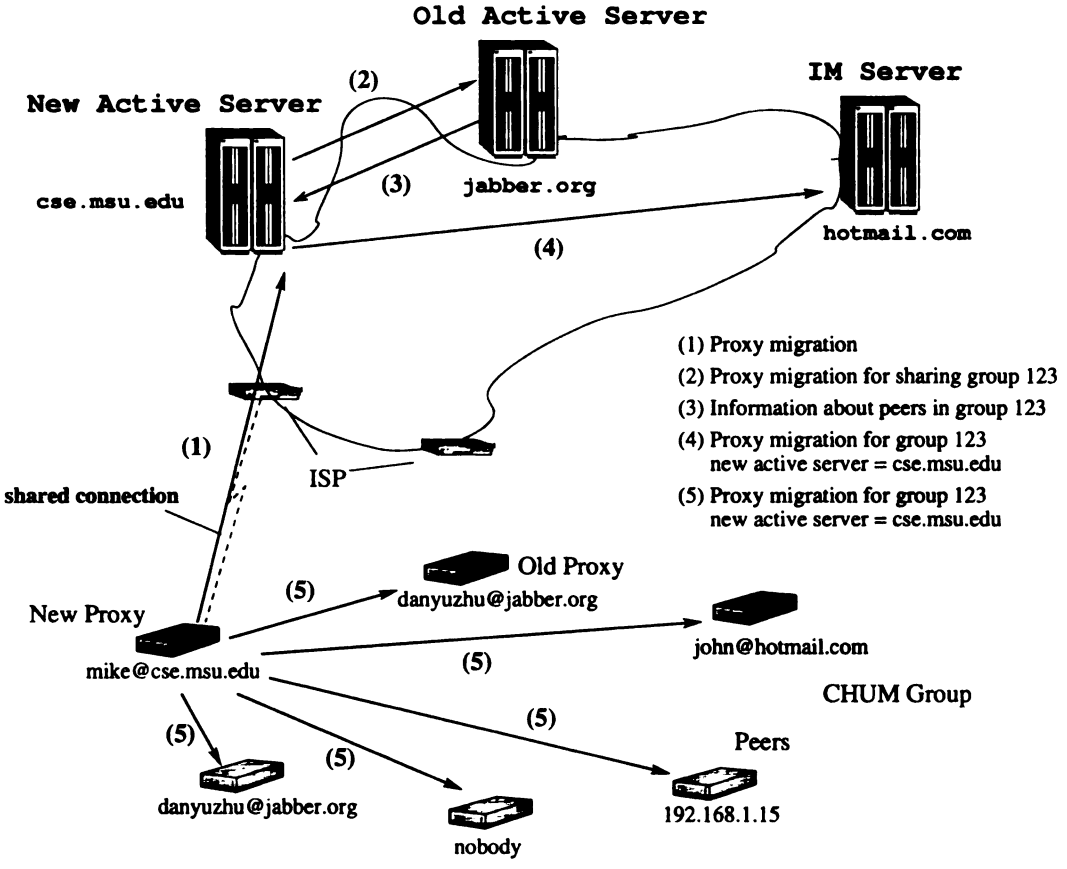


Figure 4.3: Proxy migration.

Unsubscription from the message notification service occurs when the peer decides to leave the sharing group or is involved in a long period of connection with the IM server due to chatting or file transfer, such that direct notification from the IM server is preferred. An unsubscription request is sent by the peer to its IM server. The request includes the peer's PID, GID and the active server address. Upon receiving the request, the IM server updates the state of this peer and removes it from the sharing group by sending an unregister request to the active server. Future messages for this peer from the IM server will be sent directly to the peer.

By adopting the MNP in CHUM, the proxy only receives and rebroadcasts notifications. With this simple responsibility, the proxy does not need to know exactly who are in the CHUM group at a given moment. Since knowledge of the group composition is not required by the proxy, there is no need to maintain the high overhead that is required for group maintenance in a highly dynamic and unreliable environment. Groups may be maintained by the IM servers, who operate in a more reliable wired Internet environment, and may develop trust relationships with each other. Furthermore, instant messages do not travel through the proxy, and therefore there are no privacy and security issues associated with the proxy seeing messages. In addition, steps may be taken so that a proxy does not know the names of the peers receiving message notifications. We describe how this is possible in the next section.

4.3 Using Bloom Filters to support Message Notification

In the MNP, the frequent notification between the proxy and the active server may consume bandwidth, battery power, and increase data cost. If the normal method is used for identifying a peer, such as using his/her email address or instant message login identification, then dozens of bytes in length are required for each peer notified. Furthermore, the proxy knows the names of all peers receiving notification, and has potential to misuse it. A new data structure is needed to represent the message notifications in MNP.

Bloom filters [163] provide an excellent method for message notification representation. It is a compact data structure for a probabilistic representation of a set in order to support membership queries. It needs only a small amount of space and may provide an answer to a membership query in "constant" time (time to hash). There may be a small rate of false positives to requests, which may be controlled by the parameters of the Bloom filter. A simple compression may be performed on the original Bloom filter to further reduce the size of the notification.

4.3.1 Bloom filter Background

Suppose a set $P = \{p_1, p_2, ..., p_n\}$ has n elements. A Bloom filter is a vector BF of m bits, initially all set to 0. K independent hash functions $h_1, h_2, ..., h_k$ are chosen with range $[1, m]$. For each element $p_i \in P$, the bits at positions $h_1(p_i), h_2(p_i), ..., h_k(p_i)$ in the Bloom filter BF are set to 1. When checking the membership for a unknown element u, the position of $h_1(u), h_2(u), ..., h_k(u)$ is checked. If any bit position is 0, then u is not in the set P. Otherwise, u is in the set P with small false probability. This is called "false positive." There is a tradeoff between the size of Bloom filter m, the number of hash functions k and the probability of false positive f. By [164], $f \approx \left(1 - e^{-kn/m}\right)^k$. With determined m/n, when $k = \ln 2 \times m/n$, f will give a minimum value: $f \approx (0.5)^k = (0.6185)^{m/n}$. In order to have better data compression while keeping the false positive probability low, we increase the value of m/n but decrease the value of k.

4.3.2 Bloom filters as message notification in CHUM

In CHUM, the Bloom filter BF is constructed by the notification set $P = \{p_1, p_2, ..., p_n\}$. Element p_i in the set P is the normal text identification for $peer_i$ in the group. $Peer_i$ and the IM server for $peer_i$ may agree on an alias name or other means to obscure the identity of p_i for $peer_i$. The membership query for peer q will be performed by checking the bit positions of $h_1(q), ..., h_k(q)$ in the Bloom filter BF received.

```
set P = notification set
bloom filter BF;
for (each p in set P)
 {
   string MDp = MD5(p);
   compute the h1(MDp), h2(MDp), h3(MDp), h4(MDp);
   set the coresponding bits in bloom filter BF;
 }
n = number of 1s in BF;
if (n < 32)
   BF' = compress(BF);
else
   BF' = 0x00 appends with original BF;
send BF' to current proxy;
```

Figure 4.4: Algorithm for constructing compressed Bloom filter representation .

The number of elements in set P, n, ranges between $[0, N_{max}]$, in which N_{max} is the maximum possible number of peers that may be contained in the notification set. In other words, N_{max} represents the maximum number of peers that may be notified by one notification message.

If we use small m/n and follow this optimal selection k, $k = \ln 2 \times m/n$, there may be a large message overhead, while also having a high false positive probability. For example, we may choose $m/n = 6$ and $k = 4$, then the false positive rate is $f \approx 5\%$.

Suppose $n = N_{max} = 16$, then we have $m = 6 \times n = 96$ bits $= 12$ bytes. However, since notification happens soon after the message arrives at the IM server, it is likely that there is only one element in the notification set. The use of 12 bytes to represent this notification may even be worse than using the normal peer identification information (email address or IM login name).

In order to reduce the size of the Bloom filter, we follow the idea developed in [165] to compress the Bloom filter representation before transmitting. Fig. 4.4 shows the algorithm of constructing a Bloom filter. For $n = N_{max} = 16$, we choose $m/n = 16$ and $k = 4$. Therefore $m = 16 \times n = 256$ bits $= 32$ bytes. The four hash functions are built by first calculating the MD5 signature of the PID, which yields 128 bits, and then taking four groups of 32 bits from it and mapping it into the range of 0 to 256. The false positive f will be very small:

$$f \approx \left(1 - e^{-kn/m}\right)^k = 0.0024 = 0.24\%$$

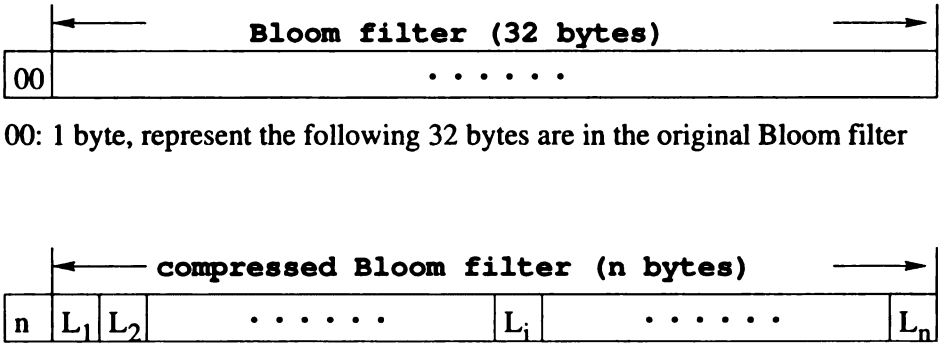


Figure 4.5: Compressed Bloom filter.

This original Bloom filter, which is 32 bytes long, is further compressed by the algorithm. Fig. 4.5 shows the compressed representation of the Bloom filter. If the number of bit positions in the original Bloom filter that have been set to 1 is more than 31, we use the original Bloom filter prepended with $0x00$. Otherwise, the compressed data compression contains a list of the bit locations that are set to 1 in the original Bloom filter. Since the original Bloom filter is 256 bits long, each location may be represented by 1 byte. The first byte of the compressed data represents the number of the locations in the following list.

The bandwidth needed by the normal text identification representation is linear to the number of peers in the notification set n_p and may grow up to hundreds of bytes. The number of bytes needed for the uncompressed Bloom filter is a constant value, 32 bytes. The compressed Bloom filter representation ranges from 5 bytes to 33 bytes, at most. For $n_p < 8$, the bytes needed in compressed approach will be at most $n_p \times 4 + 1$. In the most common case, there is only one element in the notification set. Therefore, only 5 bytes are needed for the compressed Bloom filter instead of 32 bytes for the original Bloom filter and the dozens of bytes for normal text identification representation.

Fig. 4.6 shows an example of a compressed Bloom filter. In this example, the notification set contains four peer identifications, which normally results in 70 bytes. The uncompressed Bloom filter representation needs 32 bytes. The compressed Bloom filter further reduces the length of transmitting to 17 bytes, which is less than 25% of the normal text peer identification representation. More importantly, the scheme enables IM user names to be obscured and there is no need for a proxy to know who

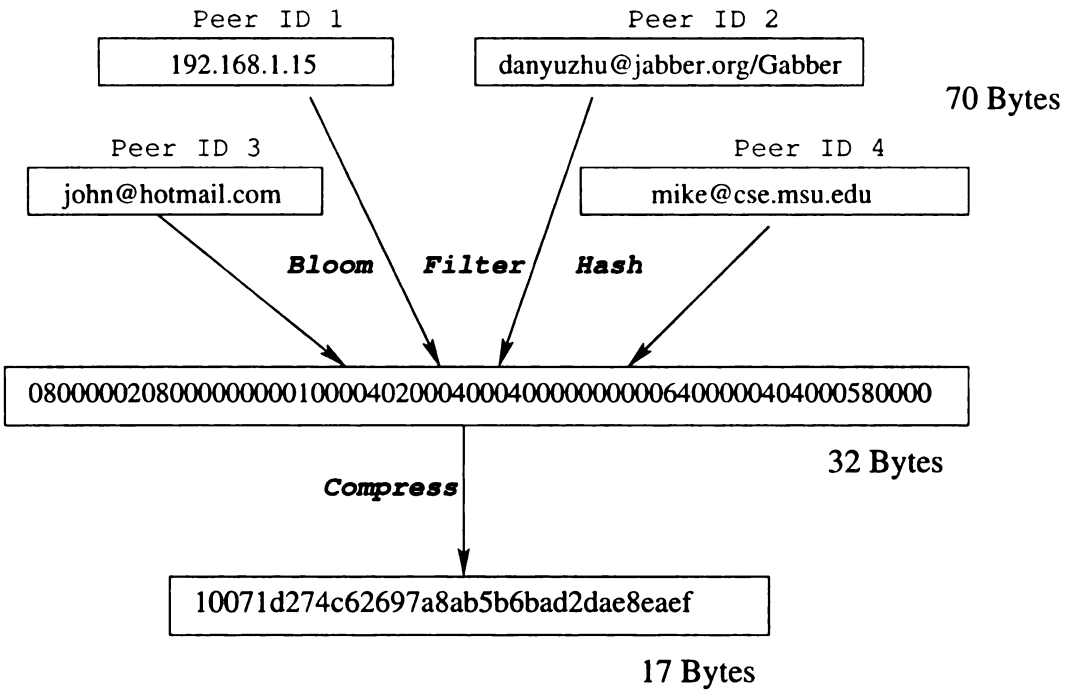


Figure 4.6: Example of the message notification represented by a Bloom filter.

have messages waiting. The proxy merely needs to broadcast the compressed Bloom filter to all peers, and each peer determines for itself if it has an IM message waiting.

4.4 Implementation

We choose the Jabber [15] IM system as the platform for the implementation of CHUM and the MNP protocol. Jabber is an open XML protocol for Instant Messaging that provides similar functionality to commercial IM systems such as AIM, ICQ, MSN, and Yahoo. It is distinguished from existing IM services by several key features:

- *XML foundation:* The design of Jabber is base on XML. Its software and protocol are communicated via XML. It allows new protocols to be transparently implemented on top of a deployed network of servers and applications.

- *distributed network:* The Jabber architecture is similar to that in email. Peers are connected and route data in a chain until it reaches the desired recipient. A client is connected to its server only, and its server is responsible for negotiating the delivery and receipt of that client's data with other servers or networks using whatever protocol available. Each server functions independently of the others, and maintains its own user list. Peers play both as the client and the server in Jabber architecture.

- *open protocol and codebase:* Jabber is an open source project. All the codes are available via Internet.

- *modular, extensible architecture:* The Jabber open-source server is designed to be modular, with specific code packages that handle functionality such as user authentication and data storage. The exchange of messages and presence information between Jabber and any given non-Jabber messaging system is made possible by means of a separate "transport" that translate Jabber XML into the foreign protocol. These transports can be easily added to core server to extend the service to new available non-Jabber messaging system.

The modular designed ***jabberd*** server makes it possible to add new functionality by supplementing the core server process ***jabberd*** with various server components. We developed a new MNP service component in the existing Jabber server version 1.4.2 on the Linux platform to enhance the standard Jabber server with the function of message notification service. No modification is needed in the standard Jabber server package, which means the MNP service component can still be easily plugged in when

the Jabber server is upgraded. The MNP service component works together with the core Jabber server to provide mobile users with the additional Jabber-related message notification service.

The MNP service component is running as a separate process, which not only makes it a more scalable solution but also isolates it from the core server, enabling a user to stop and start the MNP service without affecting the core server. A separate configuration file is provided for the MNP service. The process of the MNP service is connected to the core Jabber server over TCP port 5233.

A new Jabber client that supports the CHUM and MNP service is implemented by adding the features of CHUM and MNP into one existing Jabber client, Jarl, which is a Perl/Tk client originally developed by Ryan Eaton [15]. The enhanced Jarl client supports all the major features of CHUM and MNP, such as creating new sharing group, sharing group advertising, joining and leaving existing sharing group, proxy migrating, receiving and broadcasting messaging notifications.

There are two different types of communication in CHUM: inner-group communication and external-group communication. The XML chunk is used for external-group communication, which is to transfer a message between the Jabber server and the Jabber client. It is also used for communication between Jabber servers. UDP broadcast packets are used for inner-group communication among CHUM group members.

4.5 Performance Evaluation

We evaluate the possible power saving by using the MNP protocol in CHUM and compare it to the "always on" scheme. In the "always on" approach, the mobile device maintains an active WWAN connection all the time, and never turns off the WWAN interface. MNP and "always on" are the same with respect to mobile users maintaining their presence at the IM server. We could compare to a scheme in which the mobile user regularly turns off his/her device to save power, but the user would not maintain presence at the IM server to participate in instant messages.

4.5.1 Evaluation Settings

We suppose that there are two different wireless network interfaces in each CHUM device. The first one is a 802.11 wireless LAN (WLAN) interface, working within ad hoc mode. Otherwise, it could be a low-power, short-range wireless interface, such as Bluetooth [26] or 802.15 [166] WPAN interface. The WLAN interface will be used to form the CHUM group and perform the notification and sharing group advertisement among group members. The second interface is a wireless WAN (WWAN) interface, which generally consumes more energy than the previous WLAN interface. The use of the WWAN interface is billed based upon the number of bytes transmitted and received via this interface or the time connected to Internet. The WWAN interface could be a GPRS/GSM, CDMA2000 or other 3G wireless device and be used to access data and service in the Internet.

The WWAN interface may operate in two power modes: SEND/RECV or IDLE, which is defined for actively transmitting data or idle. The average power consumption for these two modes are PW_{active} and PW_{idle} respectively. Four different modes are defined for the 802.11 WLAN interface: DOZE, IDLE, SEND and RECEIVE, representing low-power, ready to receive, active sending and receiving, respectively. To simplify the analysis, we assume that the message/event notification application is the only one using the wireless interfaces and the responsibility of the PROXY is fairly distributed. All WLAN packets are transmitted via broadcast.

We make the following assumptions for the performance evaluation:

- **Group Setting**: The average group size is N_p. A peer may stay in the CHUM group for T_g seconds. We also assume that the responsibility of the PROXY is fairly distributed among all peers.

- **Group Management**: Since there is no need for frequent updating of the group information, we ignore the energy consumed related to group management[1]. The only WLAN traffic we consider in the evaluation is the broadcasted message notification.

- **Traffic Setting**: For each peer, there are an average of N_m messages received from the IM server when it stays in a CHUM group. Each notification is L_n bytes and each message is L_m bytes. L_c bytes are needed for each IM connection.

[1]The evaluation result shows that the total energy consumed during the short period for sending and receiving packets in WLAN interface is very small compared to that in the long period with the IDLE and DOZE states. This assumption is reasonable.

- **Environment**: To simplify the analysis, we assume that the IM application is the only application using the wireless interfaces. All WLAN packets are transmitted via broadcast.

4.5.2 Energy Analysis Model

The energy consumed for the message/event notification application is composed of two parts: the WWAN related, EW, and the WLAN related, EL. EW can be computed as the energy consumed in SEND/RECV mode plus that in IDLE mode:

$$EW = \frac{L_{wwan}}{B_{wwan}} \times PW_{active} + T_g \times PW_{idle}$$

The time interval that the interface operates in the SEND/RECV mode could be computed as the total length of packets transmitted (L_{wwan}) divided by the bandwidth of WWAN interface (B_{wwan}). The time interval in the IDLE mode is the length a peer stays in the group, T_g, minus the time interval in SEND/RECV mode, which approximates to T_g since the total traffic for the IM application is very small.

EL is composed of four parts: the energy for sending packets EL_{send}, the energy for receiving packets EL_{recv}, the energy for staying idle EL_{idle} and the energy for sleeping EL_{doze}. We adopt the linear energy model developed in [167], which is obtained by real experiments on Lucent WaveLAN 11Mb wireless cards, to compute EL_{send} and EL_{recv}. Sending and receiving a packet has an energy consumption $E_{base} + E_{byte} \times size$, where E_{base} is the energy consumption independent of packet length, E_{byte} is the energy consumption per byte, and $size$ is the packet length. The parameters presented in [167] are used : $EL_{send} = 272 + 2.1 \times size$ and $EL_{recv} = 50 + 0.26 \times size$. In the

IEEE 802.11 power saving mode [23], time is divided into beacon intervals. The mobile device wakes up periodically in a short interval called the *ATIM window* at the beginning of the beacon interval. It will stay awake during each ATIM window and it should also stay awake during the rest of beacon interval when it needs to transmit or receive a packet. Therefore, EL_{idle} and EL_{doze} will be related to the size of the ATIM window, beacon interval and the number of packets transmitted.

In the "always on" approach, there is no WLAN related energy consumption. The total packets transmitted through the WWAN interface will be one connection packet plus the message packets.

$$E_{always} = \frac{L_c + N_m \times L_m}{B_{wwan}} \times PW_{active} + T_g \times PW_{idle}$$

The total power consumption for the whole group which has N_p peers will be:

$$ET_{always} = N_p \times E_{always}$$

In the MNP protocol, the proxy maintains a continuous WWAN connection and other peers connect to Internet only when there is a message awaiting in their IM servers. Since the proxy needs to receive additional $N_m \times (N_p - 1)$ message notifications for other peers in the CHUM group, the WWAN related energy consumption for the PROXY, EW_{proxy}, will be:

$$EW_{proxy} = \frac{L_c + N_m \cdot L_m}{B_{wwan}} PW_{active} + \frac{N_m \cdot (N_p - 1) \cdot L_n}{B_{wwan}} PW_{active} + T_g \cdot PW_{idle}$$

The WWAN related energy consumption for other peers will be:

$$EW_{peer} = \frac{N_m \times (L_c + L_m)}{B_{wwan}} \times PW_{active}$$

The message notifications are transmitted by broadcasting via the WLAN interface. Therefore, all the peers should stay awake for each message notification: $EL_{idle} = \{T_{ATIM}/T_{beacon} \times T_g + (T_{beacon} - T_{ATIM}) \times N_m\} \times PL_{idle}$, where T_{ATIM} is the time interval for the ATIM window, T_{beacon} is the time interval for the beacon interval. EL_{doze} can be computed as: $EL_{doze} = \{(T_{beacon} - T_{ATIM})/T_{beacon} \times T_g - (T_{beacon} - T_{ATIM}) \times N_m\} \times PL_{doze}$

The energy consumption in the WLAN interface for the MNP protocol, $EWLAN_{mnp}$, can be expressed as:

$$EL_{mnp} = \{ES_{base} + N_m L_n ES_{byte}\} + \{ER_{base} + N_m L_n ER_{byte}\} + EL_{idle} + EL_{doze}$$

where ES_{base} and ER_{base} are independent amounts of energy consumption for broadcast send and receive, and ES_{byte} and ER_{byte} are energy consumption per byte for broadcast send and receive.

Since the responsibility of the PROXY is fairly distributed among all peers, the energy consumed in a individual CHUM peer will be:

$$E_{mnp} = EW_{proxy} \times \frac{1}{N_p} + EW_{peer} \times \frac{N_p - 1}{N_p} + EL_{mnp}$$

The total energy consumption for the whole group which has N_p peers will be:

$$ET_{mnp} = EW_{proxy} + EW_{peer} \times (N_p - 1) + EL_{mnp} \times N_p$$

4.5.3 Summary of parameter settings

Table 4.1 summaries the power consumption parameters used in our evaluation. The first two entries are parameters used for the WWAN interface[2]. The other entries are

[2]The values are product specifications for the Merlin C201 CDMA2000 1x wireless modem [168].

Table 4.1: Power consumption parameters used in the evaluation

Parameter	Description	Value
PW_{idle}	WWAN idle	300mW
PW_{active}	WWAN send/receive	4250mW
EL_{send}	802.11 broadcast send	$272 + 2.1 \times size\ \mu W{\cdot}sec$
EL_{recv}	802.11 broadcast receive	$50 + 0.26 \times size\ \mu W{\cdot}sec$
PL_{idle}	802.11 idle	$741mW$
PL_{doze}	802.11 doze	$48mW$

Table 4.2: Other parameter settings for the evaluation

Parameter	Description	Default Value
N_p	CHUM group size	10
N_m	Number of IM messages exchanged per peer	10
L_m	Length of an IM message	50bytes
L_c	Length of an IM connection traffic	300bytes
L_n	Length of a notification message	10bytes
B_{wwan}	WWAN data rate	144kbps
T_g	Time length peer stay in group	3600sec
T_{ATIM}	ATIM window size	8ms
T_{beacon}	Beacon interval	800ms

settings for the 802.11 WLAN interface card[3]. Table 4.2 summaries other parameters used in our evaluation model.

4.5.4 Evaluation Result

Based on the energy model developed in the previous section, we obtain several evaluation results. Table 4.3 presents the evaluation result of a peer that operates in the "always on" scheme and participates in the MNP protocol. For the "always

[3]We use the parameters presented in [167].

on" approach, the peer consumes 1081.9J energy for turning on its WWAN interface for an hour. However, when MNP is adopted, the energy consumption reduces to 319.31J. Therefore, the approach that uses the MNP protocol saves about 70% of the energy consumed in the mobile device comparing to the "always on" scheme.

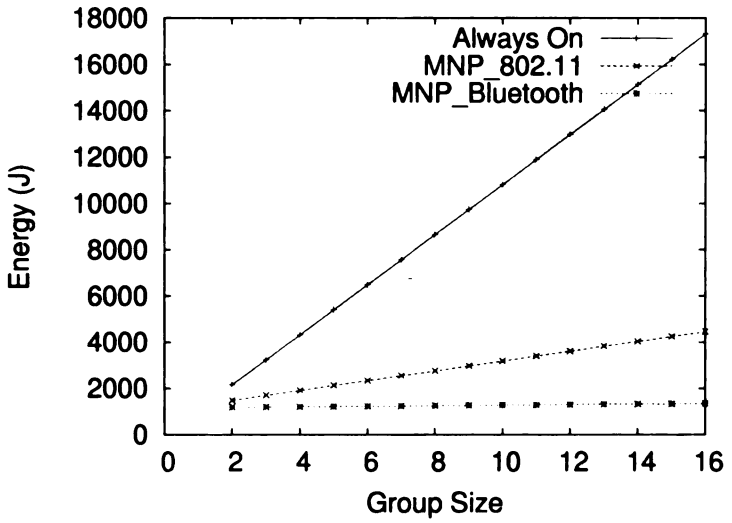


Figure 4.7: Total energy consumed by the whole group

Fig. 4.7 shows the total energy consumed by the whole group in both the "always on" and the MNP approaches. This figure reveals the relationship between the group size and the total energy consumption. In both approaches, the total energy consumption increases linearly. However, the energy consumption in the MNP approach increases much slower than the "always on" scheme. This result could be explained as follows. When the group size increases in the MNP approach, the energy consumption increases only on the WLAN interface to broadcast the increased message notifications, which is just a small fraction of the total energy consumption. On the other hand, in the "always on" scheme, every peer should always turn on its high power

consuming WWAN interface. Fig. 4.8 presents the percentage of energy saved by the MNP approach when comparing to the "always on" approach. The result shows that even with a very small group size, for example $N_p = 3$, the MNP protocol should be able to save more than 47% of the energy consumed in comparison to the "always on" scheme. The MNP protocol may save up to 75% of the energy consumed by the whole group as the group size reaches 20. The energy consumed in an individual mobile device is presented in fig. 4.9. The mobile device gains more benefit from the MNP protocol as the group size increases.

