Peer-to-Peer Applications in Ad Hoc Wireless Networks

Haixia Liu

A dissertation submitted to the University of Dublin,
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science

September 2002

Declaration

I declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed: ____________________

September 2002

Permission to lend and/or copy

I agree that Trinity College Library may lend or copy this dissertation upon request.

Signed: ____________________

September 2002

Abstract

With the rapid development of wireless technology, more and more people are equipped with mobile devices, which allow them to communicate and share resources each other. The combination of mobility and wireless networking makes the emergence of mobile computing paradigm that gives more flexibility and opportunities to people.

A mobile ad-hoc network is a system comprised of mobile devices that act as both hosts and routers, communicating wirelessly in an arbitrary way without an existing network infrastructure. It is useful for setting up a network quickly where network infrastructure does not exist or not required when people need to communicate and share information urgently and quickly. These mobile devices are free to move and the topology of this kind of networks is dynamic and unpredictable. Because of the characteristics of mobility and the limitations of wireless communication distance such as IEEE802.11b WLAN, multi-hop routing is always an important issue in ad-hoc networks.

Peer-to-peer architecture is a good solution for mobile computing systems working in ad-hoc wireless network environment where it is necessary to discover resources and dynamically route information through the network. In peer-to-peer systems each node has equivalent capabilities and responsibilities. In this way, peer-to-peer applications allow mobile users benefit from fully decentralized and share resources efficiently.

This project will address the design and implementation of a peer-to-peer infrastructure over a mobile ad-hoc network, enabling users with mobile devices to chat with each other even while moving. Using the chat application to demo how peer-to-peer architecture works well with mobile ad-hoc wireless network, and to research the advantages and challenges in the design and implementation of this kind of peer-to-peer applications.

The chat application has successfully enabled people with mobile devices to chat each other in a multi-hop mobile ad-hoc wireless environment. During the moving, every node can know the changes of network topology dynamically and the nodes in an area can act as routers and forward messages among nodes that cannot communicate each other directly.

Acknowledgements

Thanks to my supervisor, Dr. Vinny Cahill, for his guidance and advice throughout the course of this project.

Thanks to Peter Barron for his willingness to help me with questions and problems I had met during this year.

Thanks to my classmate Michael, who always gives me encouragement when I need.

Thanks to other members of the M.Sc. course for their friendship and help during the year.

Thanks to Madhu, and other DSG group members who provided me with laptops and Pocket PCs for my tests and those who helped me with the questions of this project.

Finally, I dedicate this dissertation to my family, many thanks for their love, constant support and encouragement throughout this year.

Table of content

List of Figures

Chapter 1

Introduction

1.1 Motivation

With the rapid development of wireless technologies, mobile devices have become much more common communication and computing devices than ever before. Fixed computers are no longer the only choice to network applications and people. The number of mobile devices has increased continuously and rapidly throughout the whole world. Datamonitor predicts that by 2005, there will be one billion wireless device users worldwide. [1]

Wireless networking technologies have resulted in completely new ways of communication and computing. With mobile devices, people can carry mobile devices while moving and still access shared resources and services continuously and at any time they want without the limitations of wired networks. In wireless environments, people can also communicate with each other much more easily without pre-existing network infrastructure. For example, according to IDC, mobile banking has grown in Western Europe alone from a $12 million industry in 1999, to a $116 million industry in 2001. By 2003, IDC predicts this figure will jump to $344 million. [2] Celent Communications reports that 150 million users will engage in wireless financial services by 2004 worldwide, especially in Europe and Asia-Pacific regions. [3]

The trend of integration of communication and computing is clear. According to Gartner Group, 60% of office productivity workers will carry or own at least three mobile devices within three years. Over 18 million PDAs have been sold since their introduction. With the development of wireless networks and mobile technology, applications on mobile devices and in wireless networks also are in tremendous growth.

A mobile ad-hoc network is a system comprised of mobile devices that act as both hosts and routers, communicating wirelessly in an arbitrary way without an existing network infrastructure. It is useful for setting up a network quickly where network infrastructure does not exist or not required when people need to communicate and share information urgently and quickly. These mobile devices are free to move and the topology of this kind of networks is dynamic and unpredictable. Because of the characteristics of mobility and the limitations of wireless communication distance such as IEEE802.11b WLAN, multi-hop routing is always an important issue in ad-hoc networks.

Peer-to-peer technology compared to traditional client/server technology has already a history of more than 20 years. But today, with the trends of distributed and decentralized software engineering, the availability of powerful networked computers and inexpensive bandwidth, new computing paradigms are needed to meet the situation. Mobile computing is such a paradigm that allows people with mobile devices to access resources and services while moving. Peer-to-peer technology is perfect for mobile computing paradigm because of its characteristic of equivalent capabilities among all nodes. In peer-to-peer systems each node has equivalent capabilities and responsibilities. In this way, peer-to-peer applications allow mobile users benefit from fully decentralized and share resources efficiently. Peer-to-peer architecture is a good solution for mobile computing systems working in ad-hoc wireless network environment where it is necessary to discover resources and dynamically route information through the network.

1.2 Objective

The objective of the project is to make a complete and detailed research about the fields of wireless technology, mobile computing, ad-hoc networks, peer-to-peer systems and especially the integration of peer-to-peer systems with ad-hoc networks. The goal is to research the advantages as well as challenges of these kinds of applications, the main design issues of them and how to solve them.

In order to deepen the understanding and analysis, a peer-to-peer chat application in ad-hoc wireless network has been designed and implemented. In this application, the following goals will be achieved:

- Pure peer-to-peer application

  A pure peer-to-peer application means that there is no computer that acts as any kind of server in the system. Every peer acts as a server and a client at the same time to provide services to other peers or request services from other peers. These peers communicate directly with each other without the use of any access point.

- Multi-hop routing among the peers

  One of the characteristics of ad-hoc network is that all nodes are able to move and change their location arbitrarily, frequently and independently. A node may easily move out of communication range, so the nodes in the intersectional area need to act as a router to forward messages between peers that cannot communicate directly.

- Discovery of peers automatically

  In a pure peer-to-peer architecture, there is no server for users to register. And unlike wired network environment most of the computers have stationary locations, in an ad-hoc mode, it is not possible to know in advance who will take part in and when. So each peer has to discovery other peers itself and automatically.

- Instant message service

  The chat application provides instant message service that enables people to exchanges messages in real time.

- Open different chat rooms

  The users of the chat application can open or enter different chat rooms to talk about their favorite topics.

1.3 Roadmap

Chapter 1 ***Introduction*** gives a brief introduction about the technology of wireless network and peer-to-peer computing, the motivation of the project and the goals of the project.
Chapter 2 ***State of the Art*** will review the state of the art of the technical fields of wireless technology, mobile computing, ad-hoc networks, peer-to-peer systems and the integration of these four fields.
Chapter 3 ***Background*** will provide more details about the technologies that have close relationship with the design and implementation of the project: IP multicast, Jeode for Microsoft Windows CE as well as the description of the scenarios of using this chat application.
Chapter 4 ***Design*** will describe the requirement of the ad-hoc peer-to-peer chat application, the main design issues of the application as well as the algorithms or methods to solve these issues, the functionalities of the application and the design of the components that will implement these functionalities.
Chapter 5 ***Implementation*** will give the detailed implementation of all the components designed in chapter 4, how these components integrated together to complete the application, how data flows and the tests of the implementation.
Chapter 6 ***Conclusion*** will evaluate the implementation of the project, the benefits and limitations of it and the trends of peer-to-peer systems.