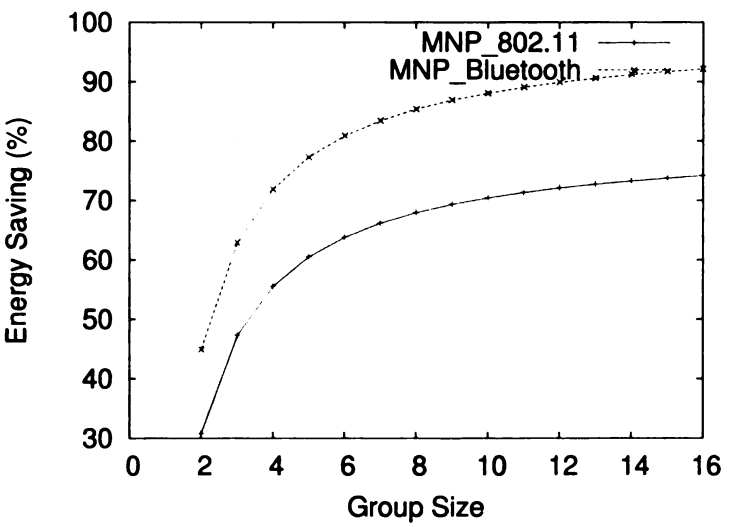


Figure 4.8: Total energy saving in the whole group

Fig. 4.10 and fig. 4.11 reveals the relationship between the duration of the CHUM group and the possible energy saving. Similar to the results for the group size, the total energy consumed increases linearly both in the "always on" and the MNP approaches. However, the total energy consumed in the MNP approach increases much more slowly than the "always on" approach. In the MNP approach, the increased

Table 4.3: Energy consumed in a peer

Parameter	Description	Energy (J)
E_{always}	"Always on" energy consumption	1081.9
E_{mnp}	MNP energy consumption	319.31
EW_{mnp}	MNP WWAN energy consumption	115.84
EL_{mnp}	MNP WLAN energy consumption	203.47
EL_{idle}	MNP WLAN idle consumption	32.54
EL_{doze}	MNP WLAN doze consumption	170.69
EL_{send}	MNP WLAN send consumption	0.210
EL_{recv}	MNP WLAN receive consumption	0.026

energy consumption in the proxy is shared by all the others in the same group. In the "always on" approach, the energy consumed in each peer is increased. Fig. 4.12 shows the energy consumed in an individual peer when it stays in the group for a longer time.

The evaluation results presented in this section are based on the assumption that all WLAN interfaces are 802.11 products. When the low-power interface, such as Bluetooth or 802.15 WPAN, is adopted, the total energy saving could be even more. For example, in CSR BC212015 BlueCore module [169], the ACL 115.2kbps MASTER mode consumes 27mW and the ACL SNIFF 1.28s SLAVE mode consumes 0.9mW. In Bluetooth, We may suppose that the PROXY in the CHUM group always operates in the MASTER mode while the other peers operate in the SNIFF SLAVE mode. Fig. 4.8 and fig. 4.11 shows the possible power savings by Bluetooth technologies. With these low-power WLAN interfaces, the total possible power savings could be more than 90%. A disadvantage of choosing these low-power WLAN interfaces is the

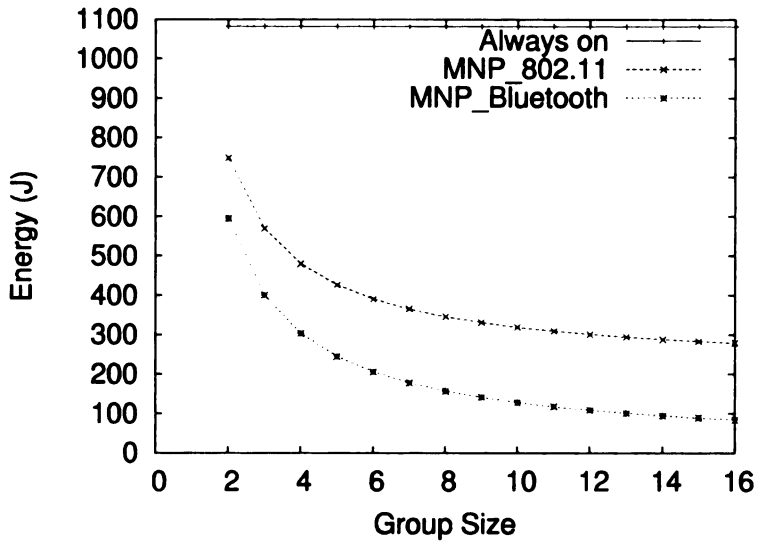


Figure 4.9: Energy consumed by a mobile device

relative short radio range. Therefore, the CHUM group may not be able to include a large number of members.

4.6 Discussion

We discuss in this section some possible enhancements and issues to MNP and the possible cost and power saving benefits by adopting the MNP and the compressed Bloom filter representation.

Orphan service subscription One possible scenario is that the peers may leave the CHUM group without unsubscribing from the message notification service. It may generate unnecessary traffic. Although it brings no benefit for the peer to do it deliberately, it may happen due to the unreliable characteristics of CHUM network. This problem may be solved by requiring the IM server to monitor the status of the

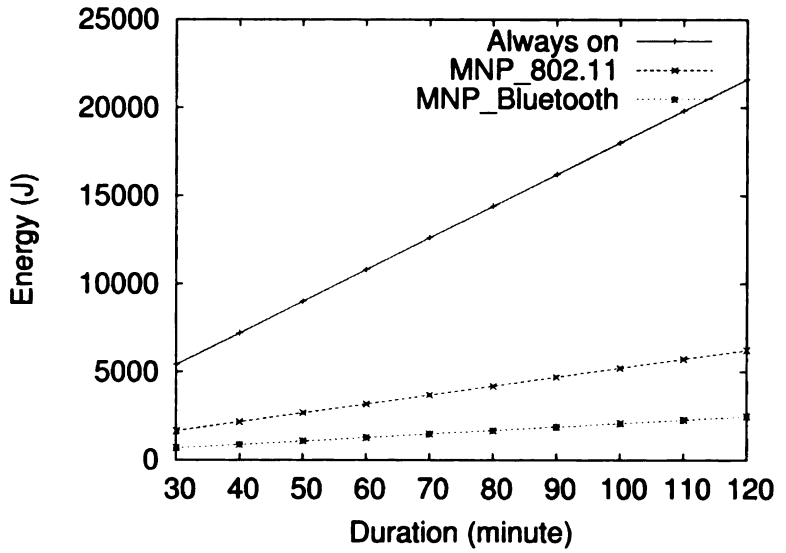


Figure 4.10: Total energy consumed by the whole group

peer. Generally, shortly after the active server sends the message notification, the peer contacts the IM server for the actual message. However, if several notifications are sent without any response, the IM server may decide that this peer is offline, unsubscribes the message notification service for this peer, and removes it from the sharing group.

Precomputing the Bloom filter Since the PID and the Bloom filter parameters are fixed during the session, we may further reduce the time and power consumption of membership query by precomputing the four hash functions on the PID and constructing a Bloom filter in which the notification set only contains the peer itself. The membership query is simply done by performing a bit AND operation with this preconstructed Bloom filter and the received Bloom filter.

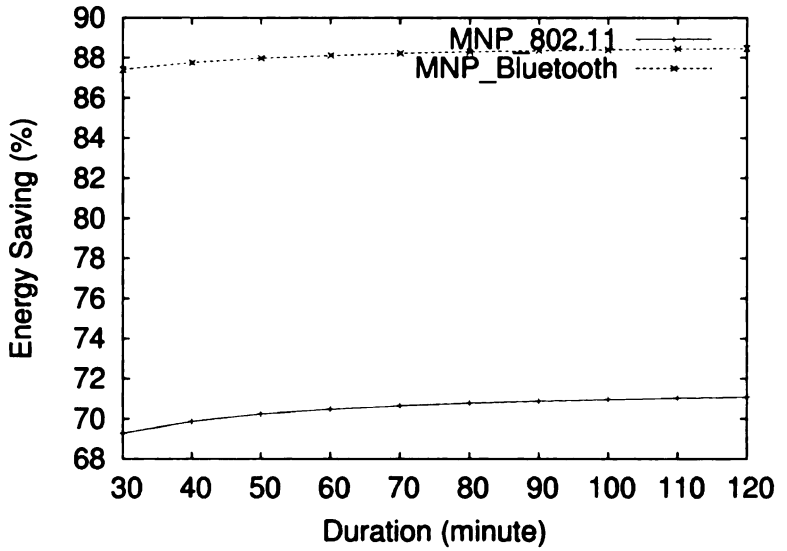


Figure 4.11: Total energy saving in the whole group

Piggy-back notifications with normal messages The overhead of the protocol may be reduced by asking the active server to postpone the notification for a short time. Within this short time period, if there is a normal packet, such as a presence information or "keep-alive" control packet, transferred from the active server to the proxy, then the compressed Bloom filter notification may be piggy-backed along with this normal packet. This piggy-back method may incur a small additional delay, but reduces the overhead for the notification exchange.

Scalability The MNP protocol works well when the size of the CHUM network is not very large. Although a larger group size increases the possible energy saving, in the real situation, it could become difficult if not impossible to find dozens of nearby peers to cooperate in CHUM. The distributed system requires every peer to record the trust information for each other peer in the group. Since the mobile devices have only limited storage capability, a group with several members would be optimal. Also the

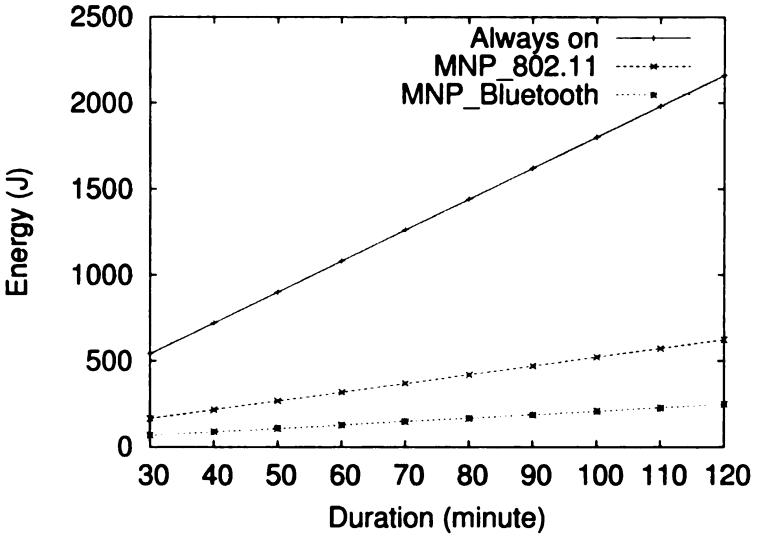


Figure 4.12: Total energy consumed by a mobile device

simulation result in [170] shows that the system becomes less stable when the group size become larger.

Multihop Scenario We assume that all the peers are within one hop WLAN radio range of the proxy. In order to join the group, the new comers always contact the proxy first, keeping the group within one hop. However, when the peers move and the proxy migrates, peers may go out of the WLAN radio range of the proxy. In this situation, some peers may lose contact with the proxy and begin the proxy competition protocol, making one CHUM group naturally partition into two one-hop groups with two proxies available. Although we could extend MNP for a multihop scenario, we keep this assumption due to several considerations. First, the power saving mode in IEEE 802.11 requires the network to be fully connected [23]. Second, a multihop network incurs additional energy consumption and reduces the benefit of

CHUM sharing. It also requires peers to cooperate to forward packets for others, which is more difficult in a mobile environment.

Security and privacy The introduction of Bloom filters to represent CHUM notifications provides a way for the peer to remain anonymous within a group of strangers and disallows the proxy to examine the content of notifications. However, it does not prevent some malicious attacks. For example, the content of the notifications may be modified or a proxy may repeatedly broadcast stale notifications. Public-keys and digital certifications may be used to maintain a secure channel and determine whether attacks occur.

4.7 Summary

Continual network access for mobile system in future pervasive computing systems may be limited by battery power. CHUM is envisioned as a system in which peers cooperate to reduce battery consumption and telecommunication charges while they maintain continuous presence at an instant messaging server.

A new message notification protocol (MNP) is developed in CHUM that enables a continuous IM presence in mobile devices with limited power consumption. The sharing of a single message notification channel enables the mobile devices to turn off their WWAN interfaces for most of the time to save power. A device only needs to contact the IM server when there is a message waiting in its IM server. MNP does not require precise group information to be maintained by a member in the ad hoc group, which

greatly reduces the overhead of group maintenance in the highly dynamic environment. A compressed Bloom filter representation is used for the message notification that is exchanged between the IM Active Server and the proxy to further reduce the protocol overhead and provide additional privacy and security protection.

In this chapter, we focus on a solution for CHUMs in an IM service. The ideas will be extended to a general event/message notification service that need continuous service connections.

Chapter 5

Power Efficient General Message Notification Service

5.1 Introduction

In this chapter, we introduce a new framework to provide an ubiquitous *CHUM Notification Service* for "always on, anywhere, mostly idle" applications in order to reduce power consumption and telecommunication cost, while maintaining a continuous network access. An example is shown in Figure 5.1. The service may be designed as a network of *CHUM agents* that provide a notification to mobile users when a pre-subscribed message/event arrives on the user's intended information server. Nearby peers form a CHUM group temporarily to share a single notification channel to the agents. The intended receiver then may choose to contact the information server to retrieve the message. A single CHUM notification channel is shared in each CHUM group, which is illustrated as four individual peers in the figure.

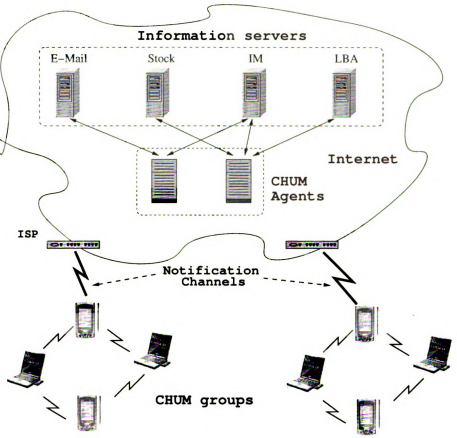


Figure 5.1: The framework of CHUM notification

Some difficulties must be solved in order for a shared message/event notification channel to be successful.

- **Group management:** Peers in the CHUM group are highly dynamic and unreliable, which causes difficulties for group maintenance. Frequently updated membership information may result in a high communication and power overhead. A simple group management protocol is needed.

- **Privacy:** A proxy should not have the opportunity to examine the content of messages intended for other peers or modify them. Some sensitive peers want to remain anonymous within the group of unknown peers.

In the following sections, we address the above challenges and describes the network functionality and its group management protocol. Section 5.2 presents the formation and functionality of CHUM networks. Section 5.3 describes the group management protocol. Discussions and conclusions are in sections 5.4 and 5.5.

5.2 Overview

5.2.1 Basic scheme

The CHUM notification service is supported by the cooperation of *information servers*, *CHUM agents* (or agents for short) and *CHUM peers* (or peers for short). The information servers are producers of message/event notifications. Peers are mobile users that form CHUM groups to share the notification channel. They are consumers of the message/event notifications. Peers may subscribe to the message/event notifications in which they are interested at the related information servers. Based on its subscription, the information server tries to send a *message notification* to the peer to inform it of the newly arrived message/event. The peer fetches the message/event from the information server after it receives the notification. Although the information server may be able to cache the messages/events for the peers when they are temporarily unavailable, a continuous network connection is needed for time-sensitive messages/events, e.g., stock quotes or auction prices. Agents are Internet servers that maintain continuous network connections and receive message/event notifications from the information servers on behalf of the peers. The agent may be a

public server providing service for several peers, or merely the home computer of a peer. We assume that a peer trusts its associated agent. The agent has a long-term identity, such as its IP address and public key. This identity is used in communication with other agents.

A CHUM *sharing group* is created at a CHUM agent. The *proxy*, which maintains the continuous Internet connection, cooperates with its associated agent, the *active agent* to provide the CHUM notification service for the other peers in the same group. After joining a sharing group, a peer receives a short *CHUM notification* from the proxy when a message/event arrives on its information server. This notification is constructed by its associated CHUM agent when it receives the message/event notification from the information server and is sent to the active agent. The active agent then forwards the CHUM notification or a new one constructed by combining several buffered CHUM notifications to the proxy. This CHUM notification contains a *notification set*, which is a set of subscribed messages/events notifications sent from participating information servers.

Note that the responsibilities of the proxy is limited to receive and rebroadcast CHUM notifications. It does not need to maintain group information, which may lead to high overhead in a highly dynamic and unreliable environment. Groups may be maintained by the CHUM agents, which operate reliably in a wired Internet environment, and may develop trust relationships with each other. Furthermore, messages/events are cached by information servers and fetched by the peers directly via their own 3G connections after receiving CHUM notifications. The proxy is not involved with message exchange, but merely provides CHUM notification in order to reduce the need

by PDAs to maintain continuous Internet connections. Therefore the proxy does not have the opportunity to examine or interfere with the content of the messages/events of other peers. In addition, steps may be taken to prevent the proxy or active agent from peeking into the contents of the CHUM notifications. We describe how this is be done in the next section.

5.2.2 Bloom filter representation of notifications

The subscribed message/event notifications may be represented as a triple-element string: (*pid@infoserver*, *class*, *condition*), which represents the peer's identification at the related information server, the type of the message/event and the condition that causes the message notification. The following is a notification example: $(mike@finance.yahoo, finance/exchange/stock, exchange = NYSE \wedge symbol = INTC \wedge change < 0)$. A peer *mike* subscribes for stock quote notification at the information server $finance.yahoo$ when the price decreases for the stock INTEL on stock exchange NYSE.

As described in [171], the Bloom filter [163] provides an excellent method to construct the CHUM notification. It is a compact data structure for a probabilistic representation of a set in order to support membership queries. It needs only a small amount of space and may provide an answer to a membership query in "constant" time (time to hash). There may be a small rate of false positives to requests, which may be controlled by the parameters.

In CHUM, the Bloom filter BF of m bits is constructed by the notification set $T = \{t_1, t_2, ..., t_n\}$. Element t_i represents a message notification. A peer p may determine if it has a message notification u by checking the bit positions of $h_1(u), ..., h_k(u)$ in the Bloom filter BF received, where $h_1, ..., h_k$ are k hash functions. If one of these B_k positions is 0, u is not contained in the notification set T. Otherwise, u contains in the set T with a small false positive f. The number of elements in set T, n, ranges between $[0, N_{max}]$, in which N_{max} is the maximum possible number of message notifications that may be contained in one notification set. In other words, N_{max} represents the maximum number of message notifications contained in one Bloom filter.

In order to reduce the size of the Bloom filter, we further compress the Bloom filter representation before transmitting. For $N_{max} = 16$, we choose $m = 256$ bits and $k = 4$. The four hash functions are constructed by first calculating the MD5 signature of the message notification, which yields 128 bits, and then taking four groups of 32 bits from it and mapping it into the range of 0 to 256. The maximum false positive f will be very small: $f \approx \left(1 - e^{-kN_{max}/m}\right)^k = 0.24\%$. The same compression algorithm as in [171] is adopted to further reduce the size of original 32 bytes Bloom filter.

A subscription list is maintained in the mobile device for the dozens or even hundreds of message notification subscriptions. For each notification, the peer needs to perform four hash functions for every entry in the list, which could consume significant energy. One possible improvement is to pre-compute the Bloom filter representation for each subscription and reduce the computation of four hash functions to bit comparisons.

The triple-element subscription implies a four-level subscription tree structure: *root*, *pid@infoserver*, *class* and *condition*. When the subscription is clustered in a few information servers, the tree structure will greatly reduce the number of comparisons required for subscriptions in the peer. Figure 5.2 shows an example of a subscription tree. Message notifications are subscribed in three different information servers, which provide message notifications for stock quotes, location-based advertising, and proxy migration, respectively.

Two more hash functions are added to the Bloom filter hash operation. They are built by first calculating the MD5 signature of the *pid@infoserver* and *class* in the notification, and then mapping it into the range of 0 to 256. In the subscription lookup, the peer will first perform the bit comparison on the pre-computed Bloom filter representation on *pid@infoserver*. If none of them matches, the subscription lookup fails. Otherwise, the peer will go further to compare the *class* part for all that matched. Finally, comparisons on the whole subscription will be performed. By this mechanism, many unnecessary comparisons will be eliminated.

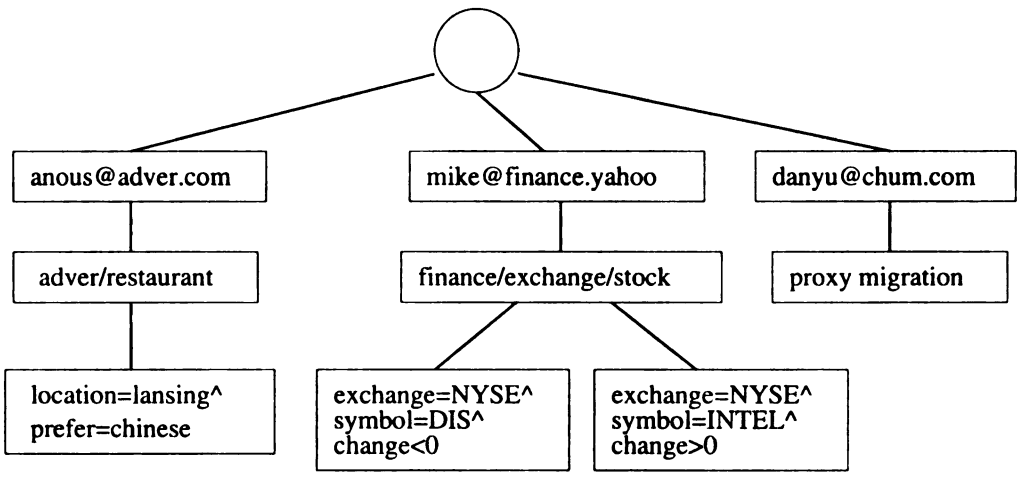


Figure 5.2: An example of a subscription list

The bandwidth needed by the normal text representation is linear to the number of subscriptions in the notification set, n, and may grow up to hundreds of bytes. The number of bytes needed for the uncompressed Bloom filter is 32 bytes. The compressed Bloom filter representation ranges from 7 bytes to 33 bytes, at most. In the most common case, there is only one element in the notification set. Therefore, only 7 bytes are needed for the compressed Bloom filter instead of 32 bytes for the original Bloom filter and the dozens of bytes for normal text representation. When the cost of WWAN communications depends on the number of bytes transmitted, our approach consumes very little cost to support notifications.

The CHUM notification provides a simple mechanism for privacy protection. It is very difficult to obtain the original text representation from the bit set in the Bloom filter vector. The active agent and the proxy cannot read the contents of the message notification.

5.3 Group Management

5.3.1 Group formation

A simple group management protocol is developed in this chapter. Within it, a mobile user creates a new sharing group by sending a *GROUP CREATION* request to its agent and announces itself as the current proxy. The agent replies with a group identification (GID), and serves as the current active agent for this sharing group after

successfully creating a group. The GID and address of active agent are advertised to all nearby peers. Figure 5.3 shows details of sharing group formation.

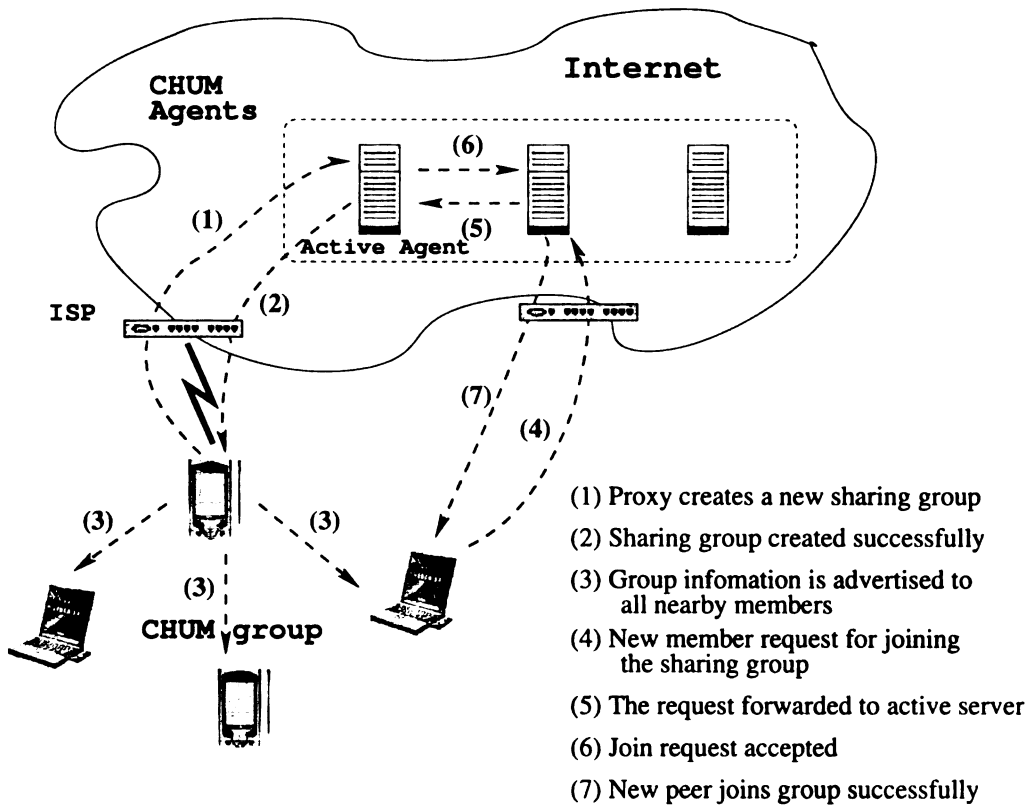


Figure 5.3: Group formation

When a peer attempts to join a sharing group, it first obtains the GID and address of the active agent either from the proxy's periodic advertisement or the group query message *GROUP QUERY*. A *GROUP JOIN* request is sent to its agent along with its peer identification (PID), the GID and the active agent address. Its agent then sends a *PEER REGISTER* request to the active agent along with the GID and a new temporary identification (TID) created for this peer in this group session[1]. The TID

[1]The active agent may not be trusted by the peer. The temporary TID instead of the real PID is used to protect the user's privacy.

is added into the sharing group in the active agent. Later, the peer may turn off its WWAN connection and switch to the power-saving mode in order to save energy.

A *GROUP LEAVE* request along with the PID and GID is sent by the peer to its agent when the peer decides to leave the sharing group. Upon receiving this request, the agent removes it from the sharing group by sending a *PEER UNREGISTER* request to the active agent. Future message notifications for this peer from its agent will be sent directly to the peer if possible.

5.3.2 Message/Event notification

The information server sends a notification to a peer's associated agent for its subscribed message/event. The agent constructs a short CHUM notification based on the PID and the content of the notification and forwards it to the active agent, which in turn routes it to the proxy of the sharing group. In order to improve performance, the active agent may buffer the CHUM notification for a short period and combine several CHUM notifications together to form a new one. The peer periodically listens to broadcasts from the proxy via the WLAN for the CHUM notification. A subscription comparison on the notification set will be performed on the received CHUM notification. Based on the result of the comparison, the peer decides whether to start a new WWAN connection to fetch the subscribed message/event. Figure 5.5 illustrates the procedure of message notification in the presence of CHUM sharing.

5.3.3 Proxy scheduling

The benefit of CHUM comes from sharing the notification channel among all participating peers in the same group to save energy and cost. Since the precise group information is not maintained in the proxy, the proxy does not perform the scheduling task. It is more reasonable for the more reliable active agent, which resides in the Internet and maintains the CHUM group information, to select the next proxy. A special subscription is reserved in each peer for proxy scheduling, e.g., the rightmost leaf in the subscription tree in figure 5.2. The length of proxy rotation cycle T_r is pre-determined at the creation of the sharing group.

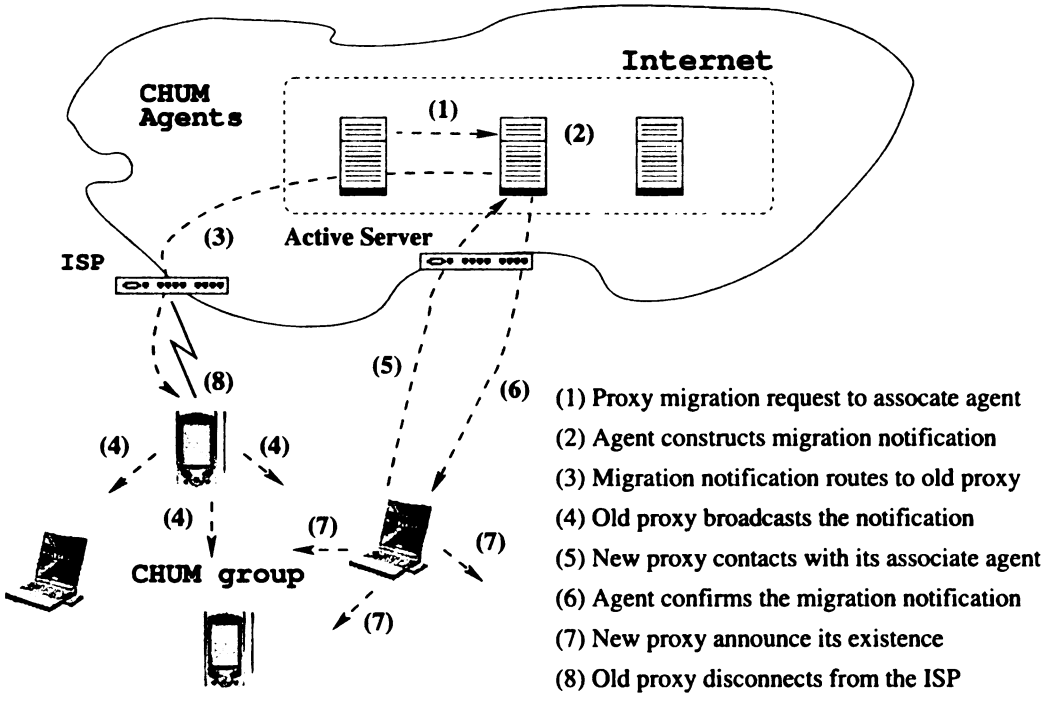


Figure 5.4: Proxy scheduling

Figure 5.4 shows the procedure for proxy migration. First, a new proxy is selected by the active agent, which sends a *MIGRATION REQUEST* to its associated agent.

The agent then constructs a migration notification based on the PID of the selected peer and sends it to the sharing group through the normal message notification procedure. When the selected new proxy receives the migration notification, it initiates an Internet connection to contact its own agent. After receiving the confirmation of proxy migration, the new proxy announces its existence via broadcasting the *NEW PROXY AVAILABLE* message. Meanwhile, its associated agent fetches the group information from the old active agent and acts as the new one. The old active agent notifies the old proxy about proxy migration, which will close its Internet connection.

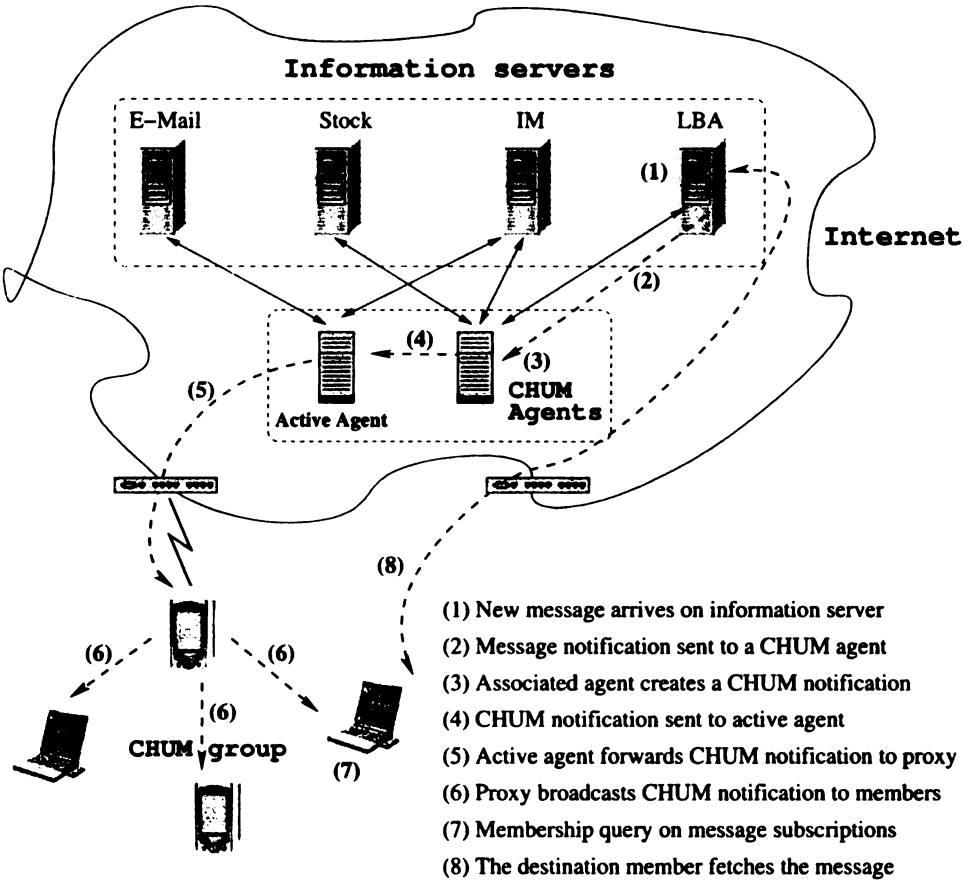


Figure 5.5: The procedure of message notification

The active server keeps a circular scheduling list of the group members. A round robin proxy scheduling algorithm is used in this chapter. When a new peer joins the group, the active server places it at the position next to the current proxy. The newly joined peer will serve as the proxy as soon as possible to reduce the effect of a "frequently join and leave" attack. This circular scheduling list transfers to the next active server when proxy migration occurs.

5.3.4 Failure recovery

Three different kinds of failure recovery are discussed in this section: the failure of a non-proxy peer, the current proxy and the selected new proxy.

A non-proxy peer may leave the group without notice while moving outside the signal range of the proxy. This failure only has a minor effect since the precise group information is not needed for the group management. Message notifications continue to be routed to the group for the leaving peer. Although some unnecessary bandwidth and energy is consumed, the loss is not significant since a compressed CHUM notification takes only a few bytes. The failure of the non-proxy peer will either be detected by the agent when it fails to take over the responsibility as a proxy or reports the failure of a proxy.

The failure of the proxy is detected by the active agent when it fails to deliver a CHUM notification. The active agent removes it from the sharing group and performs the proxy migration. The proxy also broadcasts a *HELLO* message periodically. It is easy for a non-proxy peer to detect the failure of the proxy by establishing a timeout

for the *HELLO* message. When detected, the peer initiates an Internet connection and reports the failure to the active agent through its own CHUM agent. The active agent performs the proxy migration operation after confirming the failure. If the proxy failure report is false, it may be because the reporting peer left the signal range of the current proxy. The reporting peer is then removed from current CHUM group. A new CHUM group may be created by this peer.

To detect the failure of the selected new proxy, a timeout is established on the active agent. If no response is received from the selected new proxy after the timeout, the active agent will remove it from the group. Another one will be selected as the new proxy.

5.4 Discussion

This section provides a brief discussion of possible CHUM improvements.

Intelligent adaptive message filtering: An urgency value may be attached to the notification subscriptions for the system to adapt intelligently for the current operating environment (such as remaining battery energy). We may also establish user profiles to describe the user preferences in different operating environments. The peer would be able to decide whether to "*ignore*" the advertisements, "*cache*" non-urgent messages or "*fetch*" urgent ones based on the urgency value and profile settings.

Misbehavior detection: A proxy may agree to rebroadcast CHUM notifications, and then fail to do so because either it is overloaded, selfish, malicious or broken.

Misbehaving peers should be identified and removed from the CHUM group. One possible solution is use a similar mechanism as in [103]. Each CHUM agent monitors the number of times each peer has failed and maintains a counter of failing tally for it. If the tally reaches a threshold for a peer, it is announced as a misbehaved node.

Security and privacy: The introduction of Bloom filters to represent CHUM notifications provides a way for a peer to remain anonymous within a group of strangers and disallows the proxy and other agents to examine the content of notifications. However, it does not prevent some malicious attacks. For example, the content of the notifications may be modified or a proxy may repeatedly broadcast stale notifications. Public-keys and digital certifications may be used to maintain a secure channel and determine whether attacks occur.

5.5 Summary

In this chapter, we introduce a new framework to share a single continuous WWAN notification channel fairly among a group of nearby mobile devices to avoid the long, idle and high energy consuming WWAN connections. We proposed a simple group management protocol to resolve the problem of high overhead in maintaining the precise group information in the highly dynamic and unreliable environment. The Bloom filter representation makes it possible for a mobile user to remain anonymous among strangers. A four-level subscription tree structure is introduced to reduce the number of Bloom filter comparison operation. The subscription tree may eliminate most of Bloom filter comparison operations.

Chapter 6

QAWBA: QoS Aware Wireless Bandwidth Aggregation

6.1 Introduction

Wireless users are demanding the same QoS requirements for the applications available on wired networks. *QoS Aware Wireless Bandwidth Aggregation (QAWBA)* is addressed to meet the high availability and high bandwidth QoS requirements by integrating the cellular wireless networks and the IEEE 802.11 wireless networks. A IEEE 802.11 based mobile ad hoc network is formed by mobile devices in order to share their cellular network connections. Several low bandwidth cellular network connections are aggregated to meet the high bandwidth QoS requirement of multimedia applications for which a single cellular connection is insufficient.

QAWBA requires a new QoS aware routing protocol for the MANET, which is much different from other existing QoS routing protocols. Typical QoS routing protocols

in a MANET find a path or multi-path from the source to the destination that meets certain QoS requirements (bandwidth and delay) [43, 42]. The source and destination are known before the QoS path discovery procedure. The QoS request is initiated by the source node. However, in QAWBA, only the destination node (client) is known at the beginning of the route setup. The sources, which are proxies, should be discovered by the routing protocol. The QoS request is initiated by the destination node. To the best of our knowledge, none of the existing routing protocol can be used to solve this problem.

To support QAWBA, we present a K-path on-demand QoS aware proxy discovery protocol to find suitable proxies in the MANET based on the bandwidth requirement and maximum hop limitation. Proxies are discovered along K paths starting from the client. Only K messages are needed for each session. The cellular bandwidth is reserved progressively among the nodes in the path.

The rest of the chapter is organized as follows. QoS aware on-demand routing protocol is presented in section 6.2. Performance evaluation and results are in section 6.3. Section 6.4 provides a summary of this chapter.

6.2 QoS Aware On-demand Routing

To provide QoS, QAWBA should integrate on-demand proxy discovery, bandwidth reservation and maintenance, and hop-by-hop routing. On-demand proxy discovery is provided by a K-path QoS aware proxy discovery algorithm. For each QoS session, K QoS requests are transmitted along different paths, to search for suitable

proxies. When a mobile node receives a QoS request, it determines the amount of cellular bandwidth to reserve for this session and then sends back a QoS reply for the reservation along the reverse path. The request is further forwarded to a neighbor, if needed. The total amount of bandwidth requested within K requests is the amount of bandwidth required for the session minus the bandwidth provided by the client itself. The reservation is successful if enough bandwidth is reserved within the time interval T_{setup}. Otherwise, the reservation request fails.

The overhead of connection maintenance and tear-down is eliminated by the soft-state reservation mechanism in which continuous packets from the base station destined for the client serve the purpose to provide the reservation update signals. An extended AODV [30] routing protocol is used for hop-by-hop routing in the MANET, using the QoS request (QRREQ) as route request (RREQ) and the QoS reply (QRREP) as route reply (RREP). The routing table is set up along the path from the client to the proxies in the process of proxy discovery.

6.2.1 Neighborhood Maintenance

Periodic "HELLO" messages are used by the mobile node to obtain the neighborhood information to conduct proxy discovery and traffic admission control. A node I includes the bandwidth available in its cellular network, in the MANET, and the consumed bandwidth in the MANET in the HELLO message. Every node maintains a neighborhood table composed of the information obtained from the HELLO messages.

A failure to receive a packet from a neighbor for a T_{nb} period means the link to that neighbor is broken.

6.2.2 Bandwidth Reservation Tables

The following three reservation tables are kept within each mobile node to store traffic and bandwidth allocations for the cellular network and MANET. This information is used to compute the available bandwidth for the cellular network and the MANET.

- *QoS session table*: For each QoS session, it records the session id (SID), total bandwidth required, maximum hop limitation in the MANET, total bandwidth currently reserved, status and a list of proxies contributing cellular bandwidth for this session (the client is considered to be a special proxy when providing cellular bandwidth for the session). The status of a QoS session could be REQUEST or RESERVED, which represents waiting for a QoS reply or successfully reserved.

- *Cellular flow reservation table*: For each cellular traffic flow, it records the session id, the client node id, the number of hops to the client node and the reserved bandwidth. A cellular reservation entry is inserted into the table when a cellular link reservation is made for a QoS session.

- *MANET flow reservation table*: It stores the session id, the source (proxy), the destination (client) node id, and the reserved bandwidth for each MANET flow. A reservation entry is inserted into this table when a QoS reply is received in the mobile node.

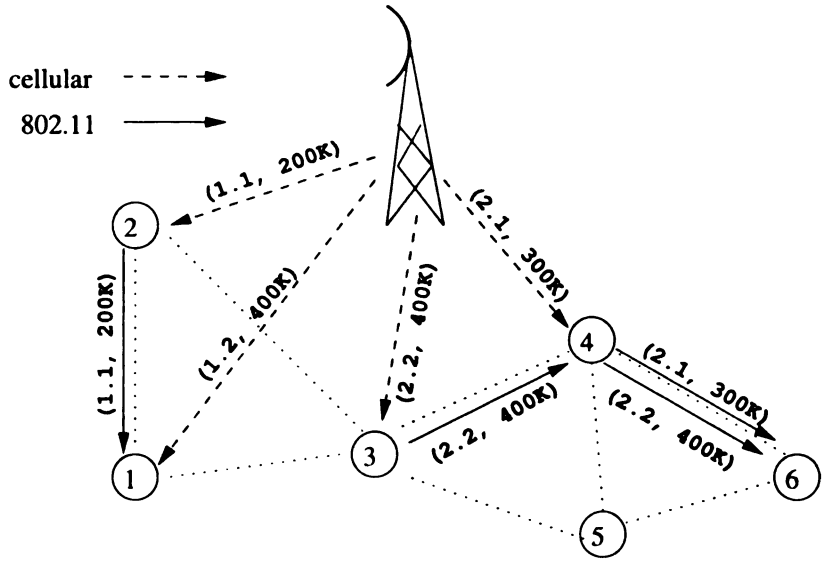


Figure 6.1: Example of the QoS session

Figure 6.1 shows an example of a QAWBA network with six mobile nodes. Session 1 and session 2 are running on node 1 and node 7, both requiring 600Kbps and 800kbps, respectively. The QoS requirements for session 1 have been fulfilled with nodes 1 and 2 reserving 400Kbps and 200Kbps cellular link bandwidth, respectively. Node 6 has discovered two proxies for session 2, which are nodes 3 and 4, providing 400Kbps and 300Kbps, respectively. Each traffic flow in the figure is represented by (SID.SubFlow, BW), indicating the session id (SID), subflow in the session (SubFlow) and reserved bandwidth (BW) in the subflow. Figures 6.2 and 6.3 show the corresponding tables for the seven nodes in the network of figure 6.1. Each proxy of the proxy list in the session table is represented as *(PID, hop count, reserved bandwidth)*, indicating the proxy ID, hop count to the client and reserved bandwidth in the cellular link.

Node	SID	Request	Reserved	MaxHop	Status	Proxies
1	1	600Kbps	600Kbps	3	RESERVED	(1, 0, 400Kbps) (2, 1, 200Kbps)
6	2	800Kbps	700Kbps	3	REQUEST	(4, 1, 300Kbps) (3, 2, 400Kbps)

Figure 6.2: QoS Session Table

MANET flows tables

Node	SID	BW	SRC	DST
1	1	200Kbps	2	1
2	1	200Kbps	2	1
3	3	400Kbps	4	7
5	2 2	400Kbps 300Kbps	3 4	6 6
6	2 2	400Kbps 300Kbps	3 4	6 6

Cellular flows tables

Node	SID	Client	Reserved	Hop Count
1	1	1	400Kbps	0
2	1	1	200Kbps	1
3	2	6	400Kbps	2
4	2	6	300Kbps	1

Figure 6.3: Cellular and MANET flow reservation tables

6.2.3 K-path proxy discovery algorithm

Generally, MANET routing protocols use flooding based discovery algorithms to find a path from the source to the destination, which are not suitable for the proxy discovery in QAWBA. The client only has local topology and traffic information, which is obtained from the periodic "HELLO" messages. Therefore, the number of possible proxies and their available cellular and MANET bandwidth are unknown for the client in the beginning of the discovery process. It is difficult for the node to determine the amount of reservation for a QoS session when receiving a QoS request.

In QAWBA, proxy discovery is done on-demand by a K-path discovery algorithm. An entry will be inserted into the QoS session table for a new QoS session S with the bandwidth requirement, $B_{req}(S)$, and the maximum hop limitation, $MHop(S)$. The status of the session is set to "REQUEST", indicating that it is in the process of

proxy discovery. The client X then decides the bandwidth reserved in its own cellular link for this session. If the available cellular bandwidth $B_{avail}(X, c)$ is greater than $B_{req}(S)$, the reserved bandwidth in the cellular link will be the requested bandwidth: $B_{res}(X, S, c) = B_{req}(S)$. Otherwise, X reserves all the available bandwidth for S: $B_{res}(X, S, c) = B_{avail}(X, c)$. A new entry will be inserted into the cellular flow reservation table indicating the new cellular flow reserved for the session S in the client X with bandwidth $B_{res}(X, S, c)$.

If the bandwidth requirement is fulfilled by the client's cellular link, the reservation process is finished and the status of the session is changed to "RESERVED". The application will be notified of the successful reservation. Otherwise, one or more proxies should be discovered to meet the remaining bandwidth requirement of this QoS session. K QoS request (QRREQ) messages are generated by the client. Each QoS request carries with part of the remaining bandwidth requirement, $(B_{req}(S) - B_{res}(X, S, c))/K$, and is sent to one of the neighbors with the highest available cellular bandwidth.

When mobile node I receives a QRREQ message for a QoS session S, with a request for bandwidth $B_{qrreq}(S)$, it rejects this QoS request if the current available MANET bandwidth is smaller than the consumed MANET bandwidth for the session S. A QoS failure (QFAIL) message is then sent back to the client. Otherwise, I determines the amount of cellular bandwidth it may reserve for S based on $B_{qrreq}(S)$ and the current available cellular bandwidth, $B_{avail}(I, c)$. Similar to the reservation procedure in the client, the reserved bandwidth is the minimum of the available cellular bandwidth and the requested bandwidth. $B_{res}(I, S, c) = min\{B_{qrreq}(S), B_{avail}(I, c)\}$

QoSAwareProxyDiscovery(sid, bw, max_hop)

```
1.  K=dynamicK(bw)  // get K value based on request bw
2.  if (k<0) then
3.    return notify_app(sid, FAIL) // fail
4.  endif
5.  if (k==0) then
6.    reserved_bw=bw  // enough cellular bw
7.  else
8.    reserved_bw=cellular_availbw  // not enough cellular bw
9.  endif
10. request_bw=bw - reserved_bw
11. if (request_bw>manet_availbw) then
12.   return notify_app(sid, FAIL)  // not enough MANET bw
13. endif
    // insert a new QoS Session entry
14. QoSSession=insert_QSession(sid, bw,
        max_hop, reserved_bw, REQUEST)
15. if (reserved_bw>0) then
16.   cflow=insert_cflow(sid, reserved_bw, node_id, 0)
17.   QoSSession.insert_proxy(sid, node_id, reserved_bw)
18. endif
19. if (request_bw==0) then
20.   QoSSession.status=RESERVED
21.   return notify_app(sid, RESERVED)  // successful
22. endif
    // send K QoS requests to neighbors
23. sendQoSRequest(node_id, K, sid, request_bw/K, max_hop)
24. return notify_app(sid, REQUEST)  // wait for reply
```

Figure 6.4: QoS Aware Proxy Discovery Algorithm

If $B_{res}(I, S, c) > 0$, a new entry is inserted into the cellular flow table in the mobile node I and I is considered as one of the proxies for session S. A QoS reply (QRREP) message is sent back to the client for the new reservation in I's cellular link. If I cannot provide enough cellular bandwidth for session S, the QRREQ request will be forwarded to one of the neighbors with an updated bandwidth request: $B'_{qrreq}(S) = B_{qrreq}(S) - B_{res}(I, S, c)$. I sends back a QFAIL message to the client if the maximum hop limitation is reached or there is no available neighbor for the

QRREQ request. The QRREQ request stops propagating if the bandwidth request is satisfied ($B'_{qrreq}(S) = 0$).

```
OnRecvQoSRequest(client, hop_count, sid, bw, max_hop)
1.  if (bw>manet_availbw) then
2.    sendQoSFail(client,node_id,sid)
3.  endif
4.  if (cellular_availbw>bw) then
5.    reserved_bw=bw
6.  else
7.    reserved_bw=cellular_availbw  // reserve all available
8.  endif
9.  if (reserved_bw>0) then
10.   cflow=insert_cflow(sid,reserved_bw,client,hop_count)
11.   mflow=insert_mflow(sid,client,node_id,reserved_bw)
12.   sendQoSReply(client,node_id,sid,reserved_bw)
13. endif
14. request_bw=bw - reserved_bw
15. if (request_bw>0) then
16.   if (max_hop - 1<=0) then
17.      sendQoSFail(client,node_id,sid)
18.   else
        // send 1 request to neighbor
19.      sendQoSRequest(client,1,sid,request_bw,max_hop-1)
20.   endif
21. endif
```

Figure 6.5: Processing of QoS Request Message

When the client X receives a QRREP reply message, it adds the amount of reserved bandwidth in the QRREP message to the reserved bandwidth field for the corresponding session entry in the session table. If the reserved bandwidth is equal to the requested bandwidth for session S, the reservation for S is successful. When the client X receives a QFAIL message with the amount of unreserved bandwidth $B_{unres}(S)$ for session S, it may choose a new neighbor for the retry of QRREQ requesting for $B_{unres}(S)$. If not enough bandwidth is reserved for the session S within the T_{setup} interval, the reservation has failed.

Figure 6.4 shows details of the K-path discovery algorithm. The processing of QoS request messages and QoS reply messages are presented in figures 6.5 and 6.6.

```
OnRecvQoSReply(client, proxy, sid, reserved_bw, hop_count)
1.  mflow=insert_mflow(sid, client, proxy, reserved_bw)
2.  if (client_id==node_id)
       // the reply is for me
3.    QoSSession=session_lookup(sid)
4.    QoSSession.reserved=QoSSession.reserved+reserved_bw
5.    QoSSession.insert_proxy(sid, proxy_id, hop_count)
6.    if (QoSSession.reserved==QoSSession.required)
          // the bandwidth requirement fulfilled
7.       QoSSession.statud=RESERVED
8.       return notify_app(sid, RESERVE_SUCCESS)
9.    endif
10  else
11.   forward_reply()
12. endif
```

Figure 6.6: Processing of QoS Reply Message

The value of K could be predetermined by the client or be dynamically computed based on the bandwidth requirement in the QoS session, current available bandwidth and the maximum hop limitation. A smaller K value results in a longer path and higher risk of exceeding the maximum hop limitation. On the other hand, the larger K increases the contention on the client node. Therefore, the optimal K value should be the smallest one that meets the maximum hop limitation.

We compute the value of K based on the assumption that the network load is evenly distributed among all mobile nodes. Although this assumption is not always true, it is adequate for estimating the value of K. K is then computed as the minimum value of the maximum possible value of K, $maxK$, and $B_{req}(S)/(B_{avail}(X,c) * MHop(S)) + 1$.

$MaxK$ is determined by the minimum value of the number of available neighbors N_{nb} and a predetermined parameter $MAXK$: $maxK = min\{N_{nb}, MAXK\}$.

$$K = min\{maxK, \frac{B_{req}(S)}{B_{avail}(X,c) * MHop(S)} + 1\}$$

6.2.4 Computation of Available Bandwidth

To determine the admission of a QoS session and the amount of cellular bandwidth reservation, we need to know the available bandwidth in the cellular link and the MANET. Here, we assume that the mobile node knows the current link capacity in its cellular interface and MANET interface.

The cellular link could be viewed as a dedicated point-to-point link from node I to the base station. The available bandwidth in the cellular link of I, $B_{avail}(I,c)$, could be computed as the cellular link capacity, $B_{cap}(I,c)$, minus the total cellular bandwidth that has been reserved in the cellular flow table: $B_{avail}(I,c) = B_{cap}(I,c) - \sum_S B_{res}(I,S,c)$.

In the MANET, the radio channel of each node is shared by all neighbors. A node can successfully use the channel only when all its neighbors do not transmit or receive packets at the same time. We use the algorithm in [42] to estimate the upper bound limit of available bandwidth and consumed bandwidth in the MANET. The available bandwidth on the MANET could be computed as the MANET capacity minus the total consumed traffic in I's neighbors $N(I)$. $B_{avail}(I,m) = B_{cap}(I,m) - \sum_{J \in N(I)} B_{consumed}(J,m)$. The consumed bandwidth depends on the location of the node I in the MANET flow. For a MANET flow with bandwidth B, if I is the source

or destination, the consumed bandwidth is the flow bandwidth B. Otherwise, the consumed bandwidth is twice the flow bandwidth ($2 \times B$) since I should receive and send packets in this flow, which cannot be done simultaneously.

When receiving a QoS request with $B_{req}(S)$ requirement for session S, two MANET flows are needed if I could not fulfill this QoS request alone. I is the source of one MANET flow with bandwidth $B_{res}(I, S, c)$ and the intermediate node for the other MANET flow with bandwidth $B_{req}(S) - B_{res}(I, S, c)$. Therefore, the total consumed bandwidth for S,$B_{consumed}(I, S, m)$, in the MANET is: $B_{res}(I, S, m) + 2 \times (B_{req}(S) - B_{res}(I, S, m))$.

6.2.5 Failure Recovery and Automatic Resource Release

A MANET is characterized by frequent topology change and unreliable physical media. QAWBA needs to provide a mechanism to detect QoS violations and communication failures. A common approach is to use the HELLO message in the routing protocol, in which failure to receive a HELLO message from one of its neighbors within a timeout period indicates its failure. The node then sends a route error message back to the client node. Due to the bandwidth consumption of sending HELLO messages, the frequency of neighbor detection must not be too high. This prevents the system to detect a broken route quickly.