Chapter 2

State of the Art

2.1 Wireless Technology

2.1.1 Mobile Cellular Networks

Wireless communications have become pervasive and are closely linked to our lives. The number of mobile phones and wireless Internet users has increased significantly in recent years. Mobile phones are not only devices for voice communications but also mobile computing devices. First-generation wireless networks were targeted primarily at voice and data communications occurring at low data rates. [4] [5] [6]

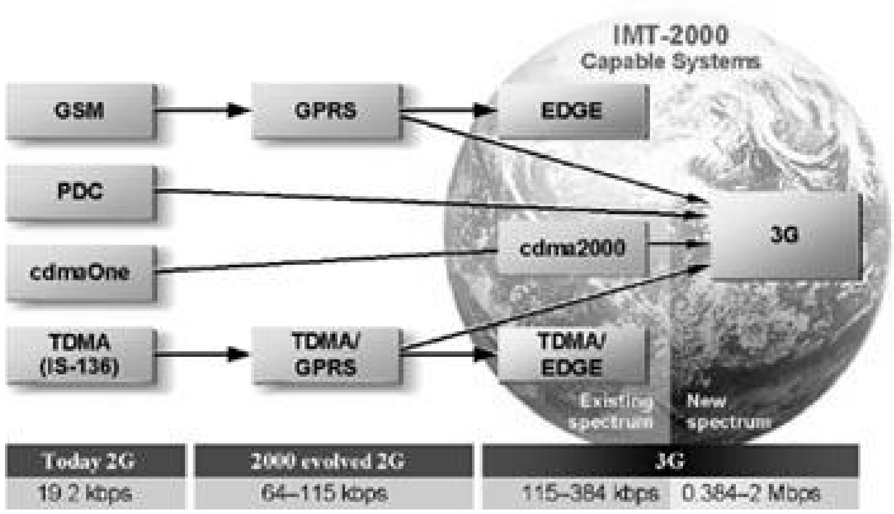


Figure 2.1 the evolution from 2G to 3G

First-generation

The first generation of wireless intended primarily for analog voice transmission. Advanced Mobile Phone Service (AMPS) was introduced to the North American market in 1983, initially in Chicago, and subsequently adopted by more than 35 countries in North and South America. Operating in the 800Mhz band, AMPS uses Frequency Division Multiple Access (FDMA), dividing this spectrum into 30Khz channels for signal transmission. The Europeans also deployed a system called Total Access Communications System (TACS). It is actually comprised of three sub-systems - Extended TACS for the United Kingdom market, International TACS covering non-UK Europe, and Narrowband TACS, which uses narrower channel spacing thereby enabling more channels in the same spectrum. Both AMPS and TACS use the frequency modulation (FM) technique for radio transmission. Traffic is multiplexed onto an FDMA system. Analog became obsolete as usage increased. Cellular communications breaks regions into zones, called 'cells' where communications can use a specific channel. As long as the cells don't overlap, the same channel can be used in discrete cells. The issue is one of exhaustion of all available channels, preventing new users. In order to expand subscriber base, wireless providers had to find a way to add more users in the same space. Because digital signals can be compressed, enabling more effective use of the spectrum, and analog cannot, it was natural that a conversion to digital technology would occur.

Second-generation Mobile Systems

Second-generation (2G) systems appear in the early 1990's resulted from the research of using digital encoding for voice wireless transmission. Compared to first-generation systems, 2G systems use digital multiple access technology, such as TDMA (time division multiple access) and CDMA (code division multiple access). GSM (Global System for Mobile Communications) GSM uses narrowband TDMA, which allows eight simultaneous calls on the same radio frequency to support multiple users. GSM is the dominant cellular technology in the European market and is also widely adopted in Asia. GSM operates in the 900Mhz and 1.8GHz bands in Europe and the 1.9GHz band in the United States. Most notably, GSM features a smart-card capability (called a Subscriber Identity Module, SIM), enabling rapid transfer of user identity from phone to phone, as well as data capability using a short messaging service (SMS) to send and receive messages of up to 160 characters. Such interoperability means greater flexibility for users, enabling them to operate different mobile communications devices in diverse locations while retaining their personal preferences and identity.

TDMA, Short for Time Division Multiple Access, is a technology for delivering digital wireless service using time-division multiplexing (TDM). TDMA works by dividing a radio frequency into time slots and then allocating slots to multiple calls. In this way, a single frequency can support multiple, simultaneous data channels.

CDMA, Short for Code-Division Multiple Access, is a digital cellular technology that uses spread-spectrum techniques. Unlike competing systems, such as TDMA, CDMA does not assign a specific frequency to each user. Instead, every channel uses the full available spectrum. Individual conversations are encoded with a pseudo-random digital sequence.

2G networks are in current use around the world. The protocols behind 2G networks support voice and some limited data communications, such as Fax and short messaging service (SMS), and most 2G protocols offer different levels of encryption, and security. While first-generation systems support primarily voice traffic, second-generation systems support voice, paging, data, and fax services.

2.5G Mobile Systems

Even with SMS for GSM, it was obvious that the original purpose of wireless networks, to provide voice communication not overly suitable for data transmission. A number of stopgap measures have been devised to increase utility and speed of data transmission for mobile email and Internet use before the use of 3G systems. One of the principal innovations is General Packet Radio Service (GPRS), which is a packet enhancement, operating at 115 Kbps (compared to GSM ’s current rate of 9.6 Kbps), supports a range of bandwidths, and is designed for data 'bursts' like email downloading and Internet browsing. GPRS is a radio technology for GSM networks that adds packet-switching protocols, shorter setup time for ISP connections, and the possibility to charge by the amount of data sent, rather than connection time. Packet switching is a technique whereby the information to be sent is broken up into packets, of at most a few Kbytes each, which are then routed by the network between different destinations based on addressing data within each packet. GPRS will support flexible data transmission rates as well as continuous connection to the network. GPRS is the most significant step towards 3G.

The next generation of data heading towards third generation and personal multimedia environments builds on GPRS and is known as Enhanced Data rate for GSM Evolution (EDGE). EDGE is a faster version of GSM wireless service. EDGE enables data to be delivered at rates up to 384 Kbps on a broadband. The standard is based on the GSM standard and uses TDMA multiplexing technology. EDGE will allow GSM operators to use existing GSM radio bands to offer wireless multimedia IP-based services and applications at theoretical maximum speeds of 384 kbps with a bit-rate of 48 kbps per timeslot and up to 69.2 kbps per timeslot in good radio conditions. EDGE will let operators function without a 3G license and compete with 3G networks offering similar data services. Implementing EDGE will be relatively painless and will require relatively small changes to network hardware and software as it uses the same TDMA (Time Division Multiple Access) frame structure, logic channel and 200kHz carrier bandwidth as today's GSM networks.

GERAN is a term used to describe a GSM and EDGE based 200kHz radio access network. The GERAN is based on GSM/EDGE Release 99, and covers all new features for GSM Release 2000 and subsequent releases, with full backward compatibility to previous releases.

Third-generation Mobile Systems

The use of hierarchical cell structures is proposed for IMT2000. The overlaying of cell structures allows different rates of mobility to be serviced and handled by different cells. Advanced multiple access techniques are also being investigated, and two promising proposals have evolved, one based on wideband CDMA and another that uses a hybrid TDMA/CDMA/FDMA approach.

It's without doubt the biggest individual telecommunications project the world has ever seen. Representing billions of dollars of investment from hundreds of the world's mobile network operators and equipment manufacturers, Third Generation (3G) Mobile will soon bring high quality multimedia services to people everywhere. Third-generation mobile systems are faced with several challenging technical issues, such as the provision of seamless services across both wired and wireless networks and universal mobility.