In QAWBA, we use the QoS timeout interval (T_{int}) to detect route breakage at both the client and proxy side. At the client side, if a proxy is not heard by the client for T_{int} (no packets are received from the proxy), the proxy is believed to be out or the

route is broken. The client then eliminates the QoS entry of the broken proxy and updates the reserved bandwidth of the current session. The base station is informed about the broken proxy and new arrived packets will not be sent to the client via that proxy. A new round of proxy searching starts if the reserved bandwidth is less than the required bandwidth. At the proxy side, if within a timeout interval it fails to receive any packets from the base station for the client node, the proxy would release the cellular bandwidth reservation and tear down the ad hoc network routes. The intermediate nodes that are along the paths from the proxy to the client node also maintain a timeout timer. If no traffic exchange occurs between the proxy and the client, the reserved ad hoc network routes are also released.

Only when the client node detects a route breakage, the system attempts to recover. Within the recovery process, the client can either unicast one QoS search request or start a new K-path QoS searching session if a single QoS fails. The newly found proxies join the existing session and provide service to the application. If the client fails to find new proxies, the application has to choose either to continue running under a relaxed QoS service or to try again later.

6.3 Performance Evaluation

6.3.1 Simulation Model

We implement the QAWBA protocol in the *ns*2 network simulator. Twenty mobile nodes are randomly placed in the area of $300m \times 300m$. Each node has a cellular

interface and an IEEE 802.11b interface. The cellular interface is used to communicate with the base station, which in turn connects to the Internet. It uses an one-hop routing scheme, from the base station to the mobile node. Different cellular interfaces use different radio channels and do not interfere with each other. The link capacity of the cellular link is uniformly distributed between 28.8Kbps to 2Mbps.

We use the IEEE 802.11b implementation from ns-2 version 2.1*b*9, where 11Mbps data rate is supported at the 100 meter range. The radio propagation model for IEEE 802.11b uses the two-ray ground reflection model. Node mobility is set according to the random waypoint model. Each node moves toward a random destination within the field at a specified speed. After reaching the destination, the node pauses for a certain amount of time and then starts moving again. A QoS request is generated in the mobile node with a predetermined bandwidth requirement. The length of each QoS request is fixed to be 20 seconds. The time interval between two QoS request is exponentially distributed with mean δ.

Two metrics are evaluated by the simulation: the QoS admission rate (r) and cellular link utilization (u). The QoS admission rate r_I for node I is defined as the ratio of the number of successful QoS requests, $N_{suc}(I)$, and the total number of QoS requests, $N_{total}(I)$. The value r is defined as the average of admission rate of all n mobile node: $r = \frac{1}{n} \times \sum_I \frac{N_{suc}(I)}{N_{total}(I)}$. The link utilization for node I is defined as the total link capacity times the simulation running time divided by total bandwidth in use in this time period for I. Suppose in time period T_j, the bandwidth used by the cellular link

of node I is $B_{res}(I,c)$, the total bandwidth in use in the time period T_j is defined as $T_j \times B_{res}(I,c)$. Therefore, u could be defined as: $u = \frac{1}{n} \times \sum_I \frac{\sum_j (T_j \times B_{res}(I,c))}{T \times B_{cap}(I,c)}$. QAWBA is compared with a simple scheme, in which a QoS request is successful if the mobile node currently has enough available cellular bandwidth, otherwise, the it fails.

6.3.2 Simulation Results

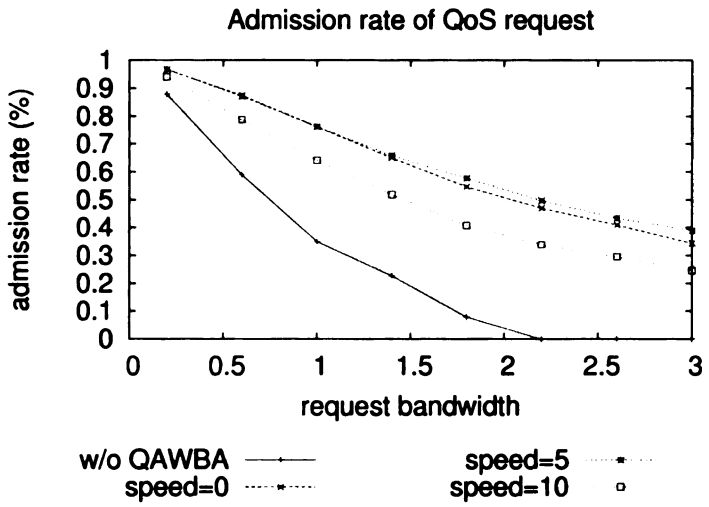


Figure 6.7: QoS request admission rate with different request bandwidths, $\delta = 40$

Figure 6.7, figure 6.8 and figure 6.9 present the average admission rate r of QoS request with different requested bandwidth (B_{req}), different traffic load ($\delta = 40$, $\delta = 20$, and $\delta = 10$) and different node moving speed ($speed = 0$, $speed = 5$, $speed = 10$). Without the help of QAWBA, the admission rate drops dramatically with the increase of the size of the QoS request. The admission rate is less than 22% when $B_{req} > 1.4Mbps$ and $\delta = 40$. It drops to zero when the B_{req} is greater than 2Mbps, the maximum of the cellular link capacity. In QAWBA, the admission rate is

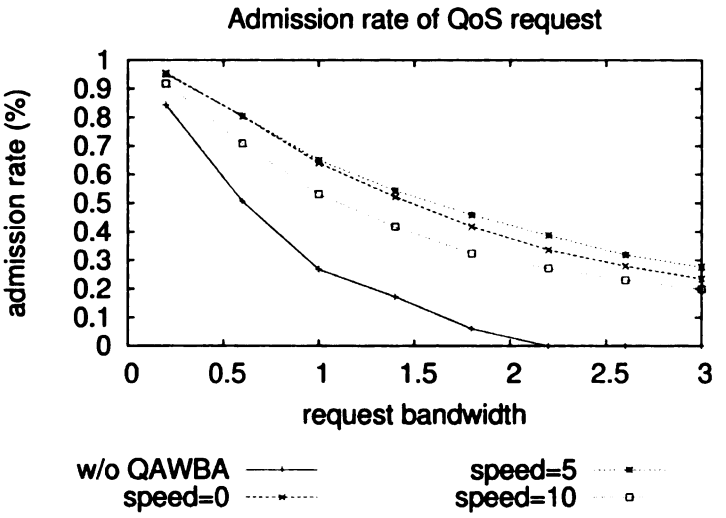


Figure 6.8: QoS request admission rate with different request bandwidths, $\delta = 20$

much higher than the one without QAWBA. The mobile nodes are able to accept more QoS requests than are possible under normal condition. For example, in figure 6.7 the QAWBA approach enables the admission rate to reach 50% when $\delta = 40$ and $B_{req} = 2Mbps$. Without QAWBA, none of the QoS requests can be accepted. With heavier system traffic load (smaller δ value), the increase of the admission rate in QAWBA becomes more significant. In figure 6.9, when $\delta = 10$ and $B_{req} = 1.4Mbps$, the admission rate in QAWBA is 35% while it is only 11% without QAWBA. QAWBA helps to increase the admission rate by 318% in this case. Mobility has some effect on the admission rate in QAWBA. The admission rate drops a little as the mobile nodes move faster. It may be explained due to an increased number of broken routes during the proxy discovery when the mobile nodes move faster.

Figure 6.10, figure 6.11 and figure 6.12 shows the utilization of the cellular link u with different requested bandwidth (B_{req}), different traffic load ($\delta = 40$, $\delta = 20$, and $\delta = 10$) and different node moving speed ($speed = 0$, $speed = 5$, $speed = 10$).

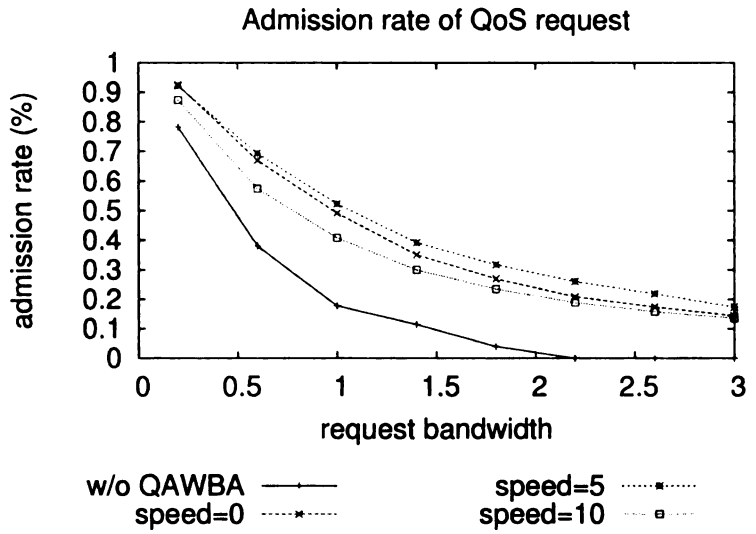


Figure 6.9: QoS request admission rate with different request bandwidths, $\delta = 10$

Without QAWBA, the utilization rate is very small, less than 10% in the most cases. The QAWBA scheme significantly increases the utilization of the cellular link under different system loads. For example, in figure 6.10 the utilization rate is only 8.9% with $\delta = 40$ and $B_{req} = 1.4Mbps$ without cooperation. In QAWBA, the utilization rate increases 3.5 times and reaches 31.3%. The increase is more significant with larger requests. Mobility also decreases the utilization of the cellular link due to its effect on the admission rate.

The average number of proxies in QAWBA is shown in figure 6.13. It increases nearly linearly with the increase of the requested QoS bandwidth. The more bandwidth requested, the more proxies are needed to contribute cellular bandwidth. As shown in figure 6.14, the delay of proxy discovery process increases with the increase of the requested bandwidth. Since more proxies should be found for a larger request, the delay of the proxy discovery is also increased.

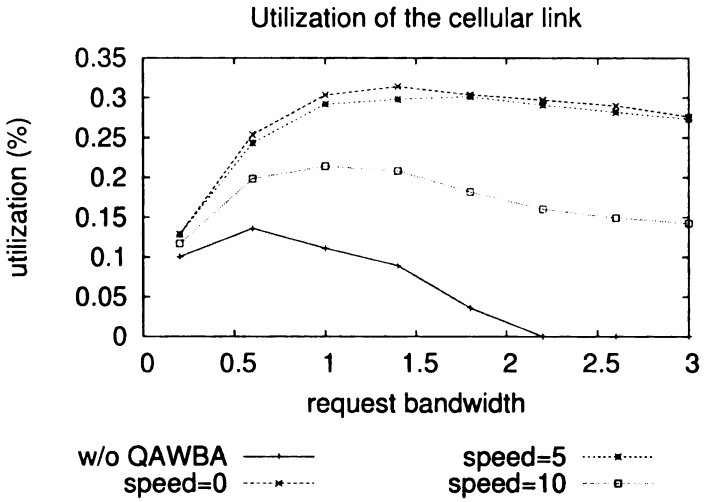


Figure 6.10: Cellular link utilization with different request bandwidths, $\delta = 40$

6.4 Summary

Current cellular networks cannot meet the QoS requirements of many multimedia applications. Although the cellular interface provides "anywhere, anytime" network access, IEEE 802.11 based network interfaces have become the de factor interface for many mobile devices. We provide in this chapter a QAWBA system that utilizes both the cellular network interface and the IEEE 802.11 ad hoc network for an integrated network architecture that provides QoS aware wireless bandwidth aggregation. Mobile nodes form a mobile ad hoc network via their IEEE 802.11 based network interface. The capacity of several low throughput cellular links are shared by all mobile nodes to provide better QoS support for the application. The simulation result shows that QAWBA could significantly increase the utilization of the cellular resource and the admission rate of the QoS requests.

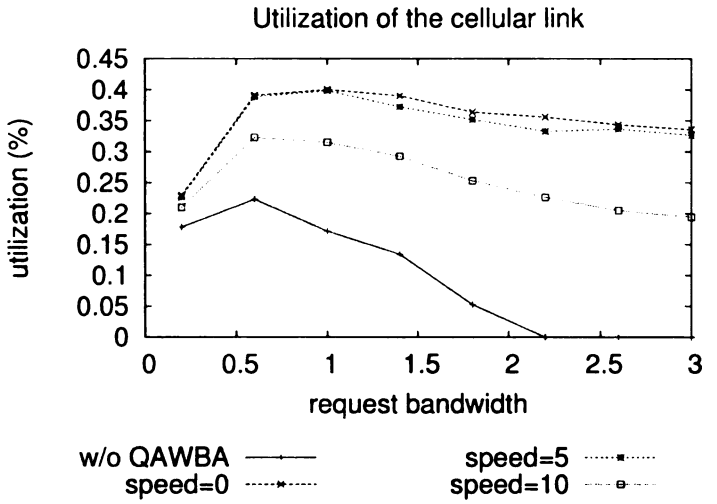


Figure 6.11: Cellular link utilization with different request bandwidths, $\delta = 20$

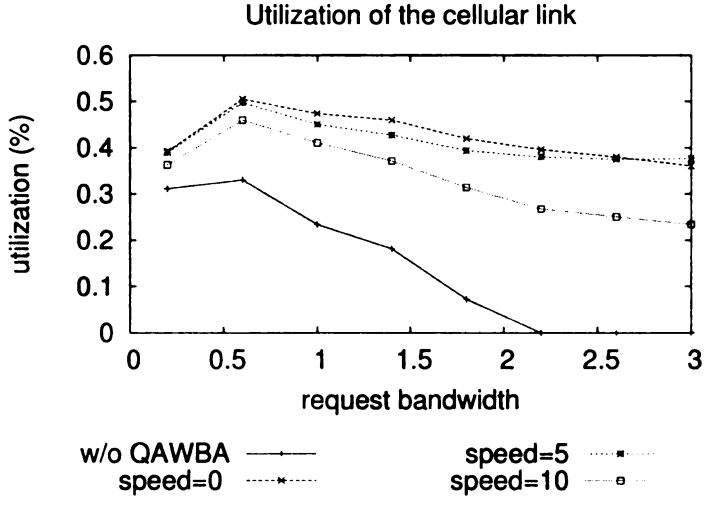


Figure 6.12: Cellular link utilization with different request bandwidths, $\delta = 10$

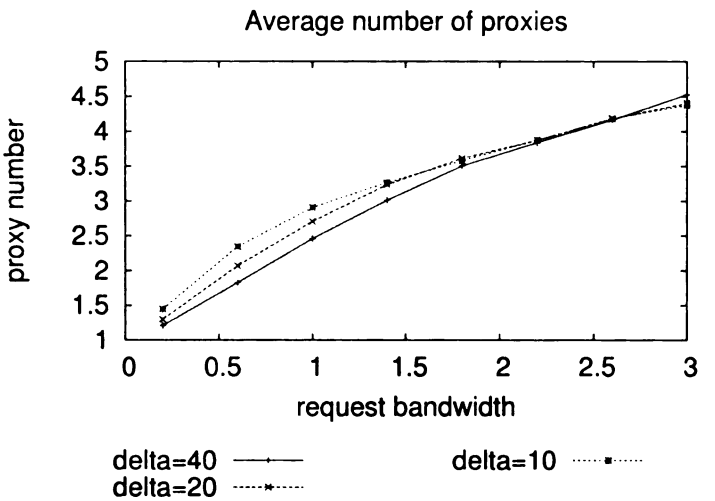


Figure 6.13: Average number of proxies in QAWBA

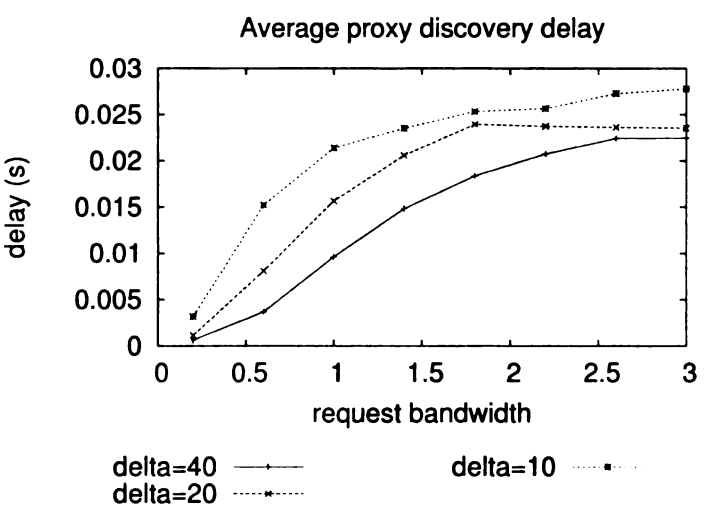


Figure 6.14: Average delay of proxy discovery in QAWBA

Chapter 7

Cooperative Multiple Paths to Reduce File Download Latency

7.1 Introduction

In this chapter, we propose a novel architecture that integrates cellular data network and IEEE 802.11-based MANET for the purpose of improving file download latencies. No special hardware is required in the mobile nodes. The base station of the cellular network is not involved in the file downloading process. We also devise a suite of protocols that enable cooperative parallel file downloading, including proxy discovery, ad-hoc routing, and failure recovery. Last, we present a new trust model to promote fairness among the participants. Through simulations and experiments, we show that cooperative parallel file downloading can significantly improve an individual's file download performance.

Several assumptions are made in this chapter. First, the participating mobile devices are equipped with both cellular-based and IEEE 802.11-based wireless network interfaces. Second, there is no radio interference between the IEEE 802.11-based MANET and the cellular data network since they operate under different frequency band ranges. Third, the Internet file servers support partial file downloading. For example, the HTTP 1.1 byte-range header may be used to indicate which portion of the file to obtain from the web server. Last, mobile users may be charged by the number of bytes transmitted, by the length of time connected, or have unlimited access within the cellular data network. They have more incentive to cooperate to download a file in parallel if there is unlimited access or per-minute service charges. They are more sensitive to fairness when there is a per-byte service charge. A carefully designed trust model and credit system is proposed to promote the incentive of cooperation.

The rest of the chapter is organized as follows. In section 7.2, the process of proxy discovery and file downloading are introduced. The trust model and credit system is given in section 7.3. Performance evaluations and results are presented in section 7.4.

7.2 Proxy Discovery and File Download

We present an on-demand proxy discovery algorithm for the client to discover possible proxies for a file downloading session. The file is split into small portions. A new round of proxy discovery is needed for each portion. The additional delay introduced by proxy rediscovery is reduced by pipelining the file transmission request. An extended

AODV [30] routing protocol is used to establish the hop-by-hop routing table from the proxy to the client.

7.2.1 On-demand Proxy Discovery

The client floods a *proxy discovery request* (PDREQ) message within a given range in the MANET in order to find the nodes that agree to act as proxies to download a portion of a file. The PDREQ message carries the client's address (SR), a sequence number (SEQ) that is incremented every new round of proxy discovery, a request broadcast range (RBR) value that is decremented every time the message is rebroadcasted, and the size of the requested file portion.

When receiving a PDREQ message, the node compares the sequence number with the largest one it has for the source and drops the PDREQ message with equal or smaller sequence number. The node then makes a decision whether or not to act as a proxy or a forwarder, which is based on the trust and credit information of the client, its current network traffic load, the size of the request portion and other considerations, such as battery power. The node decrements the RBR value and rebroadcasts the request for a positive RBR and a positive forwarder decision, with its address attached in the PDREQ as a forwarder (FR). With a positive proxy decision, the node returns a *proxy discovery reply* (PDPLY) message. Otherwise, the request is dropped. A node can act as a proxy and a forwarder at the same time. The PDPLY message is sent back through the reverse route by the new proxy, containing the SEQ, a sequence of forwarder addresses and the address of the proxy (PR). Here we assume the ad

hoc link is full duplex, which is true in IEEE 802.11 based networks. A timeout value is associated with the PDPLY message indicating the time period that the proxy reserves its cellular link for the request. The client must send out a file portion download request within that period to prevent a timeout.

A set of available proxies is kept in the client for each file downloading session. A new member is inserted into the set when a new proxy is discovered. The client uses a pre-defined *degree of parallelism* (DOP), which is the number of proxies plus one (the client's cellular link), to limit the number of proxies used simultaneously in order to reduce collisions in the IEEE 802.11-based MANET. There are several instances when a client selects an available proxy from the set of discovered proxies. The client selects when (1) downloading a new file portion after finishing downloading one; (2) when receiving a new proxy reply while not exceeding the DOP limit; or (3) when it has a broken connection to a proxy due to mobility. If there is an empty available proxy set, the client starts downloading a file portion via its own cellular link and sends a new PDREQ message for a new round of proxy discovery. Proxy selection is based on trust information, hop count, and the round trip time (RTT) estimated via the proxy discovery process. The client selects a trusted proxy with the shortest round trip time. In section 7.3, we present a trust model to develop a trust relationship between nodes.

In Figure 7.1, six mobile devices form an IEEE 802.11-based MANET. Client A starts a new file download by splitting the file into several portions with size 1000 bytes. It broadcasts a PDREQ message to its neighbors B, C and D with SEQ (1), file portion size (1000), broadcasting range RBR (3) and source address (A). D drops the request

with a negative forwarder and proxy decision. C returns a PDPLY message with the route $SR = A, PR = C$, with a positive proxy decision. B would like to act as a forwarder but not as a proxy. It attaches its address into the original request and rebroadcasts the message to its neighbor E. With a positive proxy decision, E returns a PDPLY message with the route $SR = A, FR = B, PR = E$ to B, which in turn sends it back to A. Therefore, proxies E and C are discovered by the client A via a round of PDREQ/PDPLY message exchange, agreeing to download a file portion of size 1 Kbyte.

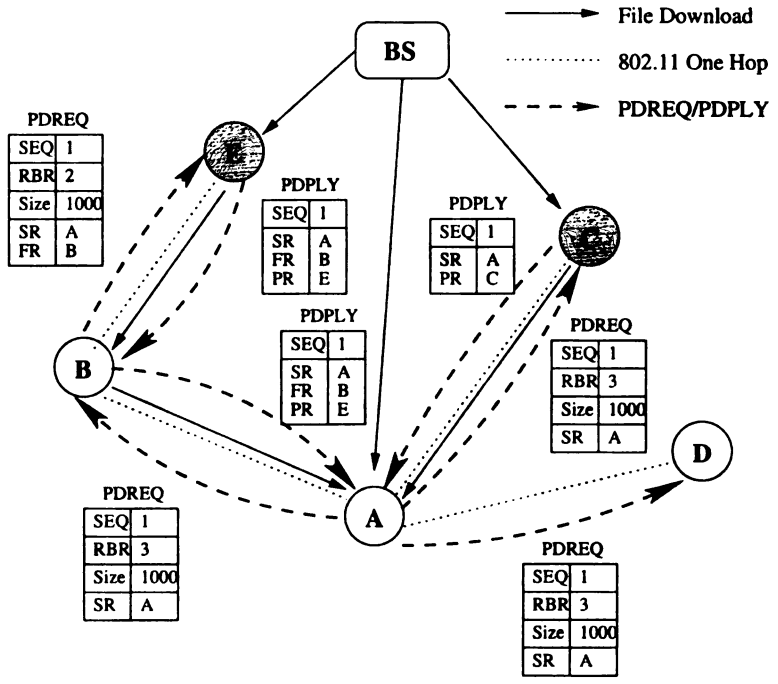


Figure 7.1: On-demand Proxy Discovery Example

An extended AODV protocol is used to establish a hop-by-hop routing table from the proxy to the client. The PDREQ and PDPLY message is used in the same manner as the route request (RREQ) and the route reply (RREP) in AODV. When receiving a PDREQ message, a routing entry is inserted into the routing table for the client node.

When receiving a PDPLY message, a routing entry is inserted for the proxy node. Therefore, the routing table is established along with the proxy discovery process. The forwarder list associated with the PDREQ message and the PDPLY message is only used by the trust model in the client and the proxy since the client should explicitly know who is contributing to the file download.

The on-demand proxy discovery protocol provides a simple and efficient solution for the routing problem. The client is able to locate as many proxies as possible for every file portion even under an unreliable and highly dynamic environment. The discovered proxy can be used for downloading any file portion with the same size as contained in the PDREQ message. Therefore one round of PDREQ/PDPLY message exchange can find several available proxies and initiate several portions to download simultaneously. In the previous example, two proxies are found by one PDREQ message, which can start downloading two portions to the client. On the other hand, the client starts file downloading immediately from its own cellular link, and gradually increases the degree of parallel downloading with the newly discovered proxies. It guarantees that the client has at least the same performance of a normal 3G network download even if it fails to discover any proxies.

7.2.2 Pipelining Requests

After selecting a proxy from the available proxy set, the client starts downloading by sending a *file portion download request* (FDREQ) message to the proxy. This message includes the location of the file, the beginning and the ending position of

the requested file portion, and the source address of the client. When receiving the FDREQ message, the proxy sets up a TCP connection to the Internet file server via its cellular link on behalf of the client and starts downloading the requested file portion. The proxy forwards all packets received from the Internet server to the client by establishing another TCP connection between the proxy and the client via the IEEE 802.11-based MANET. After finishing downloading of a file portion, the client selects another proxy from the available proxy set or performs another round of proxy discovery with an empty available set. After the client receives all file portions, it reconstructs the document.

For every file portion, there is a proceeding proxy discovery and a connection setup process. To avoid the idle time period during which no data is transmitted (see Figure 7.2), the proxy discovery and file downloading process may be pipelined. Since it is very likely that a proxy will agree to download another file portion, the client may send a new proxy discovery request directly to a current proxy just before it finishes downloading the file portion. The proxy and forwarders along the route may also make decisions for the new request while the current download is occurring. When the client receives a positive reply from this proxy, a new file download request may be sent without waiting for the current one to be finished. With pipelining, the client may reuse the TCP connection to the proxy to avoid slow start phases and avoid the idle time period of proxy rediscovery and connection setup.

Each file portion should be small enough to provide fine granularity of striping and deal with the unreliability of the proxy. However, it should also be sufficiently large to limit the idle times between block requests in comparison to the transmission time

of a block (see Figure 7.2). On the other hand, pipelining requires a minimum portion size and therefore a maximum number of portions. Suppose S is the file size, and B is the number of portions, the portion size should be such that $\frac{S}{B} > RTT \times \mu$, where μ is the download rate from the client via this proxy and RTT is the round trip time from the client to the Internet server via this proxy.

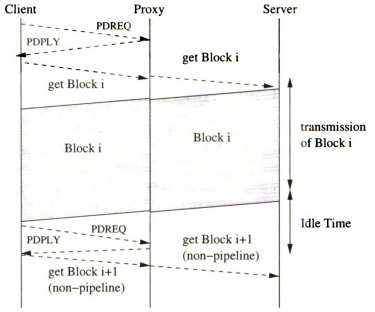


Figure 7.2: Block Request without Pipeline

7.2.3 Failure Recovery

The mobile devices (the client, forwarders, or proxies) can move out of range from the established downloading path, which results in broken routes. To detect and recover from route failure, a timeout that is two times the round trip value from the client to the file server is set up in the client for each proxy in use. If the next packet from a proxy does not arrive to the client by the time the timeout expires, the proxy is considered to have failed. The rest of the file portion will be combined into other

portions for the next round of downloading. If the client receives packets from failed proxies, it will inform the proxies to stop downloading by sending them *stop download requests* (STREQ).

On the other hand, the IEEE 802.11 MAC layer calls a call-back function to inform the mobile device of next-hop failure. This feature can be used by the proxy or by the forwarders to inform the proxy to stop downloading when a route failure is detected.

7.3 Trust Model and Feedback Report

The proxies contribute their cellular links to aid file downloading. Mobile users may be charged by their service plans for the packets they forward to the client. Even with an unlimited service plan, mobile nodes consume battery power to serve as packet forwarders. Therefore, some scheme should be developed that encourages cooperation and kicks out "free riders" who benefit as clients but do not serve as proxies.

7.3.1 Distributed Trust Model

In [170], we presented a distributed trust model and a credit system to promote cooperation among a group of strangers in the highly mobile environment. In this model, peers (mobile nodes) evaluate the trust they have for each other. The trust knowledge is further used in the credit system to ensure that each peer receives benefit from the system in proportion to the service it provides. A slightly modified trust model and credit system can be used in this chapter to promote cooperation.

A trust value $TRUST_{i,j}$ is used to represent the trust level peer p_i has of peer p_j. It will be derived from two different types of information: **perceived trust** $PT_{i,j}$, which is the information obtained from a peer's direct interaction with another peer; and **recommended trust** $RT_{i,j}$, which is obtained from a well-known trust server. Perceived trust is updated based on the honesty and the number of bytes downloaded by the service providers (proxies or forwarders). The client increases the perceived trust of an honest proxy or forwarder as more bytes are downloaded through them. The well-known trust server records the long-term activities for peers and develops their recommended trust. Peers issue feedback reports to the trust server with regard to other peer's honesty to download files. There may exist disagreements among feedback reports from proxies, forwarders or the client regarding the amount of data downloaded. The trust server never tries to resolve the disputes, but only records the total number of bytes this mobile user claimed to serve as a proxy or forwarder but not recognized by the destination client and the total number of bytes other peers claimed to provide for this mobile user but has been denied.

Feedback Report Protocol

The peers issue feedback reports to the trust server independently after one file portion downloading has finished. A *feedback report protocol* based on public key encryption is developed to provide a secure way to report feedback information to the trust server. Other techniques such as a digital signature may also be used in this protocol.

Fig. 7.3 shows the diagram of feedback report protocol. In this figure, C represents the client, P represents the service provider, a proxy or forwarder and TS represents

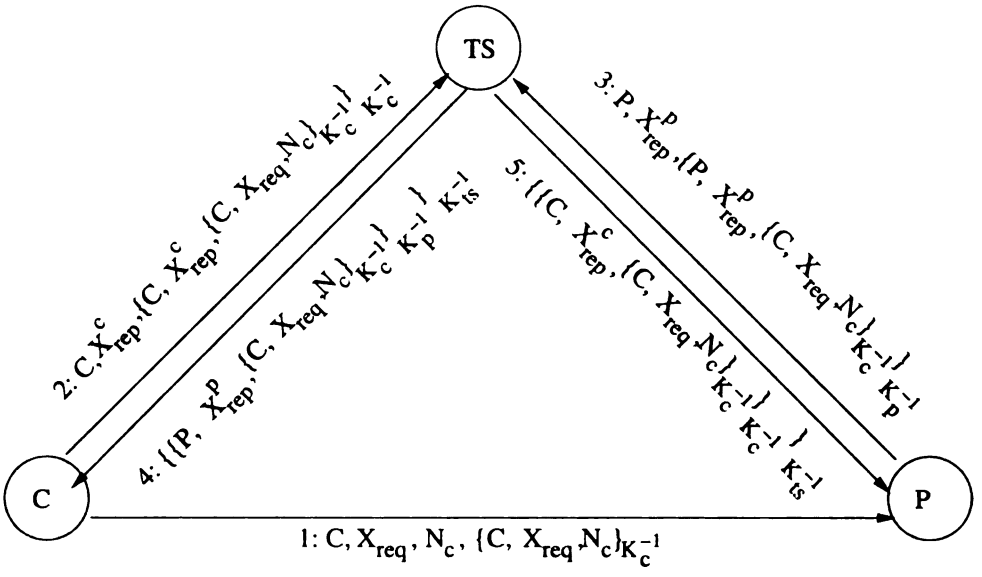


Figure 7.3: Feedback Report Protocol Diagram

a trust server. The protocol could be represented in three stages: *service request*, *feedback report* and *report forward*. In the service request stage, the client sends a file portion download request along with an unique nonce (sequence number or time stamp) encrypted with the private key of the client to the service provider. The service provider performs the file download service for the client. After the service transaction has finished, the client and the service provider issue feedback reports to the trust server independently along with the encrypted service request in the feedback report stage. In the last report forward stage, the trust server forwards the feedback report from the client to the service provider and also from the service provider to the client.

There are five messages exchanged between the client C, the service provider P and the trust server TS regarding one file downloading service. The details of the messages are shown below, where K_c^{-1}, K_p^{-1} and K_{ts}^{-1} are private keys for C, P and TS respectively. X_{req} is a service request, containing the number of bytes requested. X_{rep}^c

and X^p_{rep} are feedback reports from C and P reporting the number of bytes received and provided in this service transaction. N_c is the unique nonce generated by the mobile client for this service transaction.

1. $C \rightarrow P : C, X_{req}, N_c, \{C, X_{req}, N_c\}_{K_c^{-1}}$

2. $C \rightarrow TS : C, X^c_{rep}, \{C, X^c_{rep}, \{C, X_{req}, N_c\}_{K_c^{-1}}\}_{K_c^{-1}}$

3. $P \rightarrow TS : P, X^p_{rep}, \{P, X^p_{rep}, \{C, X_{req}, N_c\}_{K_c^{-1}}\}_{K_p^{-1}}$

4. $TS \rightarrow C : \{\{P, X^p_{rep}, \{C, X_{req}, N_c\}_{K_c^{-1}}\}_{K_p^{-1}}\}_{K_{ts}^{-1}}$

5. $TS \rightarrow P : \{\{C, X^c_{rep}, \{C, X_{req}, N_c\}_{K_c^{-1}}\}_{K_c^{-1}}\}_{K_{ts}^{-1}}$

The feedback report protocol could be idealized and authenticated by BAN logic [172]. Several conclusions can be drawn by the analysis of the protocol. We conclude that two feedback reports are sent to TS by P and C independently. TS honestly forwards these feedback reports to the counterpart. P and C read the feedback report from the counterpart and believe that TS also gets these reports. The details of the protocol analysis are omitted due to page limitation.

Dispute Resolution

Feedback reports from the client and the service providers may not be consistent with each other for a given download of a portion of a file. The disagreement may be caused by the unreliability of the mobile environment or a malicious peer. When a dispute occurs, it could be difficult, if not impossible, for the trust server to determine who

is telling the truth. To avoid complexity within the reporting protocol, we simply let the trust server record the difference between the amount of service reported in the disputed fields in the trust table for both the client and the service provider. The disputed fields can be used as indicators for the trust of a peer. A distrusted peer may have large values in these fields.

For example, the client C may report that P downloads 600 bytes for C as a proxy. However, P may report that it downloads 1000 bytes for C. The 400 bytes difference in the reports may be caused by route failure or cheating in P or C, which is difficult to distinguish by the trust server. Therefore, the trust server simple adds 600 bytes into P's total bytes that have been downloaded as a proxy and C's total bytes downloaded by a proxy simultaneously. The other 400 bytes is added in both P's and C's proxy disputed fields.

The trust server matches feedback reports issued by different peers for the same portion of a file that has been downloaded. This is based on the unique nonce, the identification of the destination client and the identification of the service provider. The trust server may have to cache the reports for a short period since different peers may issue the reports at different times. The unmatched reports will be counted as disputes, which will be recorded at both sides. This mechanism encourages peers to issue reports to avoid a reputation to be ruined.

There are several mechanisms provided in the feedback report protocol to prevent peers' reputation to be ruined by malicious reports. First, the unique nonce included in the service request makes the client and the service provider able to issue only one feedback report for each portion of a file downloaded. Second, the disagreement is

recorded at both sides, making cheating reports unattractive. The reputation of the cheating peer will be ruined by a false feedback report. Third, the amount of service requested by the client is included in the secured service request. The reported service amount cannot exceed that amount, therefore limiting the reputation damage caused by one false feedback report. Last, the trust server forwards the feedback reports from the client to the service provider and from the service provider to the client. The malicious feedback reports are read by the affected peers, which can refuse to provide further service in the future.

7.3.2 Proxy and Forwarder Decision

The proxy and forwarder decisions are made based on the trust information and the available credit for the service requester. The peer will not provide service for distrusted peers (peers with negative trust value) or peers without positive available credit. The trust value is based on the perceived trust and the recommended trust 8, where the perceived trust is updated based on the transaction conducted between two peers and the recommended trust is computed based on the trust information obtained from the well-known trust server. On the other hand, a peer will refuse to continue to forward a request if the identification of a malicious peer is included in the route path. The client also filters out distrusted proxies or forwarders in the proxy selection process mentioned in section 7.2. Gradually, the malicious users are excluded from the system without anyone accepting their service, forwarding their requests, or providing them service.

7.4 Performance Evaluation

We established a testbed and conducted experiments on a real network environment to justify our motivation that proxies may be used as IEEE 802.11 relays to reduce file downloading latency in Section 7.4.1. We also evaluate the performance of cooperative parallel file downloading by various simulation scenarios. The simulation models, metrics, and methodology are shown in Section 7.4.2. The simulation results, which investigate a wide range of parameters, such as the node density, mobility and degree of parallelism, will be presented. We start with the simplest scenario of a single client, downloading a fixed length file in Section 7.4.3, and then move to scenarios with multiple file downloads in Section 7.4.4.

7.4.1 Experiments

The experimental testbed consists of three Linux-based laptops (C, $P1$ and $P2$), an IEEE 802.11 router/access point, and a Linux server. We use the IEEE 802.11 wireless LAN to simulate the 3G cellular network with the access point acting as the base station. Each laptop is equipped with two IEEE 802.11b wireless cards. One of them operates under infrastructure mode, which connects to the access point via channel 1. The other operates under ad-hoc mode, which connects to the other laptops via channel 11. Since channel 1 and channel 11 use non-overlapping frequencies, the traffic will not have interference between the IEEE 802.11b WLAN and the ad hoc wireless network. The Linux server connects to the router/access point via Ethernet as an Internet file server. C is a client to download a file from the Linux server,

and $P1$ and $P2$ are two proxies for C. The downlink bandwidth of the laptops' WLAN interfaces are limited by Linux *iproute2* traffic control tools in order to emulate different downlink bandwidths in cellular data networks. We conduct experiments over different combinations of downlink speed in laptops C, $P1$ and $P2$.

A file server is established in the Linux server that can accept requests to download portions of a specific file. A TCP server is set up in each proxy to accept proxy requests from the client. File downloading requests are sent by the client to a proxy, which in turn sets up another TCP connection to the file server to download the requested part of the file. Packets are forwarded in the proxy between the two TCP connections.

We conducted experiments in three cases to download a 2 Mbyte file: without a proxy, with one proxy, and with two proxies. In the first case, the file is directly downloaded by C via its WLAN interface. For the other cases, the file is split into portions with 20 Kbytes each and downloaded in parallel by C's WLAN interface and the ad hoc interface. When downloading via the ad hoc interface, the data streams are routed from the access point either to the proxy $P1$ or the proxy $P2$ in the WLAN, which further forwards to the client via their ad hoc interfaces. For each downlink bandwidth combination, we conducted the experiment three times and computed the average downloading latency. The downlink bandwidth of C, $P1$, $P2$ and the corresponding downloading latency are recorded in table 7.1. The gain field shows the throughput gain in comparison to the case with no proxy.

The results of the experiments show that the mobile devices can fully utilize the nearby idle cellular links by using cooperative parallel file downloading. The client can

Table 7.1: Download Latency and Throughput Gain w/o Parallel Downloading

Client	Proxy1	Proxy2	Latency	Gain
256K	—	—	65.70s	—
256K	256K	—	35.22s	87%
256K	256K	256K	24.51s	168%
512K	—	—	33.02s	—
512K	512K	—	17.60s	87%
512K	512K	512K	12.22s	169%
512K	256K	256K	18.83s	75%
1M	—	—	16.43s	—
1M	1M	—	9.16s	79%
1M	1M	1M	6.66s	147%
1M	512K	512K	10.12s	62%
1M	512K	256K	11.24s	46%

achieve up to 169% throughput gain by utilizing two proxies. Because of increasing collisions on the ad hoc network, the gain decreases slightly as the downlink speed increases.

7.4.2 Simulation Model

We implement the cooperative parallel file downloading scheme in the *ns*2 network simulator. N mobile nodes are randomly placed in the area of $886m \times 886m$. We vary the value of N to show the effect of different node densities. Each node has a cellular interface and an IEEE 802.11b interface. The cellular interface is used to communicate with the base station, which in turn connects to the Internet. It uses an one-hop routing scheme from the base station to the mobile node. Different cellular interfaces use different radio channels and do not interfere with each other. The base station is located in the center of the simulation area and the mobile nodes can connect to the base station from any location in the simulation area.

We use the IEEE 802.11b implementation in ns-2 version 2.1b9, where 11 Mbps data rate is supported at the 115 meter range. The radio propagation model for IEEE 802.11b uses the two-ray ground reflection model. Node mobility is set according to the random waypoint model. Each node moves toward a random destination within the field at a random speed within the range of 1 to the maximum speed parameter.[1] After reaching the destination, the node pauses for 4 seconds and then starts moving again. We vary the maximum speed to investigate the impact of node mobility.

We use four metrics to evaluate the performance of the cooperative parallel downloading scheme and the effect of various simulation parameters: the download latency of a file; the minimum, maximum and average performance gain in comparison to a non-proxy download; the average number of proxies used for each file download; and the routing overhead in the proxy discovery protocol, which is represented as the number of routing messages transmitted in the IEEE 802.11 network.

7.4.3 Single File Downloading Scenario

In this section, we start with a simple scenario of a single client with one file to download. We assume that all mobile nodes have the same cellular link bandwidth. We limit the maximum number of proxy request hops to be 3. We vary the size of the download portion from 5K to 950K, the maximum moving speed of the mobile nodes from 0m/s, 5m/s to 15m/s, the cellular link bandwidth from 256 Kbps, 512 Kbps, 1 Mbps to 2 Mbps, the file size from 100 KBytes, 500 Kbytes, 1 Mbytes, 2 Mbytes

[1]The minimum speed is not set to zero according to the results shown in [173].

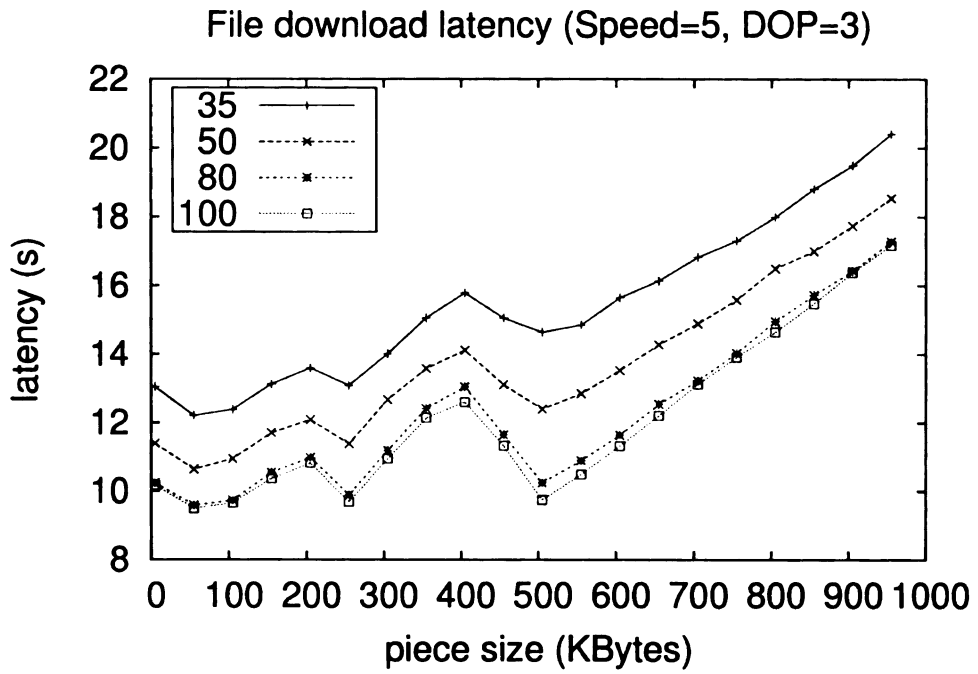


Figure 7.4: Performance with different node density

to 4 Mbytes, and experiment with different client densities by placing 35, 50, 80 and 100 clients, including the destination client, in the simulation area.

Figure 7.4 shows the impact of the size of the file portion downloaded under different node density. The results indicate that a small size will increase the downloading latency due to the high overhead of routing and limited pipelining. On the other hand, a large size does not provide enough modularity for parallel downloading, and also results in performance degradation. It is obvious that the optimum size for this simulation scenario is between 5K to 50K. As expected, mobile nodes more easily find proxies when there is a higher node density, which are also shown in the results of Figure 7.4.

Figure 7.5 shows the performance gain with different degree of parallelism settings under different cellular downlink bandwidth. The results indicate that the average

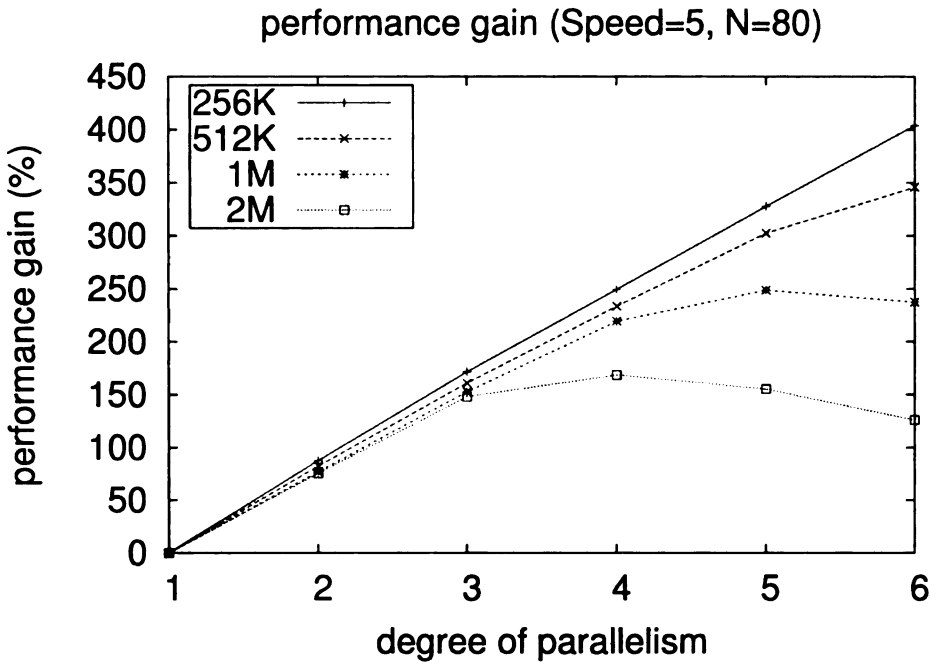


Figure 7.5: Performance with different cellular link bandwidth

number of proxies used increases linearly with the increasing of degree of parallelism without affection by the cellular downlink bandwidth. However, increasing the degree of parallelism using a slower cellular downlink results in higher performance improvement. The reason is the IEEE 802.11-based MANET more easily saturates with higher cellular link capacity with the same number of proxies, which is also justified by the experiments. A higher degree of parallelism may be chosen by mobile nodes with a slower cellular downlink capacity. Figure 7.6 reveals the relation between different downloading file sizes and the performance gain. The results show that the larger files benefit more from parallel downloading.

The effect of request pipelining is shown in figure 7.7. The simulation result shows the maximum, minimum and average performance gain for downloading a 4M file with or without pipelining proxy discovery and downloading requests for the case of 50

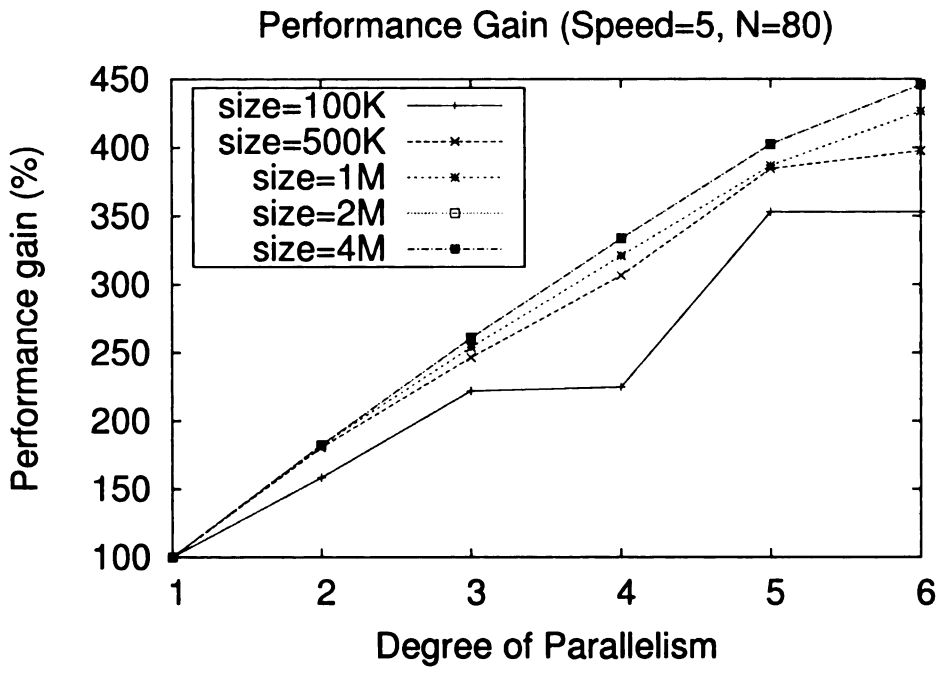


Figure 7.6: Performance with different file size

mobile nodes density and 5m/s maximum moving speed. Pipelining may significantly increase the system performance. For example, with a degree of parallelism of 5, the average performance increases 127% with pipelining.

7.4.4 Multiple File Downloading

The next scenario we investigate has a cellular downlink that is a normally distributed random variable with average 600 Kbps and variance 200 Kbps. The link capacity is also limited by a minimum speed of 38.6 Kbps and maximum of 2 Mbps to simulate different cellular downlink bandwidth in the real network. We select the node density to be 50, and randomly select 5 or 10 mobile nodes to start downloading files of size 1 Mbytes simultaneously to simulate low and high traffic load. To make the case simple, we assume mobile nodes are self-interested. A mobile nodes will accept proxy

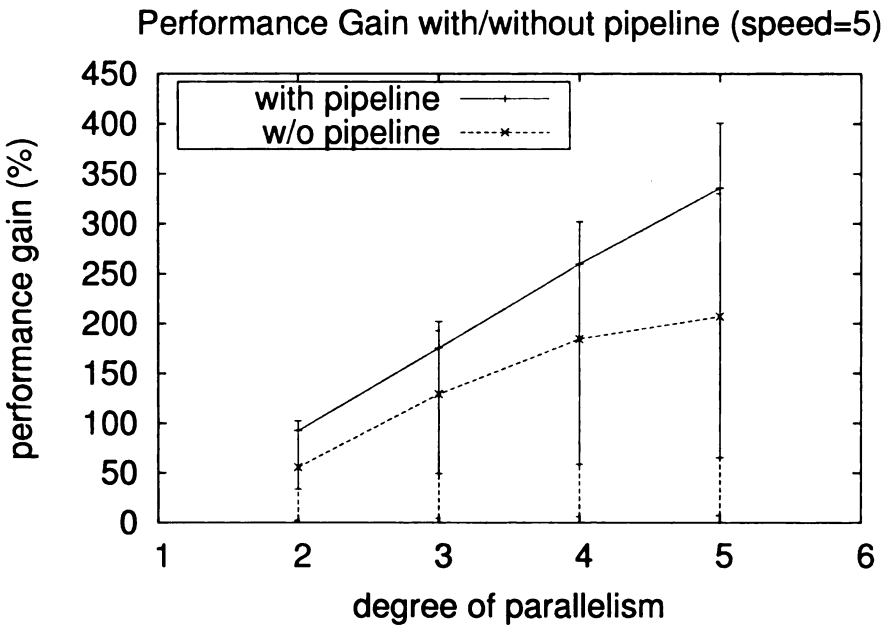


Figure 7.7: Effect of Pipeline Technique

requests from other mobile nodes when its cellular link is idle and it always accept forwarder requests. In order to reduce the contention in the IEEE 802.11 network, one mobile node can only accept one proxy request at a time.

Figure 7.8 shows the performance gain with 5 simultaneous file downloads under different degree of parallelism and different mobility. The error bars represent the maximum and minimum performance gain. The result shows that the performance increases as we increase the degree of parallelism. With higher node mobility, the IEEE 802.11 connection from the proxy to the client is more likely to break, which slightly decreases the performance gain. Figure 7.9 shows the similar result with 10 simultaneous file downloads. With higher traffic load, increasing the degree of parallelism has no effect on increasing the system performance. In both figures, the maximum performance gain increases linearly with DOP, which indicates some nodes

can fully utilize the parallelism. The minimum performance gain is always positive, which means the performance is at least same as the non-proxy scheme.

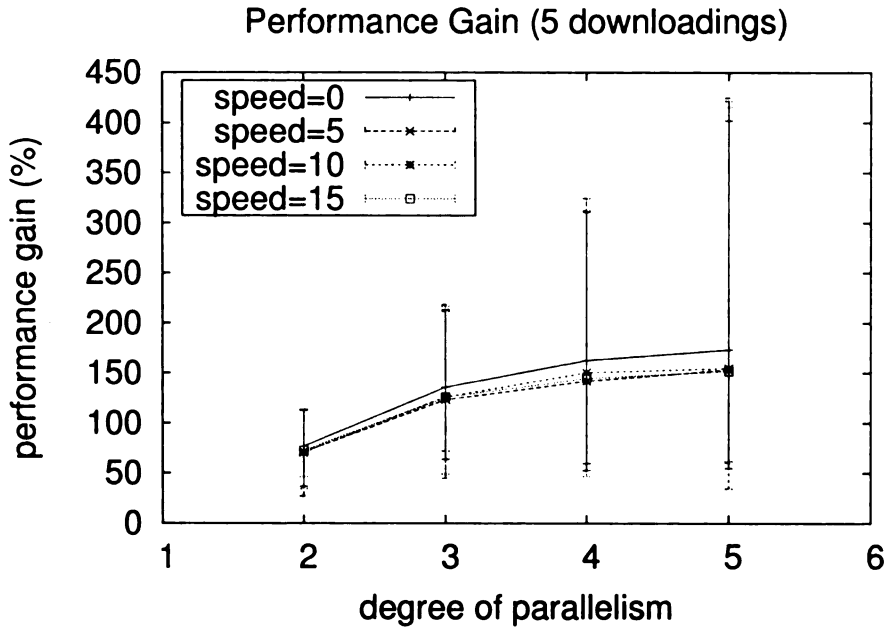


Figure 7.8: Performance Gain by 5 Downloads

7.5 Summary

We propose a novel cooperative parallel file downloading scheme to utilize multiple paths in the cellular data network with the help of the IEEE 802.11-based mobile ad hoc network. A simple and efficient proxy discovery protocol is used to find new proxies. A new trust model is presented to improve the incentive to cooperate. The experimental and simulation results show that cooperative parallel file downloading can significantly reduce file downloading latency without changing the network architecture.