UMTS Forum Market Forecasts that:

- In 2010 the average 3G subscriber will spend about $30 US dollars per month on 3G data services
- Total operator-retained annual revenues of over $300 billion US dollars for 3G services by 2010.
- Non-voice service revenues will dominate voice revenues by 2004 and comprise 66% of 3G service revenues by 2010
- Asia Pacific represents the single largest total revenue opportunity - reaching $120 billion US dollars in 2010
- Europe and North America will provide the highest annual revenue per head of population ($150 - $200 US dollars per year).

UMTS: UMTS stands for 'Universal Mobile Telecommunications System', is one of the major new 'third generation' (3G) mobile communications systems being developed within the framework defined by the ITU and known as IMT-2000. UMTS will play a key role in creating the mass market for high-quality wireless multimedia communications that will exceed 2 billion users worldwide by the year 2010. This market will be worth over 1 trillion US dollars to mobile operators over the next ten years. UMTS will enable the wireless Information Society, delivering high-value broadband information, commerce and entertainment services to mobile users via fixed, wireless and satellite networks. UMTS will speed convergence between telecommunications, IT, media and content industries to deliver new services and create fresh revenue-generating opportunities and will deliver low-cost, high-capacity mobile communications offering data rates as high as 2Mbit/sec under stationary conditions with global roaming and other advanced capabilities

WCDMA, short for wideband CDMA, is a high-speed 3G mobile wireless technology with the capacity to offer higher data speeds than CDMA. WCDMA can reach speeds of up to 2 Mbps for voice, video, data and image transmission. WCDMA was adopted as a standard by the ITU under the name "IMT-2000 direct spread."

Forth-generation Mobile Systems

As expectations for the speeds of 3G networks decrease, interest rises in fourth-generation (4G) wireless networks. 4G networks are expected to be based on orthogonal frequency division multiplexing (OFDM), which sends data over hundreds of parallel streams, increasing the available bandwidth. Fourth-generation systems will also use technologies like adaptive processing (which helps clear up interference in transmission) and so-called smart antennas, which include a signal-processing capability to optimize their reception and radiation patterns.

2.1.2 Wireless Networking Technologies

3G has a bright future, but it isn't the ideal answer for all wireless data needs. 3G is a wide-area network technology and WLAN is not. [7]

WLAN

802.11 is a family of wireless networking protocols out of the IEEE. The most popular of these, 802.11b, has been in commercial use since 1999. It's the standard used by Apple Computer in its AirPort technology, as well as being adopted in wireless PC cards from Agere (formerly of Lucent) and Cisco. It has a maximum theoretical throughput of 11 Mbps, which is only about one-tenth the speed of common Ethernets, but much faster than broadband solutions. Other standards include 802.11g, an upcoming wireless networking protocol with speeds up to 22 Mbps (due in late 2001) and 802.11a, with speeds up to 54 Mbps (due in 2002).

WLANs are excellent technologies. The speed of WLANs is at up to 11 Mbps, WLANs offer much higher data rates than 3G's maximum 2 Mbps. For applications requiring high bandwidths, Wi-Fi or 802.11b WLAN is uniquely placed. Secondly, WLANs are cheaper then 3G to use, install, and maintain. WLANs are practical and effective today. Thirdly, WLANs are available and mature compared to 3G, which may not be available for several years. Many business offices and education spots like universities have aggressively deployed WLANs as an easy and flexible way to provide network services. WLAN technology extends the existing wired networks. Wired LANs are commonly used in offices and inside buildings but they are expensive for relocating all computer facilities and with the constraint of cables and distance. Where WLANs are used, they can connect to wired line backbone network to access resources and services. In our classroom, laptop computers installing with wireless cards can connect to wired network through access point and perform all functions and services provided by college. And we can move freely within the coverage area of the access point and get access to the network in and out of classroom or when we are moving.

WLANs can be used either to replace wired LANs or act as an extension of the wired LAN infrastructure.

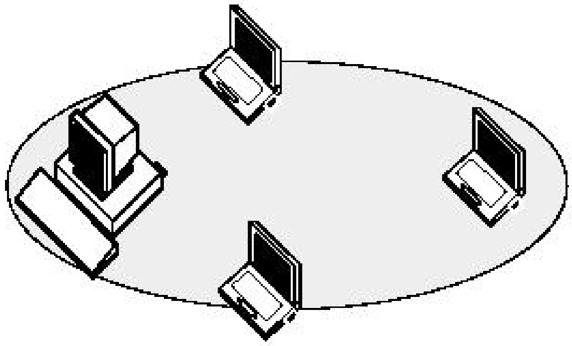


Figure 2.2 peer-to-peer wireless workgroup

Peer-to-Peer Wireless Workgroup option can be used to setup a temporary or ad-hoc network in environments where no access points are available (for example in Small Office/Home Office environments). As long as the stations are within range of one another, this is the easiest and least expensive way to set up a wireless network.

Bluetooth

[15] Bluetooth wireless technology is emerging as a de facto standard for short-range wireless connections between cell phones and other mobile and portable devices. The Bluetooth Special Interest Group consists of companies in telecommunications, computing, and networking industries that are driving the development of the technology and backing it on the market.

Bluetooth is an emerging standard and a specification, as the words of the Bluetooth SIG [special interest group]), bluetooth is small-form factor, low-cost, short range radio links between mobile PCs, mobile phones and other portable devices. It is a wireless connection between PCs, peripherals, and mobile devices that will let the devices share information and access services, without having to depend on a wired connection.

The bluetooth specification's origins date back to 1994, when four companies joined Ericsson to develop the technology: Nokia, Toshiba, Intel, and IBM. Today, the Bluetooth SIG includes nearly 2,000 companies, and prototype devices are beginning to make their way into the marketplace.

Bluetooth uses the radio waves located in the frequency band of 2.4 GHz (2400 to 2483.5 MHz), an increasingly popular (and crowded) slice of the spectrum. In this band, Bluetooth transmits voice and data at flows lower than 1 megabit per second.

A Bluetooth network (known as Piconet) can allow the interconnection of eight devices in a radius of 10 meters. This network can be fixed or provisional (a mobile or transitory network). In a Piconet, the Master seeks the devices in its entourage by emitting requests (broadcast). The slave answers with its identification number. By default, Piconets transmit up to 10 meters. However, it can be increased to 100 meters by increasing the power output of 100 milliwatts, as opposed to the 1 mW of default Bluetooth. However, compared to GSM, which consumes between 1.5 and 2 Watts, this is still a weak signal. Manufacturers are working to make Bluetooth devices that adapt to the necessary proximity, so as not to consume more energy than is necessary.

Bluetooth's promoters are positioning it as the technology for the Personal Area Network (PAN), and are targeting appliances that don't require large flows -- like printers, personal computers, and mobile phones. One concept that's been put forward is the mobile PAN: a communication device clipped to your belt could contain a GSM transceiver that communicates with the wider world. Meanwhile, the same device has a Bluetooth transceiver that communicates with the handsets, PDA, MP3 player, allowing all these devices to communicate with each other and the larger world. Since it is not a very expensive technology (between $5 and $20 per chip), it can easily be placed in many devices. Also, Bluetooth doesn't require an access point, unlike the traditional wireless networks. It's well suited for mobile devices, since it can join a local Piconet quickly, as soon as the two devices are in a sufficient perimeter. And unlike infrared networks (like two Palm computers beaming each other), Bluetooth doesn't require the objects to been aligned to communicate.