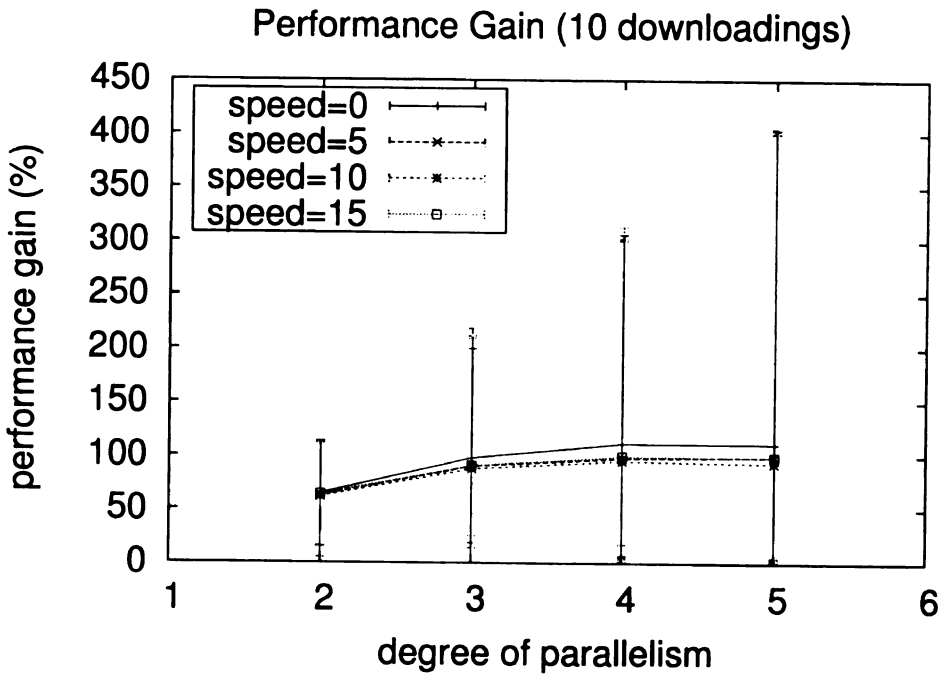


Figure 7.9: Performance Gain by 10 Downloads

Chapter 8

Promoting Fairness Among Strangers in MANET

8.1 Introduction

In pervasive computing environments, many applications require cooperation among a group of users to gain benefit for the whole group. However, when the users do not belong to the same authority, they lack the motivation to cooperate when selfish behavior may provide more benefit. For example, mobile ad hoc networks work properly only if the participating nodes cooperate in routing and forwarding. A selfish node may be unwilling to spend its precious resources, such as battery life, CPU cycles, or available network bandwidth, to forward packets for others, although it expects others to serve on its behalf.

The situation becomes more difficult when cooperation is needed among strangers in temporarily formed mobile ad hoc networks. In this case, the strangers have

neither initial trust information nor a long term relationship with each other. Why should one stranger provide service to others? How should one assure that others will return service in the future? These are all questions that make strangers reluctant to cooperate even if it will bring benefit to all of them. If the users are not charged money for their access to the shared resource (charging may be difficult in pervasive computing environments), it appears rational for them to use the service provided by others without contributing their own. However, if all in the group *free ride* on the service of others, the whole system will not work[1]. It is what is called "the tragedy of the digital commons" [102].

In this chapter, we present a distributed trust model and a credit system to promote the incentive of cooperation among a group of strangers in the highly mobile environment. In this model, peers evaluate the trust they have for other peers. A peer that obeys the established protocols trusts other peers who are perceived to follow the protocols. Trustworthy peers reap the benefit of service provided by others. This trust model and credit system promote fairness and remove "free riders" in the CHUM (Cooperating ad Hoc environment to sUpport Messaging) project. The same idea may also be extended to other pervasive computing applications.

A description of the approach for distributed evaluation of trust and building the credit system is given in section 8.2. Section 8.3 provides the mechanism to promote cooperation and improve fairness among group members. An evaluation is given in section 8.4 and conclusions are given in section 8.5.

[1]Free riders are those users that use a service but do not contribute. Researchers at Xerox PARC determined that almost 70% of Gnutella users do not share files, and 50% of all responses are returned by the top 1% of sharing hosts [101].

8.2 Developing Trust Knowledge

We develop trust knowledge to support a distributed scheme for proxy scheduling. Peers might only contribute to CHUM if they believe others will provide service for them (share the costs) and be prevented from intruding upon their privacy. While privacy should be maintained with appropriate encryption schemes, fairness will be promoted using the credit system that is built on top of the trust model. Good behavior should be rewarded, and cheating should be punished. The trust knowledge is further used in the credit system to ensure that each peer receives the benefit from the group in proportion to the service it provides.

We assume that peers have no initial knowledge of trustworthiness of each other. They are considered to be strangers when they first meet. Although there may exist some solutions that use centralized trusted authorities (TAs) to develop knowledge of trust of two mutually mistrusting peers, these TAs are either not available or too costly to access in mobile environments. Since the members within a cooperating group may change frequently, it is difficult for a mobile node to be trusted by all the others and acts as a TA. On the other hand, a highly secured protocol is needed for the communication between the mobile node and a centralized TA located in the Internet, which results in a high communication overhead and power consumption for a mobile device. Here, we consider the cases that peers are all equal and there is no expert upon whom to rely. The only manner for trust knowledge to evolve is that peers exchange messages among themselves to describe the behavior of others.

8.2.1 Trust Among Peers

Let $N = \{p_1, ..., p_n\}$ be the set of peers within a cooperating group. A trust value $TRUST_{i,j}$ within the range [-1,+1] is defined between two different peers p_i and p_j. It represents the trust level p_i has of p_j. $TRUST_{i,j}$ and $TRUST_{j,i}$ are not necessarily equal. The value -1 represents "total mistrust", the value +1 represents "total trust", and the value 0 represents "neutral" or "no knowledge." We chose a similar range as used in [129]. Instead of viewing +1 as blind trust in [129], we consider it as a high trust level. The continuous range reflects a trust continuum and allows quantitative trust evaluation. The positive and negative values easily represent positive and negative trust without misunderstanding.

Trust will be derived from two different types of trust information. One is called **perceived trust**, which is the information obtained from a peer's direct interaction with another peer, such as a successful transaction, or a perceived cheating regarding a transaction. Perceived trust is a value within the range [-1,+1]. We represent the perceived trust p_i has of p_j by $PT_{i,j}$. The other type of trust information is called **recommended trust**. This trust value is summarized from other peers' opinions and is obtained from the trust update messages obtained from other peers. Recommended trust may or may not be true since some peers may provide false information. It is also within the range [-1,+1]. The recommended trust p_i has of p_j is represented by $RT_{i,j}$.

The trust p_i has of p_j, $TRUST_{i,j}$, can be derived from the perceived trust of p_j, $PT_{i,j}$, and recommended trust of p_j, $RT_{i,j}$. Parameter δ is used to place a weight on the

perceived trust value. If we are more willing to trust our own "eyes", a larger δ will be used. On the other hand, if we are more willing to accept the "opinion" from others, a smaller δ would be used.

$$TRUST_{i,j} = PT_{i,j} \times \delta + RT_{i,j} \times (1 - \delta)$$

In our trust model, trust is a nontransitive relationship. For example, if A trusts B with level 0.7 and B trusts C with level 0.3, the amount of trust that A has of C is determined by the direct interaction between A and C with some influence from the trust A has of B and B has of C. The detailed algorithm is presented in the section 8.2.2.

8.2.2 Trust Evaluation and Update

Each peer determines its trust value of others based on the knowledge it acquired when other peers perform service or report the trust they perceive of one another. The *Trust Information Exchange and Query Protocol* (TIEQP) is used to exchange and request trust information among peers, and the *Trust Update Algorithm* is used to modify the perceived and recommended trust at the peers.

After an evaluation of a new perceived trust value of another peer, a peer broadcasts the new value via the TIEQP. The trust information message contains the identification of the sending and receiving peer, the identification of the peer for whom the recommendation is specified, and the recommended trust value. By collecting trust information from others, the recommended trust for a particular peer may be eval-

uated. Furthermore, a new member may also request trust information from other peers via the TIEQP.

Trust Update Algorithm

The recommended trust that a peer has of others is based on trust update information obtained from others. When a peer receives new information from other peers regarding their perceived trust of others, the peer first evaluates how much the peer trusts the information provider. If the provider is trusted, then the provided trust information has greater effect on the calculated updated trust perception of the peers for whom reported information is received. Suppose p_i receives a message from p_j using TIEQP regarding $p_j's$ perceived trust of another peer p_k in the group. The new recommended trust that p_i will have for p_k, $RT'_{i,k}$, will be a function composed of the old recommended trust value of p_k and the new perceived trust that p_j has of p_k multiplied by the trust that p_i has for p_j, that is:

If $TRUST_{i,j} \geq 0$, then $RT'_{i,k} = RT_{i,k} \times \alpha + TRUST_{i,j} \times PT_{j,k} \times (1 - \alpha)$. Otherwise $RT'_{i,k} = RT_{i,k}$.

This means that if p_j is not trusted by p_i, then p_j does not affect p_i's view of others. The parameter α, where $0 \leq \alpha \leq 1$, smoothes the value of the current recommended trust with the past recommended trust.

Furthermore, perceived trust is updated based on the amount of service provided in one *service provision transaction* after this transaction has finished. Different kinds of services may have different definitions for the service provision transaction and the amount of service. In CHUM, the service provision transaction is defined as the

time period from the beginning of a proxy service to the end of a proxy service. The amount of service of one transaction is defined as the length of the service provision transaction.

Let's suppose one service provision transaction $t_{j,i}$ (in this notation, the first index is the service provider while the second is service receiver) has finished between p_i and p_j, while p_j is the provider and p_i is the receiver. Furthermore, let's say that $s^i_{t_{j,i}}$ is the amount of service given to peer p_i in this transaction.

A new perceived trust value $PT'_{i,j}$ will be computed in the service receiver p_i for the transaction $t_{j,i}$. This new perceived trust value depends on service provider p_j's honesty in this transaction. If provider p_j is honest, the perceived trust of p_j will increase in peer p_i since p_i receives some service from p_j. Otherwise, p_i will set $PT'_{i,j} = -1$. In the example of CHUM described in the next section, the honesty of a peer may be determined by periodically contacting a message server through the proxy. If contact to the server can be made, it is assumed the proxy is honest. Otherwise it is assumed that the proxy is dishonest.[2] The updating algorithm is:

If p_j is honest in transaction $t_{j,i}$: $PT'_{i,j} = min\{PT_{i,j} + h_i(s^i_{t_{j,i}}), 1\}$. Otherwise: $PT'_{i,j} = -1$.

The function h_i is a satisfactory function defined in peer p_i, expressing the perceived trust increase in p_i of the other peer for one successful transaction. The satisfactory

[2]If the IM server is down, then the peer could contact other well known Internet sites to determine if the proxy is honest. If the IM server is truly down, the peer would likely want to leave the group.

function may vary in different peers. The parameter β, where $0 \leq \beta \leq 1$, smoothes the value of current perceived trust with the past value.

$$PT''_{i,j} = PT_{i,j} \times \beta + PT'_{i,j} \times (1 - \beta)$$

The same update is performed in peer p_j if p_j also receives some service from p_i during this transaction period.

8.2.3 The System of Credits

A system of credits will be built on top of trust to promote cooperation and achieve fairness among peers. This credit system has some features resembling a credit card system. It enables one peer to enjoy service now and "pay" later.

Peer p_i assigns a *credit limit* $CL_{i,j}$ for all other peer p_j in the group. This credit limit represents the maximum amount of service that p_j may "borrow" from p_i. In other words, p_j may enjoy the service of p_i at most for $CL_{i,j}$ without providing service for p_i. In order to reward good behavior, a larger credit limit will be assigned to "good" peers (peers that have a larger trust value). The $CL_{i,j}$ may be formalized as a function of $TRUST_{i,j}$: $CL_{i,j} = g_i(TRUST_{i,j})$. Function g_i is the credit limit updating function used in peer p_i. It should be monolithic increasing on $TRUST_{i,j}$ and be positive for non-negative trust.

Each peer also maintains a table that records information and credit limits for all other peers in the group. The information in the table may be obtained locally. The protocol TIEQP is needed only for updating the recommended trust. This table has as many entries as there are peers in the group. For a given peer, the entries include

trust values, a credit limit assignment and a transaction history. A more detailed description of the table is given below. Within peer p_i's entry for peer p_j, it includes:

- PID_j: unique ID for p_j.
- $PT_{i,j}$ and $RT_{i,j}$: perceived and recommended trust values for p_j. They are maintained by TIEQP and update algorithm of section 8.2.2.
- $TRUST_{i,j}$: current trust value for p_j, which is derived from $PT_{i,j}$ and $RT_{i,j}$.
- $CL_{i,j}$: credit limit that peer p_i has assigned for peer p_j. It is computed by the function g_i on the current trust value $TRUST_{i,j}$.
- $SR_{i,j}$ and $SR_{j,i}$: Time length of proxy service that p_i has provided for p_j and p_j has provided for p_i.

The available credit $AC_{i,j}$ for peer p_j in p_i is the amount of service that p_j may receive from p_i without providing service to p_i. It is computed as:

$$AC_{i,j} = CL_{i,j} - SR_{i,j} + SR_{j,i}$$

Peer p_i refuses to continue providing further service for p_j if the available credit $AC_{i,j} \leq 0$. For p_j, the only way to make $AC_{i,j}$ positive is to provide service for p_i, which increases the value of $SR_{j,i}$. Another choice for p_j is to ask for service from another peer other than p_i. However, after p_j has consumed the credit limit from all other peers in the group, and if it still wants the service, it must provide the service itself. The trust knowledge together with the credit system provides an incentive of cooperation for peers. The longer they honestly cooperate in the system, the higher

Table 8.1: Notation of Trust Model

Symbol	Description
$TRUST_{i,j}$	Trust level between peer p_i and p_j
$PT_{i,j}$	Perceived trust p_i has of p_j
$RT_{i,j}$	Recommended trust p_i has of p_j
δ	The weight placed on the perceived trust value
α	Smooth the value of current and past recommend trust
β	Smooth the value of current and past perceived trust
h_i	The satisfactory function in p_i
$CL_{i,j}$	Credit limit p_i assigns for p_j
g_i	The credit limit updating function used in p_i
$SR_{i,j}$	Length of proxy service p_i has provided for p_j
$AC_{i,j}$	Available credit p_j in p_i

trust others have for them. Thus the more free and continuous services they can receive from others. To make the notation accessible, we present it in table 8.1.

8.3 Promoting Fairness in CHUM

The proxy provides the message notification service for all the clients in the group, which requires it to maintain a continuous cellular channel to the Internet. A peer may send a service request to the current proxy. When accepted, the peer becomes a client and receives service from the proxy. The proxy decides to accept or reject the initial request from a peer, or stop providing service for a current client. It may also quit being a proxy and request service from others. A peer or client may announce itself to be a proxy and begin to provide service for others. There are several questions need to be answered before we lead to the peer's strategy:

1. When will a proxy accept the service request from a peer?

2. When will a proxy stop providing service for a current client?

3. When will a proxy quit being the proxy and request service from others?

4. What will a client do when it is rejected by the current proxy?

5. When will a client or peer become a proxy?

To answer all these questions, we need to examine the peer's behavior and the features of the trust model and the credit system. In the following section, we will present an analysis of the peer's behavior from the view of game theory. A basic introduction to game theory may be found in [174].

8.3.1 Game Theoretic Analysis

The mobile nodes (peers) are the players in this game. We assume that a peer's action is economically rational, which is a basic assumption of game theory. Each peer knows that other peers are rational. Peers choose the strategy that maximizes their profit. The trust and credit system described in section 8.2.2 and section 8.2.3 have been implemented for all peers. All of the peers in the group want Internet access continuously.

Suppose that the unit cost for the Internet connection is C and the connection is charged by the time length. The total cost for t seconds Internet connection can be represented as $C \times t$. This is the value of profit when a client receives free service from a proxy. The proxy requires an additional cost to provide service for the clients

(such as additional power consumption). This additional cost will be proportional to the amount of services provided by a small constant factor Δ. On the other hand, when a proxy provides service for a client, it expects some reward from the client in the future. The reward is proportional to the service it provided by a factor P_{reward}, $0 \leq P_{reward} \leq 1$. The value of P_{reward} depends on the trust of the client and the available credit of the client. We can imagine that a trusted client with a large available credit is more likely to stay in the system.

To illustrate the game theoretic model, we only present the analysis of a 2-peer group (peer x and peer y). The N-peer model is similar. During the time period t_0, suppose x acts as a proxy for length t_p and acts as a client for length t_c, the net profit for x can be represented as the free service it received from y when it acts as a client ($C \times t_c$) plus the rewards it expects from y when it acts as a proxy ($C \times t_p \times P_{reward}^{x \to y}$) minus the additional cost required for being a proxy ($C \times t_p \times \Delta$). Here, t_0, t_p and t_c follow the constraint: $t_0 \geq t_p + t_c$. If we define $P_{reward}^{x \to y} - \Delta$ as θ_x and $P_{reward}^{y \to x} - \Delta$ as θ_y, the utility functions for x and y can be represented as:

$$U_x(t_0, t_p, t_c) = C \times (t_c + t_p \times \theta_x)$$

$$U_y(t_0, t_p, t_c) = C \times (t_p + t_c \times \theta_y)$$

Here, since Δ is a constant value, θ_x represents the expectation of reward x has of y and θ_y represents the expectation of reward y has of x. The expectation of reward is closely related to the available credit. The available credit that y has on x is the maximum amount of reward x expecting from y based on the credit limit (which is determined by the current trust level x has of y), the amount of service x provided for

y and the amount of service y paid back for x. Therefore, we may assume a positive reward expectation from y ($\theta_x > 0$) when the available credit $AC_{x,y} > 0$ and a zero reward expectation from y when the available credit $AC_{x,y} = 0$.

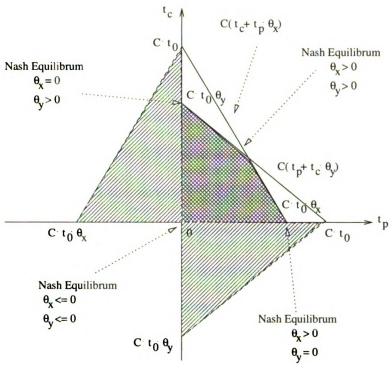


Figure 8.1: Graph to find equilibrium

We graph the utility functions for x and y in figure 8.1. The Nash equilibrium of the game can be identified in the graph by maximizing the profit for both x and y. We present four different cases of the Nash equilibrium based on the value of θ_x and θ_y. For $\theta_x \leq 0$ and $\theta_y \leq 0$, we get $t_p = t_c = 0$, which results in zero profit for both x and y. It means if the peers view each other as selfish freeriders in the beginning, they tend to stay outside the system. Therefore, we have to assign a positive credit limit for zero trust. For $\theta_x > 0$ and $\theta_y > 0$, we get $t_p = t_c \times \frac{1-\theta_y}{1-\theta_x}$ and positive profits, which means that the peers gain profits by cooperating in CHUM. When $\theta_x = \theta_y$,

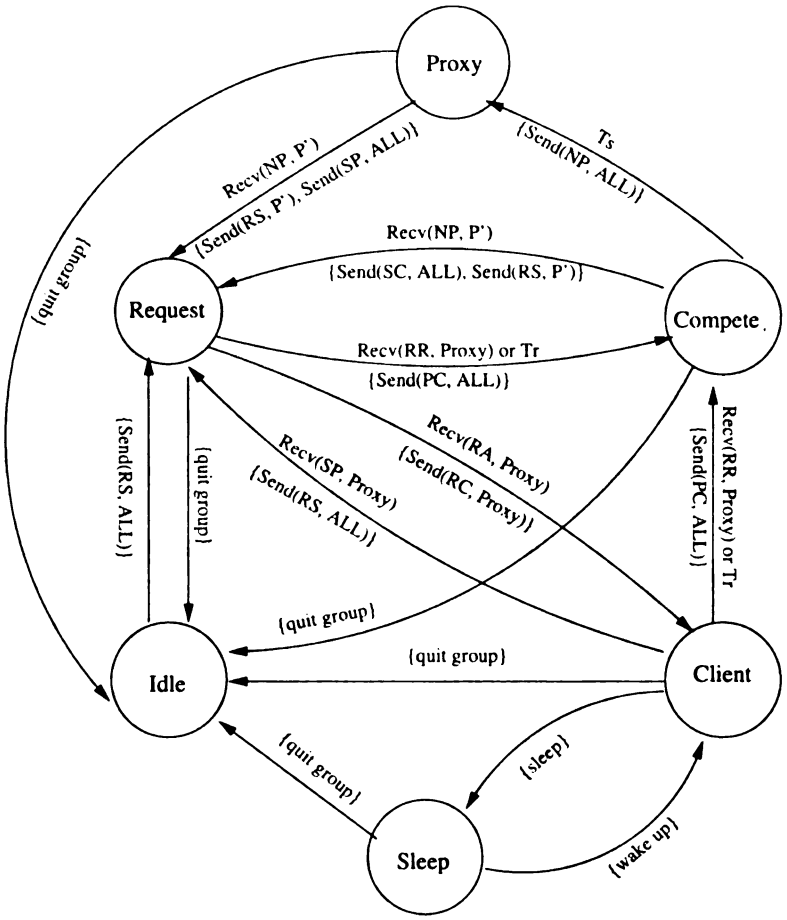
$t_p = t_c = t_0/2$. This result is reasonable since we expect x and y to share the burden of proxy when they have the same reward expectation for each other. For $\theta_x = 0$ and $\theta_y > 0$, we get $t_p = 0$ and $t_c = C \times t_0 \times \theta_y$. It means that if the reward expectation x has of y is equal to zero, x will not provide any proxy service for y. In other words, x will stop or reject service for y when $\theta_x = 0$ or the available credit $AC_{x,y} = 0$.

To summarize, x maintains a trust value for y, which is used to assign a credit limit $CL_{x,y}$ for y. Based on the credit limit and the service x provided for y and the service x received from y, x can compute the available credit y on x, which in turn is used to determine the acceptance of the service request from y. Thus the algorithms of the proxy and the client can be explained as follows. The proxy stops or rejects service for another peer if the available credit is negative. It will quit being a proxy when another proxy is available, which can be explained in the graph that being a client can provide more profit than being a proxy. When rejected, the client will announce itself as a proxy and begin to provide service for those with positive available credit. When $t_c = 0$ (since the client is rejected by the proxy), increasing the value of t_p for a positive θ will increase the total profit.

8.3.2 A Peer's Strategy

Based on the game theoretical analysis on the previous section, we define the state machine for CHUM in figure 8.2 to illustrate the peer strategy. A peer may be in one of the following six states: *IDLE* (outside a CHUM group), *REQUEST* (wait for response to a service request from the current proxy), *CLIENT* (receive service from

the current proxy), *SLEEP* (power saving mode), *PROXY* (provide service for other peers), and *COMPETE* (compete with other peers to become the proxy).



RS: Service Request — RC: Request Confirm
RR: Request Rejected — PC: Proxy Competing
RA: Request Accepted — SC: Cancel proxy competing
NP: New Proxy avaiable — SP: Stop providing proxy service

Tr: Timer for waiting proxy to respond
Ts: Timer for listening for competing message
It is a random number between 0 to Max Ts

Figure 8.2: State machine for a peer in the group

A proxy competition protocol is defined to resolve the collision of several peers competing to be a proxy at the same time. For the protocol, first broadcast a message that this peer wishes to become a new proxy. Then, listen for a random period. If

there is no other proxy available during this period, announce itself as the new proxy, and start providing service for others. If another peer announces its availability as a proxy during the listening period, broadcast the cancellation of the competition and ask the new proxy for service.

An IDLE peer may join the group by broadcasting a *service request* (RS) message to all other peers in the group, and then enter the REQUEST state. An IDLE peer learns the identity of the peers in group via group formation protocols that we do not address in this paper.

For the peer in the REQUEST state, if it receives a *request rejected* (RR) message in response to the *RS* message or does not receive a response from the current proxy by the T_r timeout, this peer begins the proxy competition protocol by sending a *proxy competing* (PC) message while entering the COMPETE state. If a *request accepted* (RA) message is received in response to the *RS* message, this peer enters the CLIENT state after replying with a *request confirm* (RC) message and begins enjoying the proxy service from the current proxy. A peer stops providing proxy service when it receives a *new proxy available* (NP) message. Then it sends a *RS* message to request service from the new proxy, and enters the REQUEST state.

A peer may leave the CLIENT state and begin the proxy competition protocol by receiving the *RR* message from the current proxy (it means this peer has consumed its available credits on the current proxy and the proxy refuses to serve further) or not receiving a response from the current proxy within T_p time (it means the proxy may have left the group or stopped providing service). A peer may change to a power saving mode while entering the SLEEP state. A sleeping peer may "wake up"

periodically, and return to the CLIENT state to contact the previous proxy. If the previous proxy remains active, this peer will receive the message waiting on the proxy and then return to the SLEEP state. Otherwise, this peer sends out a RS message to request service from the new proxy and enters the REQUEST state.

The competition for the proxy is successful if the competing peer does not find any other proxy available within the random T_s timer. When a peer in the COMPETE state receives a NP message, it leaves the COMPETE state and requests service from the new proxy by sending a RS message. Otherwise, it announces itself as the new proxy by broadcasting the NP message and enters the PROXY state.

8.3.3 Discussion

In this section, we discuss some potential attacks on the approach and the possible solutions. We also compare our approach to another possible approach to deal with the "free riding" problem.

Comparison to simple round robin: A simple round robin scheme may be considered for the "free riding" problem in CHUM. In a round robin scheme, a token is assigned to the group, which is organized as a ring. The peer holding the token acts as the proxy for a predetermined length of time. The token is passed among all peers. The peers with bad behavior will be removed from the group. Nevertheless, there are two major restrictions preventing this scheme to work well in CHUM: first, token management causes high overhead because of the highly dynamic and unreliable characteristics of the group; second, it is difficult for all in the group to agree when a

bad behaving member should be removed from the group. In section 8.4, simulation results show that our approach works well even when compared to a perfect round robin scheme.

Flooding false trust information: A malicious peer may broadcast false information about other peers in order to destroy their reputations. However, only the recommended trust is gathered from other peers, which is only part of derived trust. The peer may also increase the δ value to reduce the effect of the recommended trust. On the other hand, the update of recommended trust depends on the trust of the peer who sends the information. Flooding false information can be detected. The trust of this peer may be set to a negative value to prevent malicious false information affecting other peers' reputations.

Refuse to provide service: A proxy may refuse to provide service for peers that have suitable credits. However, rejecting service for other peers does not benefit the proxy. The proxy will not be able to increase its available credit at other peers, and therefore may not enjoy the service from other peers. A malicious peer may also pretend to be the proxy, and then neglect to provide service to others. It will prevent other peers from becoming the proxy. Again, when detected, the trust value of the malicious peer may be set to a negative value. Its future messages may be ignored and its credit line reduced.

Frequently join and leave: Because the credit limit is always positive for a newly joined peer, this may become a source of an attack. A selfish peer may join one group, using up all its credit limit from other peers, and then leave the group. He may later rejoin the group, and restart the whole process. This kind of attack is

very difficult to prevent without some long-term accountability, which is impossible in CHUM. However, a clever design of the credit increase function g_i may reduce the loss on this attack to an acceptable level.

Proxy recovery: The proxy may stop providing service or leave the group without providing notice to its clients. The missing proxy needs to be detected and the system needs to recover. A lost proxy is detected by asking a proxy periodically to broadcast its existence to all group members (by the *proxy active* (PA) message). If the preset timer (that may equal to twice of the broadcasting period) has expired, a client will notice that the proxy disappeared, and begins the competition process.

Sleeping mode: The power-saving mode creates some problems. First, we need to distinguish between a sleeping peer and one that has disappeared. A timeout is then set up in the proxy for each client. If after a certain amount of time, the proxy does not receive a response from the client, it will assume the client left the group. Second, if the proxy stops providing service when a client sleeps, there is no simple way for the client to determine when a proxy stopped its service. One simple approach approximates by assuming half of the sleep time the proxy provided service. Our performance evaluation section shows that this approximation is useful. Furthermore, a sleeping peer misses recommended trust update information provided by TIEQP. However, the peer may still update its perceived trust, and therefore update its final derived trust value. For those peers that periodically go to the SLEEP state, we may set δ to zero to ignore the recommended trust information from others.

Table 8.2: Parameters for Simulation

Para	Description	Value
α	Weight for updating RT	0.6
β	Weight for updating PT	0.7
δ	Weight for derived trust	0.7
Tr	Timer for REQUEST state	2 s
Ts	Timer for COMPETING state	4 s
CL_{start}	Start credit	60 s
CL_{incr}	Increase credit	300 s
L	Average peer lifetime	120 min
λ	Peer join rate	0.05

Message delay: Message delay may cause a transaction recording synchronization problem. Our performance evaluation shows that message delay does not have much effect on the results.

8.4 Performance Evaluation

We conducted simulations to evaluate our trust model and reputation system in the CHUM project. The simulation is done via a event driven simulator developed for CHUM following the strategy described in the section 8.3.2. Since the routing model in CHUM is limited in one hop [171] and messages are broadcasted to all peers in the group, we do not consider mobility in the simulation. When a peer moves outside the radio range of the proxy, we just consider it as has left the CHUM group. Three different metrics are used in the evaluation:

1. *Fairness (or Unfairness)*: The peer will consider that it receives fair service if and only if its proxy length is proportional to the group size. For example, if

there are 10 peers in one group, a peer will feel satisfied if it spends 10% of its time acting as the proxy providing service and 90% of its time acting as client receiving service from others. For a peer, suppose the average group size during its lifetime in the group is s_g, the length of time that it stays in the group is l_g and the length of time that it acts as proxy is l_p. *Fairness* for this peer will be defined as $fairness = (l_p \times s_g)/l_g$. $Fairness = 1$ means absolute fairness. $Fairness < 1$ means the peer used more service than that which it was entitled. $Fairness > 1$ means that the peer contributed more than its fair share. $Unfairness = |Fairness - 1|$.

2. *Stability* of the system is measured by the average length of each proxy service period. Longer periods mean that the system was more stable with fewer proxy changes.

3. *Overhead*: Peers that stay either in state PROXY or in state CLIENT may receive IM service. The overhead of system is defined by the fraction of time that peers stay in other states.

8.4.1 Simulation Setup

The peers join the group according to a Poisson process with rate λ. The lifetime of peers in group is exponential distributed with mean L. The stability of one simulation is evaluated as the total length of simulation divided by the number of proxy changes. The unfairness of one simulation is measured by averaging the unfairness of all peers generated.

For simplicity, we assume that all of the peers use the same satisfaction function and credit limit increase function. The credit limit increase function g_i used in peer p_i is: $g_i(TRUST_{i,j}) = TRUST_{i,j} \times CL_{incr} + CL_{start}$. Here CL_{start} is said to be the *start credit.* It is the parameter used to represent the credit limit peer p_i assigned for p_j when p_i has zero trust of p_j (p_j is a new comer). And CL_{incr} is said to be the *increase credit.* It is the parameter used to represent the credit limit increase for trust increase. The satisfaction function h_i used in peer p_i is: $h_i(s^i_{t_{j,i}}) = s^i_{t_{j,i}}/(CL_{incr}+CL_{start})$. Since $CL_{incr} + CL_{start}$ is the maximum length of one service provision transaction, the value of function h_i is always within the range of [0, 1].

With regard to fairness and stability, we measured the stability of the peer strategy, and the influence of the start credit CL_{start}, increase credit CL_{incr}, and the average group size $L \times \lambda$. Table 8.2 lists the parameter settings for the simulation.

We also compare our trust model with a Perfect Round Robin (PRR) scheme. In this scheme, peers join into the group according to a Poisson process with rate λ and stay in the group with mean L. Peers in one group form a ring and take turns to become the proxy. The proxy rotates every X seconds to the next one in the ring. Here, we do not consider the overhead of the maintaining the ring. *PRR X* is used to represent perfect round robin in which the proxy rotates every X seconds, where X = 60, 120 or 240 seconds.

We first assume all the peers honestly follow the protocols, and present the simulation results on Section 8.4.2. Then some unpredictable behavior of the peers, such as leaving group without notification, message delay, and influence of sleeping client will be simulated in Section 8.4.3 to show the robustness of the system. In Section 8.4.4,

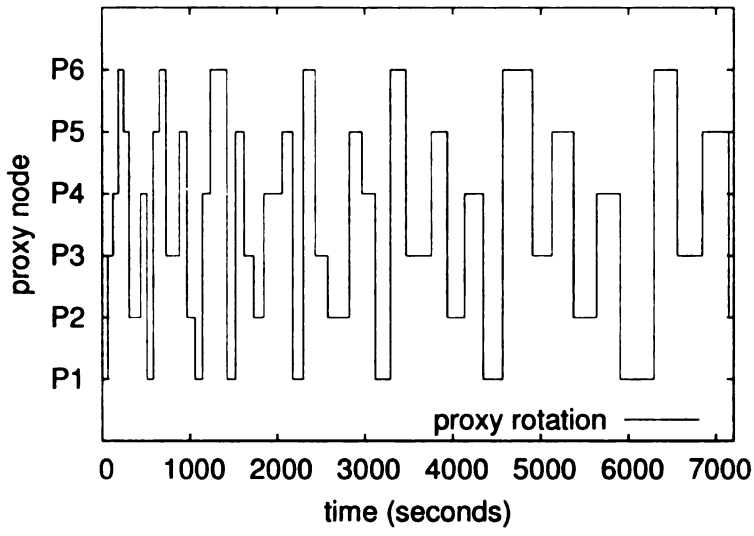


Figure 8.3: Proxy rotation in one CHUM group

we simulate two different kinds of attacks on the system: free-riding and flooding false information.

8.4.2 Simulation Results of Honest Peers

We first run simulation for a single group with six static group members (P_1, P_2, P_3, P_4, P_5 and P_6), which stay in the group for 120 minutes. Figure 8.3 records the proxy rotation of the group over time. The result shows that the responsibility of proxy distributes fairly over six peers. Over time, the peers build trust for each other, which in turn extends the length of each proxy period. Figure 8.4 illustrates the state transition in peer P_1. P_1 spends most of its time either in the $PROXY$ state or the $CLIENT$ state. The system is stable and has a very low overhead.

Figure 8.5 reveals the relationship between the peer lifetime and fairness based on the fairness of 30,000 peers with an average group size of 6. The simulation runs for one

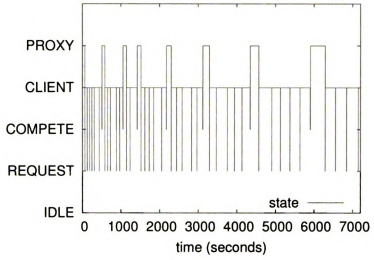


Figure 8.4: State transition in one CHUM group

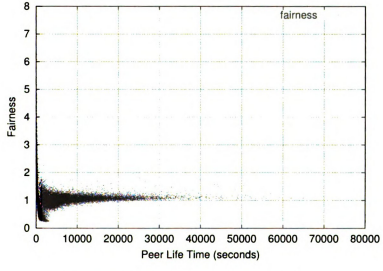
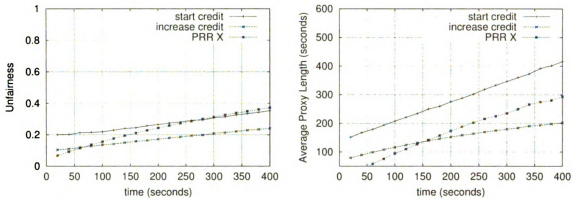


Figure 8.5: Fairness vs. Peer lifetime



(a) Unfairness vs. Credit/Rotate Cycle

(b) Stability vs. Credit/Rotate Cycle

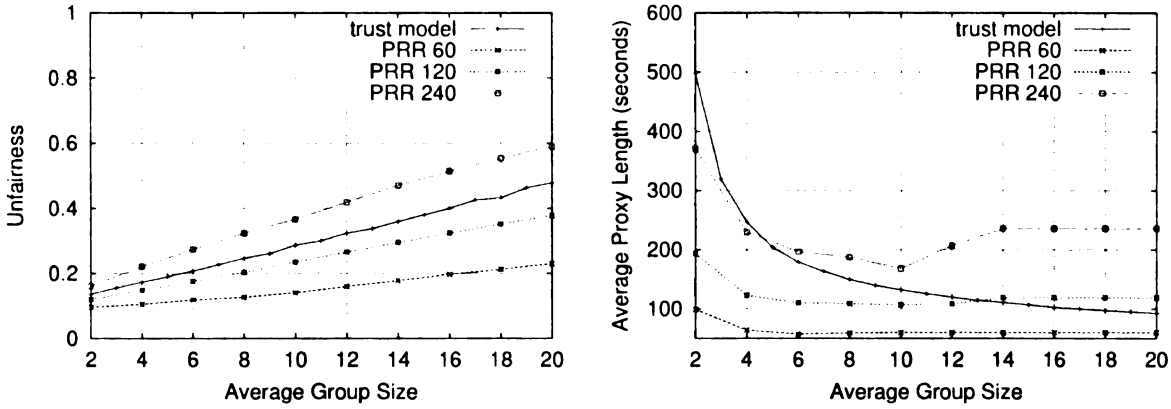
Figure 8.6: Effect of Credit/Rotate Cycle

group with dynamic members. Peers continuously join the group, stay for some time and then leave the group. One point in the figure represents the fairness of one peer, which is recorded when it leaves the group. The fairness of 95% of the peers ranges between 1.66 to 0.34. More than 80% of peers have fairness between 1.29 to 0.71. For peers that stay within the group for more than 30 minutes, 95% have fairness between 1.34 to 0.66, and more than 80% have fairness between 1.19 to 0.81. For a better understanding of the fairness value, let's assume a peer stays 120 minutes in a group with size 6. By the definition, the fairness value of 1.19 for this peer means it spends 23.8 minutes as a proxy for others. However, its fair share in the group will be 20 minutes. In other words, it provides 3.8 minutes additional proxy service during the 120 minute period. This shows that our approach is relatively fair. For peers that remain within the group for an extended time, fairness improves.

Figure 8.6 shows the effect of the start credit and the increase credit on the system performance. For the start credit, the y value of every point represents the fairness

or stability when we fix the increase credit and set the start credit to the value of the x axis. Similarly, for the increase credit, we fix the start credit and change the value of the increase credit. Since the start credit and increase credit determine the proxy rotate cycle in our trust model, we draw the simulation result of $PRR\ X$ in the same figure. The y value of every point in the line $PRR\ X$ represents the fairness or stability when we set the proxy rotate cycle in $PRR\ X$ to the value of x axis. The left figure of figure 5 shows that the average unfairness of peers increases linearly as the start credit and increase credit increase. Nevertheless, the right one shows that the stability of a group also increases linearly as the start credit and increase credit increase. A larger start credit and increase credit mean less frequent proxy changes. By varying the starting and increase credit, we trade off fairness and stability. This figure also shows that by choosing a small start credit and a large increase credit, the trust model may be more fair and stable than a perfect round robin scheme.

Figure 8.4.2 shows the influence of the average group size. The left figure shows that unfairness increases linearly as the group size increases. Larger groups mean greater benefit achieved for an individual peer, since more peers contribute as proxies. The trust model and credit system do not limit the group size in theory. However, as shown in the figure, smaller groups are more fair. On the other hand, as group size increases, the overhead of communication also increases due to the high probability of packet collision which is not desirable. A group size that reaches 10 may be most suitable for CHUM. This figure also shows that with same group size, if the proxy rotates more frequently, the PRR approach may increase the fairness by a small value. Nevertheless, a smaller proxy rotate cycle also makes the system less stable than our



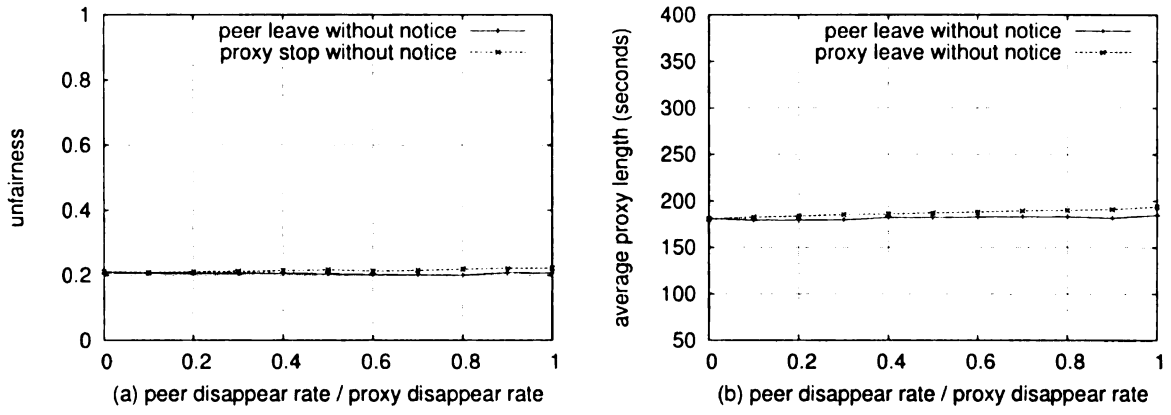
(a) Unfairness vs. Average group size

(b) Stability vs. Average group size

Figure 8.7: Effect of Average Group Size

trust model. In this figure, *PRR* 60 and *PRR* 120 are more fair than our trust model, but less stable. On the other hand, *PRR* 240 is more stable with smaller fairness. The satisfaction level of a service provision is determined by the length of the service in the simulation setup. The choice of satisfaction function does have influence on the result of the simulation. When we set the satisfaction level to a fixed low point, 0.1, the simulation result shows that the fairness of peers increases a little bit with a significantly decreased efficiency. On the other hand, when we set the satisfaction level to a fixed high point, 0.9, the fairness of the peers decreases significantly with a more efficient system. It can be explained as follows. With a low satisfaction level, the trust increases slowly between peers which results in a high fairness but a low efficiency. The selection of the satisfaction is a trade-off between the fairness and system efficiency.

In all situations, the overhead of system is less than 1%.



(a) Unfairness

(b) Stability

Figure 8.8: Effect of Peer/Proxy Leaving Group without Notice

8.4.3 Simulation Results of Unpredictable Behaviors

For the influence of the unpredictable behavior of peers, we first simulate the scenario in which the peers may leave the group or the proxies may stop providing service without notifying other members. Figure 8.4.3 presents the effect on unfairness and stability when there are varying possibilities to leave a group without informing the other group members. The X axis represents the possibilities that a peer may leave group or stop service without informing others. The flat lines in the graphs shows that the system is robust. Even if all peers leave without notice and all proxies stop service without notice, the unfairness and the stability of system only increases about 7%. Therefore, the performance of the system does not suffer significantly by the unpredictable behavior of peers.

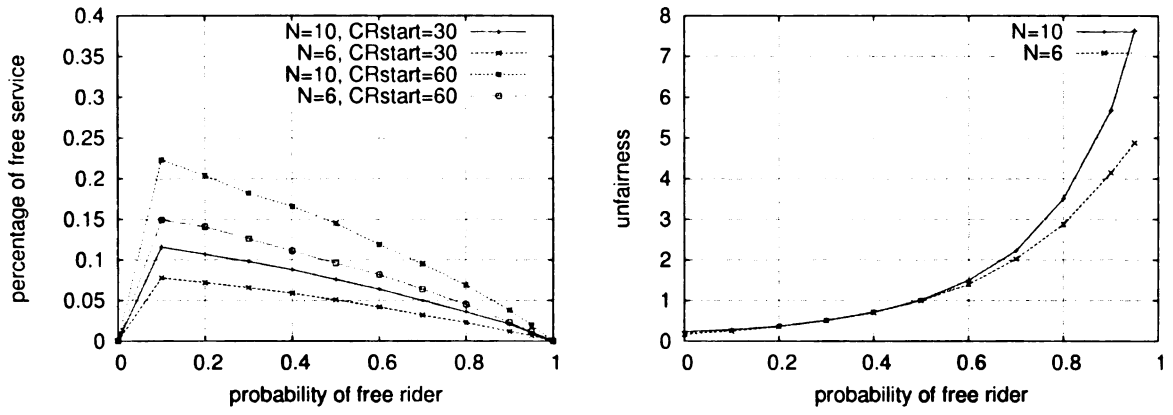
Message delay has some effect on the results. For a maximum delay of 1 s, unfairness increases from 0.213 to 0.217. Stability decreases from 179.21 to 144.27, mostly due

to message delay during the proxy competition protocol. Also, there is an average of 5.17 s difference in the amount of service recorded between peers.

For peers that periodically enter a power saving mode, the major problem is they lose contact with the proxy during sleeping, and if the proxy changes, there will be differences in the amount of time recorded for serving peers. The selection of the sleeping period is based on two considerations. A longer sleeping period reduces power consumption, but increases the possibility that contact with the proxy is lost. In our simulation, we set the sleeping period to 60 s. This amount is related to the selection of $CL_{start} = 60s$, which causes the minimum time that a proxy serves to be 60 s. The clients go to a sleep mode for 60 s, then awaken to contact the proxy as described in section 8.3.2. They immediately return to sleep mode if the previous proxy remains active. Otherwise, they ask for service from a new proxy. The simulation shows that the average lack of synchronization of service recorded between peers is 129.15 s. This is not large, and should not affect fairness significantly for peers that remain in a group for periods of tens of minutes.

8.4.4 Cheating and Attacks

We simulate the effect of free riders. The free riders will join the group, enjoy all the possible free services and never give service to other peers. In the first scenario, there is always a client in the system trying to obtain free service. The simulation result shows that for an average group size of 10, if there is only one free rider, only 8.1% of the time the free rider may receive service. For average group size of 6, this fraction



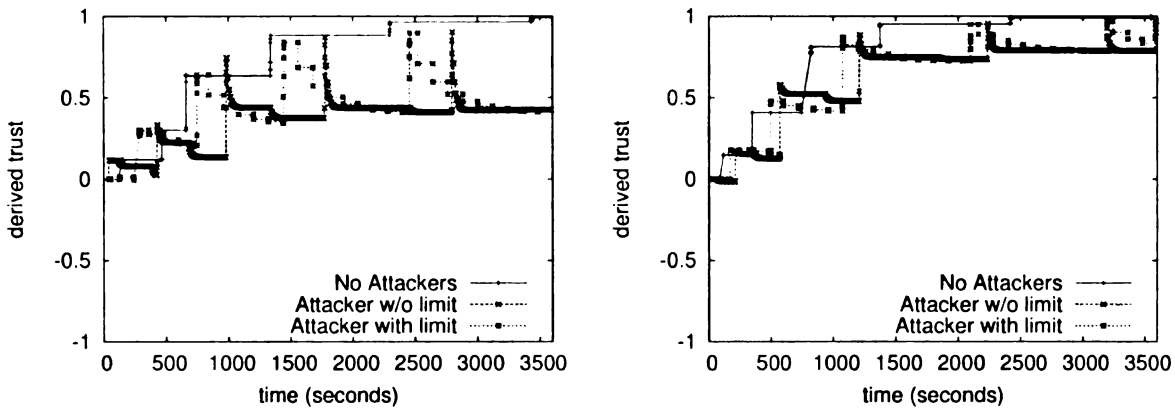
(a) Free Service

(b) Unfairness

Figure 8.9: Effect of Free Riders in the System

drops to 4.9%. In the second scenario, we change the probability of free riders in the system. The simulation result is shown in figure 8. Figure 8(a) illustrates the free service that can be obtained by the free riders. The result shows that a small group size and small start credit may limit the damage of the free riders. Figure 8(b) shows that the unfairness increases as the number of free riders increases. In a system that full of free riders, the unfairness is very high. However, everyone can only get a small fraction of service. Figure 8.4.4 is also a very good example of how the system collapses with free riders. As we have discussed in section 8.3.1, in our system the peer will gain more benefit by providing service to the others. Therefore, a rational peer will not chose to be a free rider.

We also simulate an attack of flooding of false trust information, which will ruin the reputation of an honest peer. We simulate a single static group with six peers. Peer 6 will flood the system with negative trust information (-1) for peer 2. First, we do not limit the rate of exchanging trust information. Peer 6 will broadcast the false



(a) Trust Evolution with $\delta = 0.7$

(b) Trust Evolution with $\delta = 0.9$

Figure 8.10: Trust about Peer 2 in Peer 1 over Time

trust value into the system every 10 seconds. Second, we limit the rate of exchanging trust information by 120 seconds. In figure 8.4.4, we show how the trust about peer 2 evolves in peer 1 over time. The result shows that even without any prevention of flooding of false information, the trust of peer 2 by peer 1 still remains positive. It means that peer 2 can still get service from peer 1 within a relatively short time period. The simple limitation on the information exchanging rate or an increase in the value of δ will help limit the damage due to false information.

8.5 Summary

We developed a distributed trust model and a mechanism for using credits in order to promote cooperation among strangers in mobile ad hoc networks. Peers benefit cooperatively when they follow the scheme. Malicious peers are excluded from the system. This approach has its advantages and disadvantages. First, it is simple and easy to implement even in resource limited mobile devices. It does not require

sophisticated encryption algorithms or security protocols. Second, the distributed model eliminates the necessity of a costly centralized trust authority connection. All the trust information and decisions are made locally. However, the disadvantages are also obvious. First, the trust information of every other peer in the group needs to be stored locally, which makes the scheme difficult to scale. Second, the lack of a security mechanism and long-term account leaves open some holes for attack, for example, the "frequently join and leave" attack in CHUM. Nevertheless, the model is most suitable for applications that have an incentive to cooperate and does not require strict security.

Our evaluation shows that peers who remain in a group for a long period are more likely to share services fairly. Nevertheless, short periods of cooperation are also relatively fair. The scheme is efficient, since proxy changes are only likely to occur every few minutes. When malicious peers join the group, they may be identified and excluded. Malicious peers or unexpected proxy failures do not significantly affect the fairness and stability of the scheme. Although the work in this chapter focused on a solution for CHUM, the ideas may be extended to other pervasive computing applications.

Chapter 9

Conclusion

9.1 Summary

Several different wireless technologies co-exist in today's wireless networks. We envision that mobile devices will carry multiple wireless interfaces to access them simultaneously. Mobile users are demanding for "anywhere, anytime, always-on and high-speed" wireless services. However, this requirement can not be meet by current wireless technologies. The popular WLANs and WWANs are limited in either coverage range or transmission data rates.

In this document, we try to utilize and combine the current wireless technologies to leverage their advantages and share the resources of the mobile devices without any changes to the underlying network infrastructure. A novel *Cooperative Integrated Wireless Network Architecture (CIWNA)* is proposed to take the advantage of both the high-speed WLANs and the cellular data networks for a better utilization of the costly cellular resource. The mobile devices cooperate and share their idle cellular

links to the WWAN via a mobile ad hoc network (MANET) formed from WLAN technologies. Some of the peers may act as servers or proxies, providing different kinds of services for the other members in the same MANET by contributing their idle cellular resource. By joining the cooperative network, the mobile devices may reduce their power consumption, improve the QoS, or reduce the latency of large file downloading. No modifications in the underlying wireless infrastructure and special hardware in the mobile devices are needed. The performance improvement in the cellular data networks comes from the efficient peer-to-peer sharing of the idle resources among the mobile users.

With the basic network architecture and the concept of peer-to-peer resource sharing, four different applications are proposed aiming to reduce the power consumption of the cellular interfaces, to improve the QoS of the cellular links, and to reduce the download latency of large files from Internet servers. A new set of network protocols are developed for CIWNA to address on the network formation, group management, proxy rotation, message routing, service discovery, failure recovery, and security and privacy.

In the application of *Cooperating ad Hoc network to sUpport Messaging* (CHUM), a new message notification protocol (MNP) is developed to enable a continuous IM presence in mobile devices with limited power consumption. The similar idea is used in a new framework to support a continuous WWAN event/message notification channel. We proposed a simple group management protocol to resolve the problem of high overhead in maintaining the precise group information in the highly dynamic and unreliable environment. The *QoS Aware Wireless Bandwidth Aggregation* system

can utilize both the cellular network interface and the IEEE 802.11 ad hoc network for an integrated network architecture that provides QoS aware wireless bandwidth aggregation. The capacity of several low throughput cellular links are shared by all mobile nodes to provide better QoS support for the application. The same idea of aggregating cellular data links is applied in another application for a novel cooperative parallel file downloading scheme to reduce the latency of downloading large files. This scheme utilizes multiple paths in the cellular data network with the help of the IEEE 802.11-based mobile ad hoc network. A simple and efficient proxy discovery protocol is used to find new proxies.

CIWNA requires cooperation among a group of users to gain benefit for the whole community. A distributed trust model and a mechanism for using credits are presented to promote the incentive of cooperation among a group of strangers in CIWNA. Peers benefit cooperatively when they follow the scheme. Malicious peers are excluded from the system. Game theoretical analysis is provided to justify the proposed trust model and credit system.

9.2 Future Work

9.2.1 Potential Applications

We have explored four applications of the cooperative integrated wireless architecture in this document. In these applications, the integrated wireless architecture is formed by utilizing cellular interfaces and WLANs interfaces in the mobile users. The mobile

resources such as the batter power and the cellular bandwidth are shared among a group of mobile users. More wireless applications may be benefit from the CIWNA. In these applications, the integrated network may be formed by all means of wireless technologies not only the cellular and the WLANs interfaces, and the shared resource may not be limited to the batter power and the bandwidth as we have discussed.

Let's imagine a scenario that a mobile user A is equipped with Bluetooth wireless interface and the IEEE 802.11 wireless interface. A nearby mobile user B may have both the 3G cellular interface and the IEEE 802.11 wireless interface. With the Bluetooth interface, A is capable of accessing nearby devices, while B has the Internet access via his 3G interface. Follow the similar idea we have discussed in this document, an integrated wireless network may be formed by utilizing all the wireless interfaces. The IEEE 802.11 based wireless interfaces can be used to form a mobile ad hoc network. Thus, A and B can share their different mobile resource: the Bluetooth connection to the nearby devices and the cellular link to the Internet. Still no infrastructure support or special hardware are needed. In another example, a mobile device with the GPS ability may contribute its precise location information to the nearby mobile devices. In return, it may be entitled to access the MP3 music collections in another mobile device. Content, storage, and processing power are other examples of resource that can be shared among mobile users. The similar idea may be extended into sensor networks and vehicle networks.

9.2.2 Centralized Infrastructure Support

The performance gain by CIWNA is limited by lacking the centralized infrastructure support. For example, in the parallel file downloading scheme described in chapter 7, the downloading path from the proxy to the destination client is determined on-demand by the proxy discovery protocol. The selected path is used for the whole piece downloading without considering the changing environment during this period. With some extent of centralized infrastructure support, packet level stripping may be achieved by letting the base station to select the best path for each packet to the destination client. Similarly, the QoS aware path selection is also limited by the incomplete information maintained in each mobile node in QAWBA. The mobile node may report its current traffic load and neighbors to the base station. With the complete information from all the mobile nodes, the utilization of the cellular network will be improved in both parallel file downloading and QAWBA.