Infrared

Infrared is another technology that enables users to connect instantly and communicate in a wireless environment with mobile devices and other networking devices with infrared interfaces. Infrared devices are currently the most wide-used devices for wireless networks; almost all mobile devices are IrDA compatible devices. [19]

IrDA is an international organization that creates and promotes interoperable, low cost, infrared data interconnection standards that support point-to-point user model. It is a protocol suite designed to support transmission of data between two devices over short-range point-to-point infrared at speeds between 9.6 kilobits per second (Kb/s) and 4Mb/s. [21]

IrDA Data Protocols consist of a mandatory set of protocols and a set of optional protocols. The mandatory protocols are: [20]

- PHY (Physical Signaling Layer)

  The range of operation for contact is at least 1 meter and typically 2 meters can be reached. The data transmission speed is from 9600 b/s up to 4 Mb/s. The Fast IrDA physical layer specification defines short-range low power operation at 4 Mb/s (half duplex). Data packets are protected using a CRC (CRC-16 for speeds up to 1.152Mb/s and CRC-32 at 4 Mb/s).
- IrLAP (Link Access Protocol)

  IrLAP provides a device-to-device connection for the reliable, ordered transfer of data.
- IrLMP (Link Management Protocol and Information Access Service (IAS))

  IrLMP provides multiplexing of the IrLAP layer, provides multiple channels above an IrLAP connection.

The benefits of Infrared technology are that it has low power requirements and low costs, the hardware is very simple, so it can be spread to wild ranges of mobile devices; Infrared technology has specifications and standards provided by IrDA, so it has good compatibility; When using infrared interface, two devices must beam each other to communicate, so it is hard to get the information leaked.

But the Infrared technology has some constraints. Firstly, two devices must see each other to communicate so objects may block the communication path. Secondly, the distance of communication is short and the speed is low.

2.2 Mobile computing

Development of wireless networking technology have brought out a new paradigm of computing, it is the mobile computing, in which users carrying mobile devices can access shard services or resources independent of their location and while moving. This paradigm is much more flexible than general fix wired network computing paradigm. People with mobile computing devices equipped with wireless interfaces can access network or communicate each other even while moving. With more and more people now are equipped with laptop computers, Pocket PCs and PDAs, mobile computing extends the utility of these devices. The combination of mobility and networking will engender new fields of computing applications and give more flexibility, places and chances for peoples.

Compared to wired networks, there are some challenges of mobile computing. The main challenge is the influence of environment such as signal, noise and shield of buildings etc. The networks do not always have good availability and quality. Secondly, network connection and information may get lost during the mobility. It may because by short and suddenly disconnection of network, or, it may because of the latency and out of the coverage of the network. Thirdly, mobile devices themselves have limited capabilities such as lower computing power, battery power, smaller screen size and so on. The wireless networks have lower bandwidth compared to wired network and less security control capabilities.

So design mobile computing systems, people have to consider the issues: [18]

- Disconnection

  Wireless communications are susceptible to disconnection because of much more frequent disconnection compared to wired ones.

- Low bandwidth

  Wireless networks have much lower bandwidth compared to wired networks. IEEE802.11b has 11Mbps while Ethernet provides 10M, 100M, 1000Mbps.

- Security

  Wireless networks are easily to be connected and compromised.

- Mobility

  The ability of change locations while connected to the network increase the volatility of some information that are often static for wired network, such as address, locate dependent information.

- Limited capabilities of mobile devices themselves

  Mobile devices normally have low computing power, limited storage capability, low battery capability, small user interface and more difficulty to input.

2.3 Ad-hoc Networks and Routing Protocols

2.3.1 Ad-hoc Networks

Ad-hoc network is an 802.11 networking structure in which devices communicate directly with each other, without the use of infrastructure. Ad-hoc mode is useful for establishing a network where wireless infrastructure does not exist or not required. Ad-hoc mode is ideal for mobile computing. As IETF described, “A ‘mobile ad hoc network’ (MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links--the union of which form an arbitrary graph. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet."[12]

MANET has some typical characteristics:

- Infrastructureless network

  MANETs are always short of network backbone. Wireless nodes organize MANET spontaneously.

- Mobility and dynamic topology

  All nodes of MANET are capable of movement. They move their locations in an arbitrary way frequently and independently. The topology of MANET is always changing and the nodes establish their communication dynamically.

- Routing

  Routing is the main issue in ad-hoc networks. Because normally there is no default router working in an ad-hoc network, every node of the network may act as a router and forward the messages among nodes.

  There are a lot of routing protocols developed for ad-hoc networks. Some of them will be discussed in the part of 2.3.2.

- Limitation of reliability and bandwidth

  The nodes in ad-hoc networks are communicate through wireless links. Wireless links compared to wire links have low bandwidth and easily be influenced by the environments such as noise and the shield of objects.

- Constraints of mobile devices

  Mobile devices themselves always have less power, less battery and less capabilities.

2.3.2 Routing in Ad-hoc Networks

Ad-hoc networks should maintain the routes between mobile nodes. There are two kinds of routing protocols in ad-hoc networks. One is table-driven routing protocol; the other is on-demand routing protocol. [12]

In table-driven routing protocols consistent and latest routing information to all nodes is maintained at each node. If any changes of the network topology, the update messages are propagated to the whole network for the nodes to maintain up-to-date consistent routing information. Typical table-driven protocols include:

- DSDV: Destination-Sequenced Distance-Vector Protocol
- CGSR: Clusterhead Gateway Switch Routing Protocol
- WRP: The Wireless Routing Protocol

In contrast to table-driven routing protocols, the routes of on-demand routing protocols are not maintained at every node, they are created when required. When a source wants to send to a destination, it invokes the route discovery mechanisms to find the path to the destination. Typical on-demand routing protocols include:

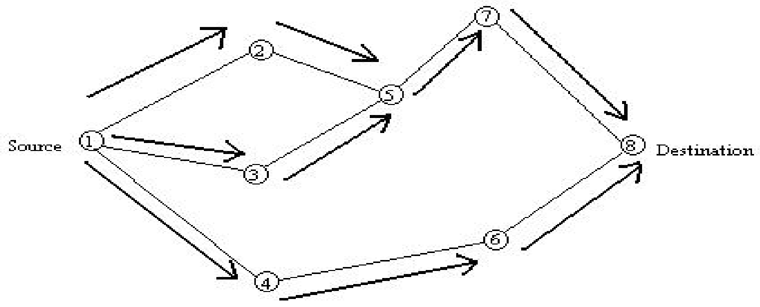
- AODV: Ad Hoc On-Demand Distance Vector Routing
- DSR: Dynamic Source Routing
- TORA: Temporally Ordered Routing Algorithm
- ABR: Associativity-Base Routing
- SSR: Signal Stability Routing

2.3.3 Case Study

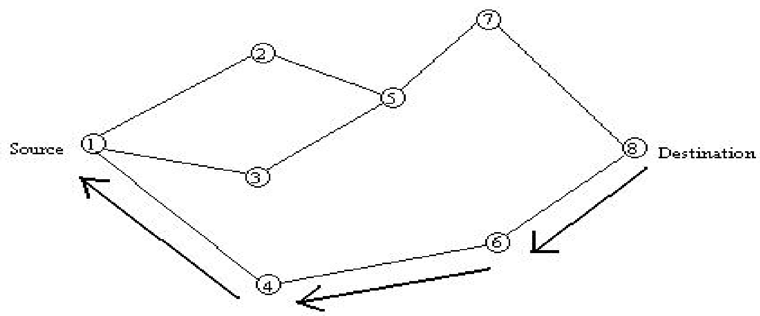
AODV [12]

In AODV, in order to find a path to the destination, the source broadcasts a route request packet. The neighbors in turn broadcast the packet to their neighbors till it reaches an intermediate node that has a recent route information about the destination or till it reaches the destination (Figure 2.3a). A node discards a route request packet that it has already seen. The route request packet uses sequence numbers to ensure that the routes are loop free and to make sure that if the intermediate nodes reply to route requests, they reply with the latest information only.