On the other hand, a centralized secure charging scheme may be used to eliminate the burden of the distributed trust and credit system. Some kinds of micropayment scheme may also be considered.

Bibliography

Bibliography

[1] J.C. Liberti and T.S. Rappaport. *Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications.* Prentice Hall.

[2] Timothy J. Shepard. A Channel Access Scheme for Large Dense Packet Radio Networks. In *Proceedings of SIGCOMM*, pages 219–230, 1996.

[3] S. Lu, T. Nandagopal, and V. Bharghavan. A wireless fair service algorithm for packet cellular networks. In *Proceedings of ACM MOBICOM*, 1998.

[4] S. Seshan, H. Balakrishnan, and R. Katz. Handoffs in Cellular Wireless Networks: The Daedalus implementation and experience. 4(2):141–162, 1997.

[5] Ajay V. Bakre and B. R. Badrinath. I-TCP: Indirect TCP for mobile hosts. In *International Conference on Distributed Computing Systems*, pages 136–143, 1995.

[6] Hung-Yun Hsieh and Raghupathy Sivakumar. On Using Peer-to-Peer Communication in Cellular Wireless Data Networks. *IEEE Transactions on Mobile Computing*, 3(1):57–72, 2004.

[7] H. Luo et al. Internet Roaming: A WLAN/3G Integration System for Enterpises. In *Proceedings of Asia-Pacific Optical and Wireless Communication*, 2002.

[8] S.-H. Chan X. Wu and B. Mukherjee. Madf: A novel approach to add an ad-hoc overlay on a fixed cellular infrastructure. In *Proceedings of IEEE WCNC*, September 2000.

[9] C. Qiao and H. Wu. icar: An intelligent cellular and ad-hoc relay system. In *Proceedings of IEEE IC3N*, October 2000.

[10] A. Oram, editor. *Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology.* O'Reilly & Associates, March 2001.

[11] Microsoft Corporation. The Microsoft Instant Messaging Home Page.

[12] AIM. The AOL Instant Messaging Home Page.

[13] ICQ. The ICQ home page.

[14] YahooIM. The Yahoo! Instant Messaging Home Page.

[15] Jabber. The Jabber Home Page.

[16] B. Segall P. Sutton, R. Arkins. Supporting Disconnectness - Transparent Information Delivery for Mobile and Invisible Computing. In *CCGrid 2001 IEEE International Symposium on Cluster Computing and Grid*, May 2001.

[17] Sierra Wireless Inc. Sierra Wireless Inc. Homepage.

[18] P. Rodriguez and E. W. Biersack. Dynamic parallel access to replicated content in the Internet. *IEEE/ACM Transactions on Networking*, 10(4), August 2002.

[19] A. Zeitoun, H. Jomjoom, and M. El-Gendy. Scalable parallel-access for mirrored servers. In *Proceedings of IASTED International Conference on Applied Informatics*, February 2002.

[20] B. Cohen. Incentives build robustness in bittorrent. In *First Workshop on the Economics of Peer-to-Peer Systems*, June 2003.

[21] J. W. Byers, M. Luby, and M. Mitzenmacher. Accessing multiple mirror sites in parallel: Using tornado codes to speed up downloads. In *INFOCOM*, pages 275–283, New York, NY, March 1999. IEEE.

[22] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. In *SIGCOMM*, pages 56–67, 1998.

[23] IEEE. Wireless LAN medium access (MAC) and physical layer (PHY) Spec. IEEE 802.11 standard, 1998.

[24] M. Johnson. HiperLAN/2-The broadband radion transmission technology operating in the 5GHz frequency band.

[25] Jyh-Cheng Chen, Krishna M. Sivalingam, P. Agrawal, and Shalinee Kishore. A Comparison of MAC Protocols for Wireless Local Networks Based on Battery Power Consumption. In *Proceedings of IEEE INFOCOM*, pages 150–157, March 1998.

[26] Bluetooth SIG. "the bluetooth specification 1.1".

[27] Nitin Vaidya. TUTORIAL: Mobile Ad-hoc Networks: Routing, MAC and Transport Issues. In: ACM MobiCom Turotials. Boston, MA, 2000.

[28] Philippe Jacquet, Paul Muhlethaler, and Amir Qayyum. Optimized Link State Routing Protocol (Internet-Draft) draft-ietf-manet-olsr-00.txt. Technical report, Mobile Ad hoc Network (MANET) Working Group, IETF, Nov 1998.

[29] Charles Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994.

[30] C. E. Perkins and E. M. Royer. Ad-Hoc On-Demand Distance Vector Routing. In *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA99)*, pages 90–100, February 1999.

[31] David B Johnson and David A Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Imielinski and Korth, editors, *Mobile Computing*, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[32] Z. J. Haas and M. R. Pearlman. The zone routing protocol (ZRP) for ad hoc networks (Internet-Draft). Technical report, Mobile Ad hoc Network (MANET) Working Group, IETF, Aug. 1998.

[33] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking*, pages 85–97, 1998.

[34] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *International Conference on Mobile Computing and Neavorking (MobiCom'99)*, pages 195–206, 1999.

[35] Samir Ranjan Das, Charles E. Perkins, and Elizabeth E. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proceedsing of IEEE INFOCOM*, pages 3–12, 2000.

[36] R. D. Samir, C. Robert, Y. Jiangtao, and S. Rimli. Comparative performance evaluation of routing protocols for mobile, ad hoc networks. In *7th International Conference on Computer Communications and Networks (IC3N)*, pages 153–161, October 1998.

[37] Ching-Chuan Chiang, Hsiao-Kuang Wu, Winston Liu, and Mario Gerla. Routing in clustered multihop mobile wireless networks with fading channel. In *Proc. of IEEE Singapore International Conference on Networks*, pages 197–211, 1997.

[38] Prasun Sinha, Raghupathy Sivakumar, and Vaduvur Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. In *Proceedings of IEEE INFOCOM*, pages 202–209, 1999.

[39] A. Veres, A. T. Campbell, M. Barry, and L. H. Sun. Supporting Service Differentation in Wireless Packet Networks Using Distributed Control. *IEEE Journal of Selected Areas in Communications*, 2001.

[40] M. Mirhakkak, N. Schult, and D. Thomson. Dynamic quality-of-service for mobile ad hoc networks. In *International Conference on Mobile Computing and Networking*, pages 137–138, Boston, Massachusetts, 2000.

[41] Seoung-Bum Lee, Gahng-Seop Ahn, Xiaowei Zhang, and Andrew T. Campbell. INSIGNIA: An IP-Based Quality of Service Framework for Mobile ad Hoc Networks. *Journal of Parallel and Distributed Computing*, 60(4):374–406, 2000.

[42] Qi Xue and Aura Ganz. Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 63:154–165, 2003.

[43] Shigang Chen and Klara Nahrstedt. Distributed Quality-of-Service Routing in Ad-Hoc Networks. *IEEE Journal on Special Areas in Communications*, 17(8):1–18, 1999.

[44] Wen-Hwa Liao, Yu-Chee Tseng, Shu-Ling Wang, and Jang-Ping Sheu. A Multi-Path QoS Routing Protocol in a Wireless Mobile Ad Hoc Network. *Telecommunication Systems*, 19(3-4):329–347, 2002.

[45] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-Aware Routing in Mobile Ad Hoc Networks. In *the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 181–190, Dallas, TX, October 1998.

[46] Sanjay Udani and Jonathan Smith. Power Management in Mobile Computing (A Survey). Technical Report MS-CIS-98-26, University of Pennsylvania, 1996.

[47] Microsoft Corp. and Intel Corp. Advanced power management bios interface specification, Feb 1996. Available at http://www.microsoft.com/hwdev/archive/BUSBIOS/amp_12P.asp.

[48] Compaq Computer Corp., Intel Corp., Microsoft Corp., Phoenix Technologies Ltd., and Toshiba Corp. Advanced configuration and power interface specification v2.0a, March 2002. Available at http://www.acpi.info/spec.htm.

[49] Christine E. Jones, Krishna M. Sivalingam, Prathima Agrawal, and Jyh-Cheng Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7(4):343–358, 2001.

[50] R. Acharya J.-C. Chen, K. M. Sivalingam and P. Agrawal. Scheduling multimedia services for a low-power mac in wireless and mobile atm networks. *IEEE Transactions on Multimedia*, 1(2):187–201, June 1999.

[51] Christine E. Price, Krishna M. Sivalingam, Jyh-Cheng Chen, and Prathima Agarwal. Power-Aware Scheduling Algorithms for Wireless Networks. In *International Conference on Intelligent Computing and VLSI*, Kalyani, India, February 2001.

[52] Paul Lettieri, Christina Fragouli, and Mani B. Srivastava. Low power error control for wireless links. In *Proceedings of the third annual ACM/IEEE international conference on Mobile computing and networking*, pages 139–150. ACM Press, 1997.

[53] Ram Ramanathan and Regina Hain. Topology control of multihop wireless networks using transmit power adjustment. In *Proceedings of IEEE INFOCOM (2)*, pages 404–413, 2000.

[54] Roger Wattenhofer, Li Li, Paramvir Bahl, and Yi-Min Wang. Distributed topology control for wireless multihop ad-hoc networks. In *Proceedings of IEEE INFOCOM*, pages 1388–1397, 2001.

[55] Jae-Hwan Chang and Leandros Tassiulas. Energy Conserving Routing in Wireless Ad-hoc Networks. In *Proceedings of IEEE INFOCOM*, pages 22–31, Tel-Aviv, Israel, March 2000.

[56] Prathima Agrawal K. M. Sivalingam, Mani Srivastava and Jyh-Cheng Chen. Low-power access protocols based on scheduling for wireless and mobile atm networks. In *IEEE International Conference on Universal Personal Communications (ICUPC)*, pages 429 - 433, Oct 1997.

[57] S. Singh and C.S. Raghavendra. PAMAS: Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks. *SIGCOMM Computer Communication Review*, 28(3):5–26, July 1998.

[58] Phil Karn. MACA - A New Channel Access Method for Packet Radio. In *the 9th ARRL Computer Networking Conference*, pages 134–140, 1990.

[59] Michele Zorzi and Ramesh R. Rao. Error control and energy consumption in communications for nomadic computing. *IEEE Transactions on Computers*, 46(3):279–289, 1997.

[60] M. Zorzi and R. Rao. Energy constrained error control for wireless channels. *IEEE Personal Communications*, vol. 4:27–33, Dec 1997.

[61] Paul Lettieri and Mani B. Srivastava. Adaptive frame length control for improving wireless link throughput, range and energy efficiency. In *Proceedings of IEEE INFOCOM (2)*, pages 564–571, 1998.

[62] S.-L. Wu, Y.-C. Tseng, and J.-P. Sheu. Intelligent medium access for mobile ad hoc networks with busy tones and power control. *IEEE Journal on Selected Areas in Communications*, 18(9):1647–1657, Sep 2000.

[63] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Mobile Computing and Networking*, pages 85–96, 2001.

[64] Ying-Dar Jason Lin and Yu-Ching Hsu. Multihop cellular: A new architecture for wireless communications. In *INFOCOM2000*, pages 1273–1282, 2000.

[65] N. Salem, L. Buttyan, J. Hubaux, and M. Jakobsson. A charging and rewarding scheme for packet forwarding in multi-hop cellular networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, June 2003.

[66] H.-Y. Hsieh and R. Sivakumar. On using the ad-hoc network model in cellular packet data networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, June 2002.

[67] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu. UCAN: A Unified Cellular and Ad-Hoc Network Architecture. In *MOBICOM*, pages 353–367, September 2003.

[68] H. Li, M. Lott, M. Weckerle, W. Zirwas, and E. Schulz. Multihop communications in future mobile radio networks. In *PIMRC2002*, 2002.

[69] H.-Y. Hsieh and R. Sivakumar. Towards a hybrid network model for wireless packet data networks. In *IEEE Symposium on Computers and Communications (ISCC)*, July 2002.

[70] H.-Y. Hsieh and R. Sivakumar. Performance comparison of cellular and multi-hop wireless networks: A quantitative study. In *ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, June 2001.

[71] H.-Y. Hsieh and R. Sivakumar. A hybrid network model for cellular wireless packet data networks. In *IEEE Global Communications Conference (GLOBECOM)*, November 2002.

[72] H.-Y. Hsieh and R. Sivakumar. Internetworking wwans and wlans in next generation wireless data networks. In *International Conference on 3G Wireless and Beyond*, May 2002.

[73] R. Ananthapadmanabha, B. S. Manj, and C.S.R. Murthy. Multi-Hop Cellular Networks: The Architecture and Routing Protocols. In *Proceedings of IEEE Int'l Symp. Pers. Indoor and Mobile Radio Communication*, 2001.

[74] T. Harrold and A. Nix. Intelligent relaying for future personal communication systems. In *IEEE Colloquium on Capacity and Range Enhancement Techniques for the Third Generation Mobile Communications and Beyond*, February 2000.

[75] V. Sreng, H. Yanikomeroglu, and D. D. Falconer. Relayer Selection Strategies in Cellular Networks with Peer-to-Peer Relaying. In *IEEE VTC Fall 2003*, 2003.

[76] W.W. Lu, B.H. Walke, and Xuemin Shen. 4G Mobile Communications: Toward Open Wireless Architecture.

[77] N. Niebert, A. Schieder, H. Abramowicz, G. Malmgren, J. Sachs, and U. Horn. Ambient Networks: An Architecture for communication networks beyond 3G.

[78] Hung-Yun Hsieh and Raghupathy Sivakumar. A Transport Layer Approach for Achieving Aggregate Bandwidth on Multi-homed Mobile Hosts. In *Proceedings of MobiCom'02*, pages 83–94, 2002.

[79] G. Aggelou and R. Tafazolli. On the relaying capacity of next-generation gsm cellular networks. *IEEE Personal Communications Magazine*, 8(1), February 2001.

[80] 3GPP TSG-RAN. Opportunity driven multiple access. Technical Report 3G TR 25.924, December 1999.

[81] Hung-Yu Wei and Richard D. Gitlin. Two-Hop-Relay Architecture for Next-Generation WWAN/WLAN Integration. *IEEE Wireless Communication*, 4(2):24–30, April 2004.

[82] Peer to Peer Working Group. Internet peer-to-peer working group - p2p wg, 2005.

[83] JXTA. The jxta home page.

[84] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of SIGCOM*, pages 149–160, 2001.

[85] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. Oceanstore: An architecture for global-scale persistent storage. In *Proceedings of ACM ASPLOS*. ACM, November 2000.

[86] P. Druschel and A. Rowstron. PAST: A large-scale, persistent peer-to-peer storage utility. In *Proceedings of Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, May 2001.

[87] S. Ratnasamy, P. Francis, M. Handley, P. Karp, and S. Shenker. A scalable content-addressable network. In *Proceedings of the SIGCOMM*, pages 161–171, 2001.

[88] I. Clark, O. Sanberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability*, July 2000.

[89] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-peer computing. Technical Report HPL-2002-57, HP Laboratories, 2002.

[90] Napster. The Napster Home Page.

[91] D.J. Becker, T. Sterling, D. Savarese, J.E. Dorband, U.A. Ranawak, and C.V. Packer. Beowulf: A parallel workstation for scientific computation. In *Proceedings of the International Conference on Parallel Computing*, 1995.

[92] SETI@HOME. The SETI@HOME home page, 2001.

[93] Gnutella. The Gnutella Home Page.

[94] W. Bolosky, J. Douceur, D. Ely, and M. Theimer. Feasibility of a Serverless Distributed File System Deployed on an Existing Set of Desktop PCs. In *Proceedings of SIGMETRICS*, 2000.

[95] S. Gribble, A. Halevy, Z. Ives, M. Rodrig, and D. Suciu. What Can Peer-to-Peer Do for Database and Vice Versa. In *Proceedings of the WebDB: Workshop on Databases and the Web*, 2001.

[96] OpenCOLA. The OpenCOLA Home Page, 2001.

[97] A. D. Rubin, M. Waldman, and L. F. Cranor. Publius: A robust, tamper-evident, censorship-resistant, web publishing system. In *Proceedings of the 9th USENIX Security Symposium*, pages 59–72, 2000.

[98] Microsoft. Microsoft .NET Passport Technical Overview. Technical report, 2001.

[99] Groove. The Groove Networks.

[100] Endeavors Technology. Magi Enterprise Solution.

[101] E. Adar and B. Huberman. Free Riding on Gnutella. *First Monday, Peer-Reviewed Journal on the Internet*, 5(10), Oct 2000.

[102] G. Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.

[103] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 255–265, 2000.

[104] Sonja Buchegger and Jean-Yves Le Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes - Fairness In Dynamic Ad-hoc Networks. In *Proc. of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, pages 226–236, Lausanne, CH, June 2002. IEEE.

[105] L. Buttyan and JP. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. *ACM/Kluwer Mobile Networks and Applications*, 8(5):593–612, October 2003.

[106] P. Golle, K. Leyton-Brown, I. Mironov, and M. Lillibridge. Incentives for Sharing in Peer-to-Peer Networks. In *Int. Workshop on Electronic Commerce (WELCOM)*, pages 75–87, 2001.

[107] Alberto Blanc, Yi-Kai Liu, and Amin Vahdat.

[108] Michal Feldman and John Chuang. Hidden-Action in Multi-Hop Routing. In *Second Workshop on the Economics of Peer-to-Peer Systems*, June 2004.

[109] H. T. Kung and Chun-Hsin Wu. Differentiated Admission for Peer-to-Peer Systems: Incentivizing Peers to Contribute Their Resources. In *First Workshop on the Economics of Peer-to-Peer Systems*, 2003.

[110] Nicolas Christin and John Chuang. On the cost of participating in a peer-to-peer network. In *The 3rd International Workshop on Peer-to-Peer Systems (IPTPS'04)*, 2004.

[111] Jeff Shneidman and David Parkes. Rationality and Self-Interest in Peer to Peer Networks. In *The 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03)*, 2003.

[112] Luzi Anderegg and Stephan Eidenbenz. Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In *Proceedings of the 9th annual international conference on Mobile computing and networking.*

[113] David Turner and Ni Deng. Payment-Based Email. In *5th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD2004)*, 2004.

[114] Beverly Yang and Hector Garcia-Molina. PPay: Micropayments for Peer-to-Peer Systems. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, 2003.

[115] Silvio Micali and Ronald L. Rivest. Micropayments Revisited. In *Proceedings of the Cryptographer's Track at the RSA Conference 2002*, pages 149–163, 2002.

[116] Cynthia Dwork and Moni Naor. Pricing via Processing, Or, Combatting Junk Mail. In *Advances in Cryptology - CRYPTO'92*, 1992.

[117] Cynthia Dwork, Andrew Goldberg, and Moni Naor. On Memory-bound Functions for Fighting Spam. In *Crypto 2003*, 2003.

[118] Bram Cohen. Incentives Build Robustness in BitTorrent. In *First Workshop on the Economics of Peer-to-Peer Systems*, 2003.

[119] Brian F. Cooper and Hector Garcia-Molina. Peer-to-peer resource trading in a reliable distributed system. In *The 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, 2002.

[120] Tsuen-Wan Ngan, Dan Wallach, and Peter Druschel. Enforcing Fair Sharing of Peer-to-Peer Resources. In *The 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03).*

[121] K. Anagnostakis and M.B. Greenwald. Exchange-based Incentive Mechanisms for Peer-to-Peer File Sharing. In *Proceedings of The 24th IEEE International Conference on Distributed Computing (ICDCS 200)*, 2004.

[122] D. Qiu and R. Srikant. Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks. In *Proc. of ACM SIGCOMM*, 2004.

[123] Ruggero Morselli, Jonathan Katz, and Bobby Bhattacharjee. A Game-Theoretic Framework for Analyzing Trust-Inference Protocols. In *Second Workshop on the Economics of Peer-to-Peer Systems*, June 2004.

[124] Sonja Buchegger and Jean-Yves Le Boudec. A Robust Reputation System for P2P and Mobile Ad-hoc Networks. In *Second Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[125] John R. Douceur. The Sybil Attack. In *1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, 2002.

[126] E. Friedman and P. Resnick. The Social Cost of Cheap Pseudonyms. *Journal of Economics and Management Strategy*, 10(2):173–199, 2001.

[127] M. Landori. Social Norms and Community Enforcement. 59(1):63–80, 1992.

[128] P. Resnick and R. Zeckhauser.

[129] S. Marsh. Formalising Trust as a Computational Concept. Technical report, Ph.D. Dissertation, University of Stirling, 1994.

[130] T. Grandison and M. Sloman. A Survey of Trust in Internet Applications. *IEEE Communications Surveys*, 3(4), 2000.

[131] A. Abdul-Rahman and S. Hailes. Supporting Trust in Virtual Communities. In *HICSS '00: Proc. of the 33rd Hawaii International Conference on System Sciences-Volume 6*, page 6007. IEEE Computer Society, 2000.

[132] G. Suryanarayana and R. N. Taylor. A survey of trust management and resource discovery technologies in peer-to-peer applications. Technical Report UCI-ISR-04-6, ISR, 2004.

[133] L. Mui, M. Mohtashemi, and A. Halberstadt. A Computational Model of Trust and Reputation. In *HICSS'02: Proc. of the 35th Annual Hawaii International Conference on System Sciences - Volume 7*, page 188. IEEE Computer Society, 2002.

[134] C. M. Jonker and J. Treur. Formal Analysis of Models for the Dynamics of Trust Based on Experiences. In *Proceedings of the MAAMAW'99, Lecture Notes on AI*, volume 1647, pages 221–232. Springer-Verlag, 1999.

[135] M. Carbone, M. Nielsen, and V. Sassone. A Formal Model for Trust in Dynamic Networks. In *Proc. of 1st International Conference on Software Engineering and Formal Methods (SEFM 2003), 22-27 September 2003, Brisbane, Australia*, pages 54–. IEEE Computer Society, 2003.

[136] P. Zimmermann. The PGP User's Guide, October 1994. The International PGP Home Page, http://www.pgpi.org.

[137] R. Khare and A. Rifkin. Weaving a Web of Trust. *World Wide Web Journal*, 2(3):77–112, 1997.

[138] R. Chen and W. Yeager. Poblano: A Distributed Trust Model for Peer-to-Peer Networks. Technical report, Sun Microsystems, 2001. Available at http://www.jxta.org/project/www/White_papers.html.

[139] K. Aberer and Z. Despotovic. Managing Trust in a Peer-2-Peer Information System. In *Proc. of the 10th Int. Conference on Information and Knowledge Management*, pages 310–317, 2001.

[140] M. Winslett. An Introduction to Automated Trust Negotiation. In *Workshop on Credetential-Based Access Control*, October 2002.

[141] T. W. van der Horst, T. Sundelin, K.E. Seamons, and C.D. Knutson. Mobile Trust Negotiation: Authentication and Authorization in Dynamic Mobile Networks. In *Proc. of 8th IFIP Conference on Communications and Multimedia Security*, September 2004.

[142] A. Josang, R. Ismail, and C. Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. *(to appear). Decision Support Systems (2005)*, 2005.

[143] M. Blaze and J.Feigenbaum. Decentralized trust management. *IEEE Symposium on Security and Privacy*, 1996.

[144] L. Kagal and S. Cost. A framework for distributed trust management. In *Second Workshop on Norms and Institutions in MAS, Autonomous Agents*, 2001.

[145] Ting Yu, Marianne Winslett, and Kent E. Seamons. Interoperable strategies in automated trust negotiation. In *CSS'01: Proceedings of the 8th ACM conference on Computer and Communications Security*, 2001.

[146] M.Blaze and J. Feigenbaum. The Role of Trust Management in Distributed Systems Security. In *Secure Internet Programming*, pages 185–210, 1999.

[147] Matt Blaze, Joan Feigenbaum, and Angelos D. Keromytis. KeyNote: Trust Management for Public-Key Infrastructures. 1550:59–63, 1999.

[148] Yang-Hua Chu, Joan Feigenbaum, Brain LaMacchia, Paul Resnick, and Martin Strauss. REFEREE: Trust Management for Web Applications. 29(8-13), 1997.

[149] G. Zacharia and P. Maes. Trust Management Through Reputation Mechanisms. *Applied Artificial Intelligence*, 14:881–907, 2000.

[150] F. Cornelli, E. Damiani, S.C. Vimercati, S. Paraboschi, and P. Samarati.

[151] F. Cornelli, E. Damiani, S.C. Vimercat, S. Paraboschi, and P. Samarati. Choosing reputable servents in a p2p network. In *Proceedings of the eleventh international conference on World Wide Web*, 2002.

[152] S. Lee and R. Sherwood. Cooperative peer groups in NICE. In *Proceedings of IEEE INFOCOM*, 2003.

[153] M. Gupta and P. Judge. A Reputation System for Peer-to-Peer Networks. In *Thirteenth ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2003.

[154] A. Josang and R. Ismail. The Beta Reputation System. In *Proceedings of 15th Bled Electronic Commerce Conference*, 2002.

[155] K. Aberer. P-grid: A self-organizing access structure for p2p information systems. In *Proc. Of the Ninth International Conference on Cooperative Information Systems*, 2001.

[156] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In *Proceedings of the Twelfth International WWW Conference*, 2003.

[157] A. Abdul-Rahman and S. Hailes. A Distributed Trust Model. In *Proceedings of the ACM New Security Paradigms Workshop*, pages 48–60, 1997.

[158] Jordi Sabater and Carles Sierra. REGRET: repuation in gregarious societies. In *Proceedings of the Fifth INternational Conference on Autonomous Agents*, pages 194–195, 2001.

[159] J. Pujol and R. Sanguesa. Extracting reputation in multi agent systems by means of social network topology. In *First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2002.

[160] B. Yu and M.P. Singh. A Social Mechanism of Reputation Management in Electronic Communities. In *Proceedings of Fourth International Workshop on Cooperative Information Agents*, 2000.

[161] M. Schill, P. Funk, and et al. Using Trust for Detecting Deceitful Agents in Artificial Science. *Applied Artificial Intelligence Journal, Special Issue on Trust Deception and Fraud in Agent Societies*, 2000.

[162] J. Sabater and C. Sierra. Reputation and social network analysis in multi-agent systems. In *First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2002.

[163] B.H. Bloom. Space/time Trade-offs in Hash Coding with Allowable Errors. *Communications of the ACM*, 13(7):422–426, 1970.

[164] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. Summary cache: a scalable wide-area web cache sharing protocol. 8(3):281–293, 2000.

[165] M. Mitzenmacher. Compressed bloom filters. In *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pages 144–150. ACM Press, 2001.

[166] IEEE 802.15 Working Group for WPAN. The IEEE 802.15 Working Group Homepage.

[167] Laura Marie Feeney and Martin Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proc. INFOCOM'01*, volume 3, pages 1548–1557, 2001.

[168] Novatel Wireless Inc. Novatel Wireless Inc. Homepage.

[169] CSR. The csr homepage.

[170] D. Zhu and M. Mutka. Promoting Cooperation among Strangers to Access Internet Service from an Ad Hoc Network. In *The second IEEE Annual Conference on Pervasive Computing and Communications (PerCom2004)*, March 2004.

[171] D. Zhu and M. Mutka. Sharing Presence Information and Message Notification in an Ad Hoc Network. In *The first IEEE Annual Conference on Pervasive Computing and Communications (PerCom2003)*, pages 351–358, March 2003.

[172] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication, from proceedings of the royal society, volume 426, number 1871, 1989. In *William Stallings, Practical Cryptography for Data Internetworks, IEEE Computer Society Press.* 1996.

[173] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *IEEE INFOCOM2003*, 2003.

[174] D. Fudenberg and J. Tirole, editors. *Game Theory.* MIT Press, 1991.