When a node forwards a route request packet to its neighbors, it also records in its tables the node from which the first copy of the request came. This information is used to construct the reverse path for the route reply packet. AODV uses only symmetric links because the route reply packet follows the reverse path of route request packet. As the route reply packet traverses back to the source (Figure 2.3b), the nodes along the path enter the forward route into their tables.



(a) Propogation of Route Request (RREQ) Packet



(b) Path taken by the Route Reply (RREP) Packet

Figure 2.3 AODV routing protocol [12]

If the source moves then it can reinitiate route discovery to the destination. If one of the intermediate nodes move then the moved nodes neighbor realizes the link failure and sends a link failure notification to its upstream neighbors and so on till it reaches the source upon which the source can reinitiate route discovery if needed.

2.4 Peer-to-Peer Systems

2.4.1 Introduction

Client/Server architecture

Client/server is a traditional architecture of computer networking, illustrated in figure 2.4. [8] Most applications and Internet services are using this client/server architecture. This architecture has been used for decades.

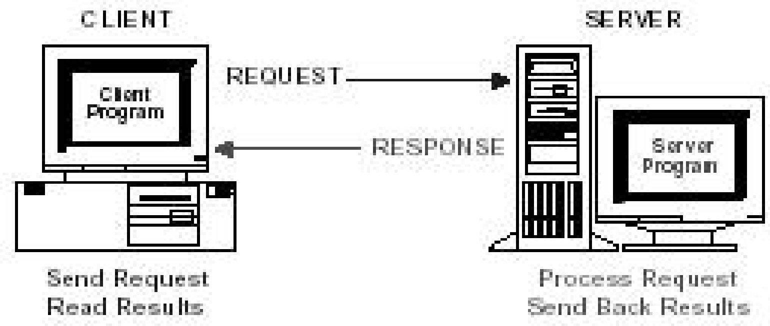


Figure 2.4 the client/server architecture [8]

In this architecture, the clients connect to a server using a specific communications protocol, such as the File Transfer Protocol (FTP), Email, telnet and etc. to request services or information from servers or obtain access to a specific resource. The servers respond by sending the requested information back to the client computers. Most of the processing involved in delivering a service usually occurs on the server, the clients require relatively little computational power. Application functionalities are implemented on the server side such as business logic and data storage. The server side has a high degree of centralization so it is a potential bottleneck, the workload, computational power and bandwidth demands on the server increase with the increase of the number of the clients. And it is too important to have failure. So some fault tolerance technologies are addressed to solve fail-over, such as high availability technologies (for example cluster), load balancing etc. Some other solutions farm out functionalities to the client side such as Java Applet shown in figure 2.5 [8]. It is an efficient way to provide information and services to many users. A network connection is only made when information needs to be accessed.

In client/server architecture, the server side is single point of failure. In case the server is not available sometime, the clients cannot access the resource and services or run the applications on the server no matter how powerful they are.

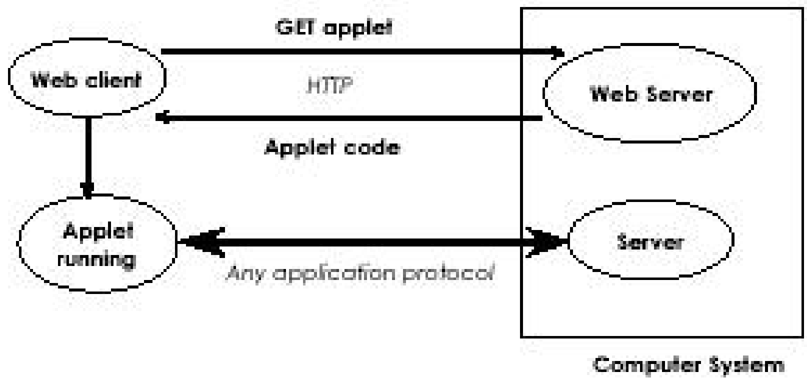


Figure 2.5 Java Applet [8]

Peer-to-Peer architecture

Peer-to-Peer applications came into existence from the early 1980's. The trend of distributed systems and the decentralizing trends in software engineering grow with the Internet. Intersecting the trend is the availability of powerful-networked computers and inexpensive bandwidth. Peer-to-Peer technology filled the need and then grew fast, some famous applications like Napster, Gnutella make this technology familiar to the end users.

Peer-to-Peer is a different architecture compared to traditional client-server architecture. Individual machines provide services to each other in an egalitarian way. Every peer is identical and has equal relationship and role in Peer-to-Peer systems. Unlike the client/server architecture, peer-to-peer systems do not rely on centralized servers to provide access to services or information. As shown in Figure 2.6, a peer-to-peer system is a highly interconnected architecture.

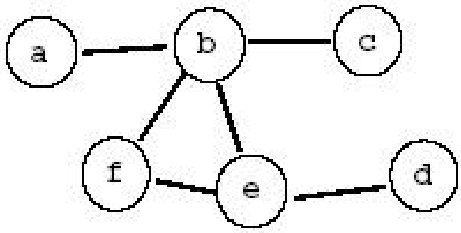


Figure 2.6 the peer-to-peer system

Through interconnected computers detecting each other, peer-to-peer system enables these machines to act as both clients and servers. The main advantage of peer-to-peer networks is that they distribute the responsibility of providing services among all peers on the network. This eliminates the heavy service workload and outage of services due to a single point of failure and provides a more scalable solution for offering services. But the disadvantage is that the peer-to-peer systems are hard to control because of high degree of distribution and anonymity.

In addition, peer-to-peer systems exploit available bandwidth across the entire network by using a variety of communication paths. Unlike traditional client/server communications, in which use specific routes to popular destinations and make the routes overloaded, peer-to-peer systems enables communication via a variety of network routes, thereby reducing network congestion. But the disadvantage is that the response to the request from a client is always non-deterministic, which means that the response may come from different machines through different paths, and is delayed or lost on the way because some machine is not available or disconnected from the network. Many peer-to-peer applications use replication to solve this kind of problems.

Each peer in a peer-to-peer system has the responsibilities both as a client and as a server. As a client, the peer sends commands to other peers to request a service, and receive responses to a request for a service. As a server, the peer receives commands from other peers requesting services, processes services request and executes the requested services, sends response with results of the requested services and propagates requests for services to other peers.

In peer-to-peer environments, when information is distributed, there is no conventional point of failure. Also, peer-to-peer platform are fault-tolerant inherently.

Peer-to-peer technology can also be used to improve security in e-business environments by providing access controls.

Pure peer-to-peer applications are relatively rare, and their utility is still suspect. Peer-to-peer products tend to use a hybrid approach with some sort of central authority.

Most popular peer-to-peer applications have three major categories: file sharing such as Napster, Gnutella, instant messaging such as ICQ, and distributed computing such as SETI@Home.

Other categories include collaboration such as Groove, WorldStreet, distributed search such as NextPage, industry infrastructure such as JXTA project, Jabber.

The more detailed classification of peer-to-peer applications if shown in Figure 2.7 [17].

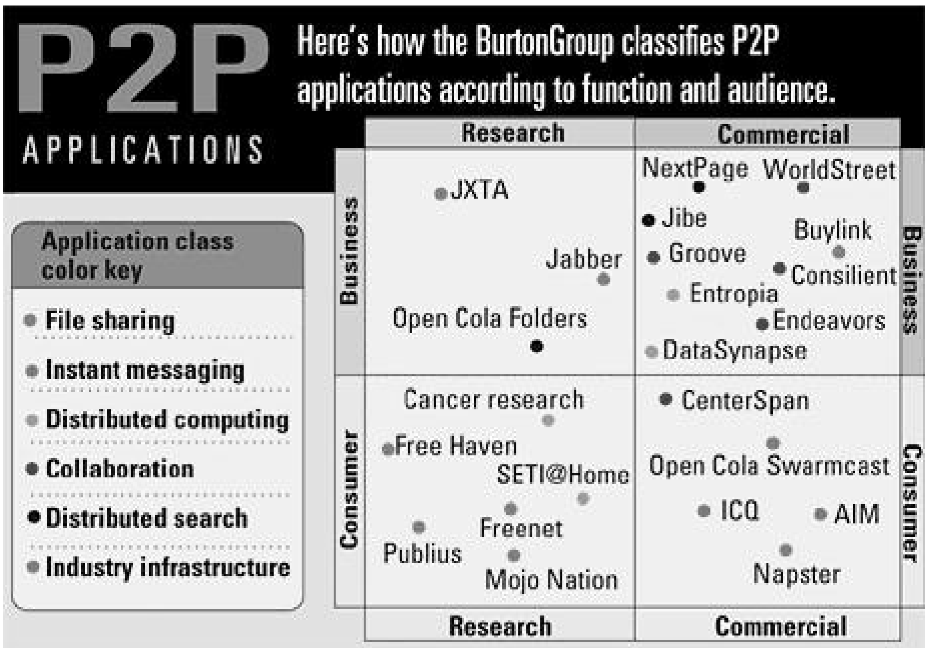


Figure 2.7 peer-to-peer applications' classification [17]

2.4.2 Case Study

Napster

Exchanging music was the first major area of peer-to-peer activity through the Internet. Napster, which began in 1999 created by Shawn Fanning, is an application that allows millions of individuals, sharing and downloading music in MP3 format. The functions of Napster are: a search engine that finds MP3 files on the Internet; a file sharing system for MP3 files without a central server for storing all the files; providing a way of online chat. It is extremely popular and until February 2001, Napster has already 50 million registered users. Caught between massive support and bitter disputes of legal and privacy issues, Napster was finally forced to give in. But other file-sharing networks have emerged. Files such as videos, music and text documents can be exchanged via peer-to-peer applications using decentralized servers. These files exist on individual computer's hard disks and are always a rich collection of resources.

Instead of storing the MP3 files on a central server, Napster use a central server holds a directory of information about clients that are online and the MP3 files they hold, it keeps an index of all the Napster users online currently and connect them to each other. The actual MP3 files are stored in a distributed fashion on clients' computers. When a client want to download a song using Napster, he is downloading it from another online user's machine, and that user may be on anywhere of the world. It is a peer-to-peer file sharing, users contact the Napster index server to locate the files act as clients. But the users act as clients or servers when in dialogue with other users for downloading files.

The scenario of process is shown as Figure 2.8 [8].

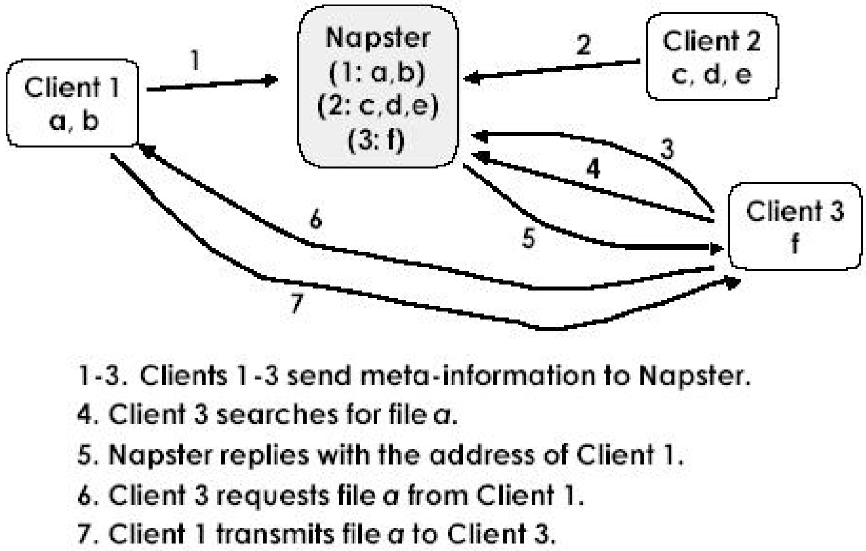


Figure 2.8 a simple Napster scenario [8]

Gnutella

Since the introduction of Napster, more and more similar applications have appeared. And they extend the file sharing not only limited to MP3 files. Napster's successor, Gnutella, adopted a pure P2P file-sharing model, allowing it to avoid the legal difficulties encountered by Napster. With no central server providing services and for index, the Gnutella software provided no target for the recording industry to attack and no easy way to stop the use of the application. Unlike Napster, Gnutella allows sharing of any type of file, not just MP3s. Gnutella program, and dozens of clones of Gnutella, improve the capabilities and performance of the file sharing technology.

Figure 2.9 how Gnutella retrieves information [9]

Gnutella is a decentralized distributed information-sharing technology. It is also a system for searching for information. The protocol for obtaining information over Gnutella is a kind of call-and-response that's more complex than simply pushing news or e-mail.

To use Gnutella, the user need a Gnutella SERVENT (SERVer+cliENT) application that lets he search for, download, and upload any type of file. With a SERVENT, user can connect to others and form a private network or can connect to the general public network.

•Connecting ------the user only have to know one thing: the IP address and port of any SERVENT that is already connected. The first thing the SERVENT does when it connects is to announce its presence. The SERVENT that are connected to passes this message on to all of the SERVENTs it is already connected to, and so on until the message propagates throughout the entire network.

Each of these SERVENTs then responds to this message with a bit of information about itself: how many files it is sharing, how many KBs of space they take up, etc.

• Searching ------ the user sends out a search request, it is propagated through the network, and each SERVENT that has matching terms passes back its result set.

In order to function, peering networks dynamically collect, distribute, and broadcast the IP addresses of their active peers.

• Downloading------when user find a search result that he want to download, he just connects to the SERVENT in the same way the web browser would connect to a web server. SERVENTs are also smart enough to compensate for firewalls.

Gnutella has some limitations. The exponential spread of requests make it is easy to get denial-of-service attacks caused by flooding the system with requests. People may misuse Gnutella for other reasons besides denial of service. And, it is difficult to authenticate the source of the data returned.

Both Napster and Gnutella allow users to control what files they share. Gnutella takes this a bit further than Napster by also allowing users to share different types of files. But Gnutella has no actual "servers". Each piece of Gnutella software is both a server and a client in one. A Napster network is closer to the traditional client-server motif.

.

JXTA

Realizing the need for a common P2P language to allow peers to communicate and perform the fundamentals of P2P networking, Sun Microsystems formed Project JXTA, a small development team under the guidance of Bill Joy and Mike Clary, to design a solution to serve all P2P applications. At its core, JXTA is simply a set of protocol specifications, which is what makes it so powerful. Anyone who wants to produce a new P2P application is spared the difficulty of properly designing protocols to handle the core functions of P2P communication. [10]

The JXTA v1.0 Protocols Specification defines the basic building blocks and protocols of P2P networking:

- Peer Discovery Protocol: Enables peers to discover peer services on the network
- Peer Resolver Protocol: Allows peers to send and process generic requests
- Rendezvous Protocol: Handles the details of propagating messages between peers
- Peer Information Protocol: Provides peers with a way to obtain status information from other peers on the network
- Pipe Binding Protocol: Provides a mechanism to bind a virtual communication channel to a peer endpoint
- Endpoint Routing Protocol: Provides a set of messages used to enable message routing from a source peer to a destination peer

The JXTA protocols are language-independent, defining a set of XML messages to coordinate some aspect of P2P networking. In addition, the simplicity of the JXTA protocols makes it possible to implement P2P solutions on any device with a "digital heartbeat," such as PDAs or cell phones, further expanding the number of potential peers. [10]

The JXTA platform defines a set of protocols designed to address the common functionality required to allow peers on a network to form robust pervasive networks, independent of the operating system, development language, and network transport employed by each peer.

The JXTA Protocol Suite [11]

The Project JXTA team designed a set of six protocols based on XML messages, shown in Figure 2.10. Each of the JXTA protocols addresses exactly one fundamental aspect of P2P networking. Each protocol conversation is divided into a portion conducted by the local peer and another portion conducted by the remote peer. The local peer's half of the protocol is responsible for generating messages and sending them to the remote peer. The remote peer's half of the protocol is responsible for handling the incoming message and processing the message to perform a task.

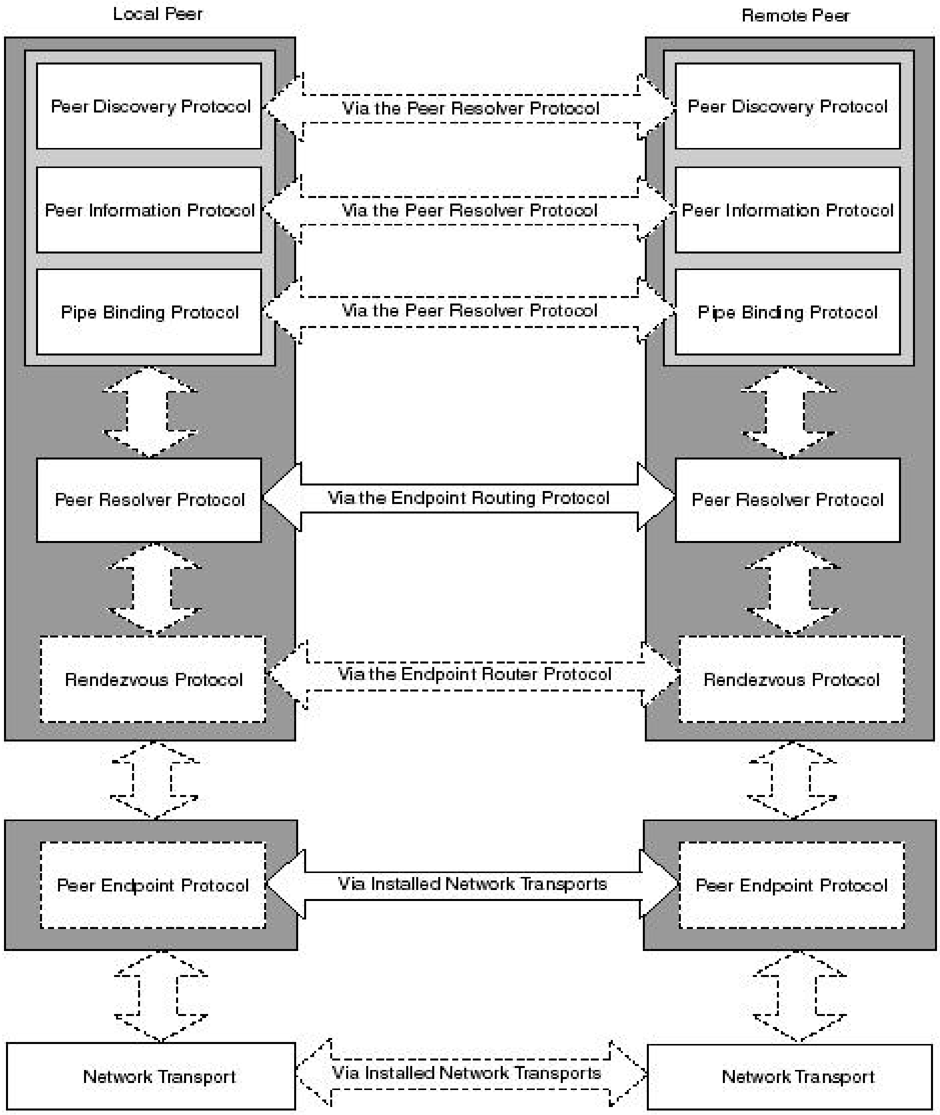


Figure 2.10 the JXTA protocol stack [11]

A peer can elect to implement only a subset of the protocols to provide functionality and each layer in the JXTA protocol stack depends on the layer below to provide connectivity to other peers.

The Logical Layers of JXTA

The JXTA platform can be broken into three layers, as shown in Figure 2.11.

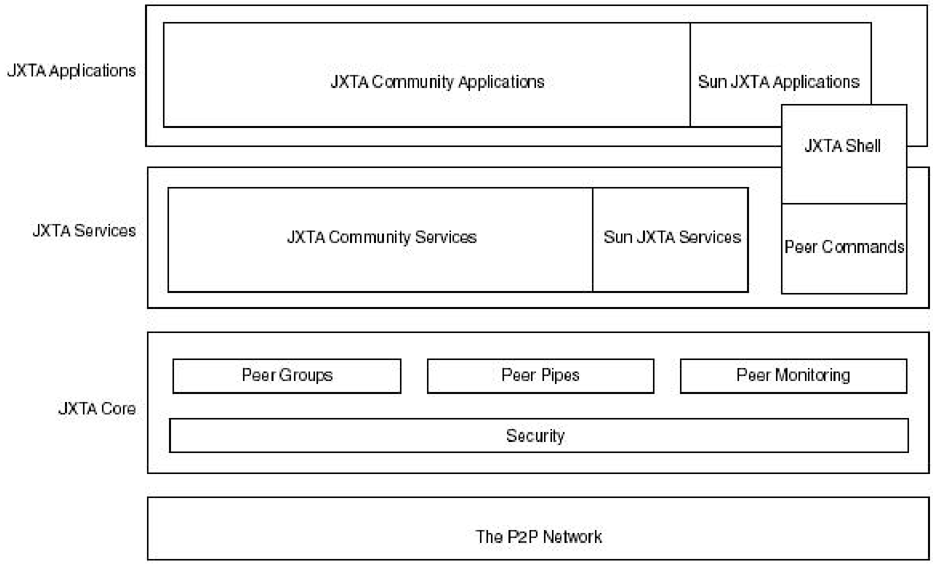


Figure 2.11 the JXTA three-layer architecture [11]

Each layer builds on the capabilities of the layer below, adding functionality and behavioral complexity.

The core layer provides the elements that are absolutely essential to every P2P solution. Ideally, the elements of this layer are shared by all P2P solutions. The elements of the core layer are:

- Peers
- Peer groups
- Network transport (pipes, endpoints, messages)
- Advertisements
- Entity naming (identifiers)
- Protocols (discovery, communication, monitoring)
- Security and authentication primitives

The core layer includes the six main protocols provided by JXTA and is the fundamental core of the JXTA solution. All other aspects of a JXTA P2P solution in the services or applications layers build on this layer to provide functionality.

The services layer provides network services that are desirable but not necessarily a part of every P2P solution. These services implement functionality that might be incorporated into several different P2P applications, such as searching for resources on a peer, sharing documents from a peer and performing peer authentication. The services layer encompasses additional functionality that is being built by the JXTA community (open-source developers working with Project JXTA) in addition to services built by the Project JXTA team. Services built on top of the JXTA platform provide specific capabilities that are required by a variety of P2P applications and can be combined to form a complete P2P solution.

The applications layer builds on the capabilities of the services layer to provide the common P2P applications.

SETI@Home

The universe is an awfully big place. How can people best search the huge sky for a radio signal from ET (extraterrestrial)? One way is to basically eavesdrop on any radio communications coming from beyond Earth. Radio is not only a cheap way of communicating, but also a sign of a technological civilization. Humanity has been unintentionally announcing its presence since the 1930s by way of the radio waves and television broadcasts that travel from Earth into outer space everyday. [16]

SETI (Search for Extraterrestrial Intelligence) is an extremely controversial scientific endeavor. Some scientists believe that it is a complete waste of time and money, while others believe that detection of a signal from ET would forever change our view of the universe.

The universe is an awfully big place. How can you best search the huge sky for a radio signal from ET? Because the sky is so big, there two basic approaches to SETI searches:

- Wide-field search - In this method, people survey large chunks of the sky, one at a time, for signals. A wide-field search allows the entire sky to be searched at a low resolution in a short period of time. However, if a signal is detected, it would be difficult to pinpoint the exact source without a subsequent high-resolution search.
- Targeted search - In this method, people make intensive investigations of a

limited number (1,000 to 2,000) of sun-like stars for ET signals. The targeted-search allows for more detailed investigations of small areas that we think might be probable locations of ET, such as stars with planets and conditions favorable for life as we know it. However, this approach ignores large portions of the sky and might yield nothing if the guesswork is wrong.

In the 1- to 10-gigahertz (GHz) range of frequencies, there is a sharp drop in background noise. In this region, there are two frequencies that are caused by excited atoms or molecules: 1.42 GHz, caused by hydrogen atoms, and 1.65 GHz, caused by hydroxyl ions. Because hydrogen and hydroxyl ions are the components of water, this area has been called the water hole. Many SETI researchers reason that ET would know about this region of frequencies and deliberately broadcast there because of the low noise. So, most SETI search protocols include this area of the spectrum. Although other "magical" frequencies have been proposed, SETI researchers have not reached a consensus on which of these frequencies to search.

Another approach does not limit the search to any one, small range of frequencies, but instead builds large, multichannel-bandwidth signal processors that can scan millions or billions of frequencies simultaneously. Many SETI projects use this approach.

In 1999, University of California at Berkeley scientists Dan Werthimer and David P. Anderson worked on Project SERENDIP. They recognized that a limiting factor in analyzing the data from the Arecibo dish used by SERENDIP was the available computing power. Instead of using one or more large supercomputers to analyze the data, many smaller desktop PCs could be used to analyze small pieces of data over the Internet. They devised a screensaver program called SETI@home that could be downloaded from UC Berkeley over the Internet and reside on a participant's home computer. The program can work in residence or as a screensaver.

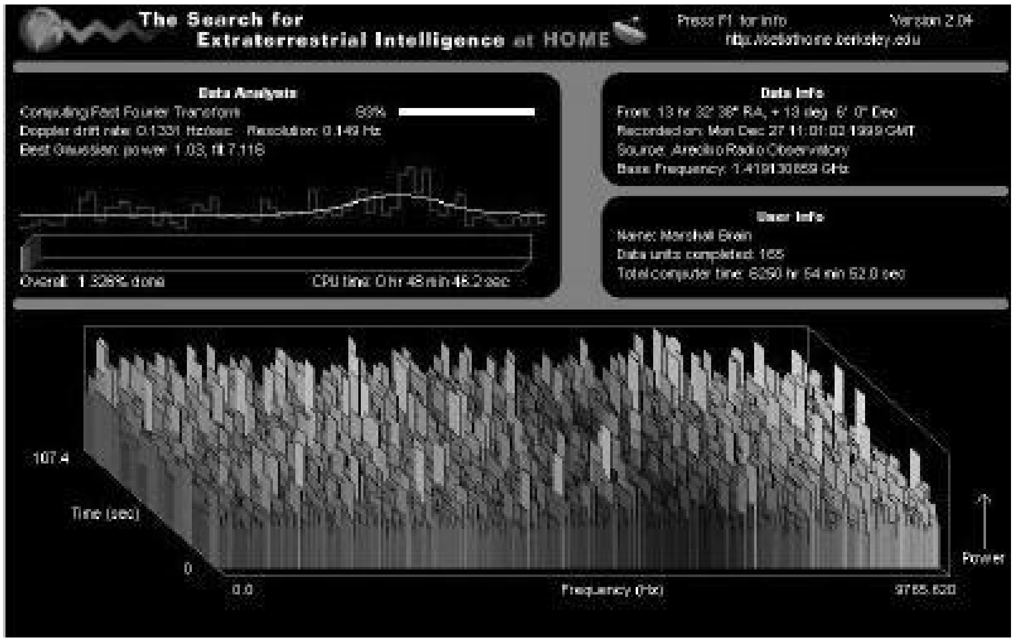


Figure 2.12 SETI@Home screen

Here's how the project works:

1. Data are collected from the Arecibo dish in Puerto Rico, where Project SERENDIP is presently conducted.
2. The data are stored on tape or disk along with notes about the observations, such as date, time, sky coordinates and notes about the receiving equipment.
3. The data are divided into small chunks (approximately 107-second blocks) that desktop PCs can utilize.
4. The SETI@home program on your PC downloads a chunk data from the computer servers at UC-Berkeley.
5. Your PC analyzes the chunk of downloaded data according to the algorithms in the SETI@home program. It takes about 10 to 20 hours to analyze the data, depending on the computer's microprocessor and amount of memory.
6. When finished, your PC uploads its results to the UC-Berkeley servers and flags any possible hits in the analysis.

7. After the upload, your PC requests another chunk of data from the server, and the process continues.

SETI@Home is the most popular and well known distributed computing project on the Net. According to David Anderson, the SETI@Home leader who now works at United Devices, revealed at his speech at the O'Reilly P2P Conference, the project has 2.7 million users in 226 countries. "We've accumulated 500,000 years of CPU time. Our rate of computing is 25 teraflops, which is twice the speed of IBM's ASCII White, the fastest supercomputer in the world). We've analyzed 45 terabytes of data." They handle all this with a staff of three to five people. [14]

2.5 Wireless Mobile Peer-to-Peer Computing

Devices in a peer-to-peer architecture communicate directly to each other instead of using central servers. The traditional approach to computing on the World Wide Web is client/server architecture of which client computers share resources or get services on a central server. In peer-to-peer networks devices connect directly to each other, forming a flexible, decentralized network architecture.

The peer-to-peer model is particularly suitable for mobile computing systems working in ad-hoc wireless network environment where it is necessary to discover resources and dynamically route information through the network. People can organize ad-hoc networks quickly, without depending on existing infrastructure of network or making complex configuration on users' computers.

There are several typical differences between wired and wireless peer-to-peer applications.

- Reliability

  In the wired networks, networking is much stable and dependable. Wireless networking is usually less reliable because it's subject to the frequent changing of electromagnetic wave propagation, interference, and battery exhaustion, less manageable because of no fix network structure. Devices may move arbitrarily.

- Different finding methods

  In wired networks, because networking is comparatively stationary, so all peers can be configured about what peers are available in the network and what the services and resources on each of peers. But in wireless networks, every one of the peers has to discovery other peers itself directly and dynamically.

- Mobility management

  Wireless peer-to-peer applications have to deal with location and topology changes, frequent disconnection/reconnection operations.

- Constraints of mobile devices

  The peers in wireless peer-to-peer applications always have lower power, capabilities and smaller screen size. Wireless links have less bandwidth compared to wired networks.

2.6 Summary

In this chapter we introduce the state of the art of wireless technologies, which boosts the development of mobile computing. Ad-hoc mode is useful for establishing a network where wireless infrastructure does not exist or where services are not required. Peer-to-peer architecture meets the trends of distributed and decentralized software engineering and the availability of powerful networked computers and inexpensive bandwidth. Next chapter we will introduce the background of the design and implementation of a peer-to-peer chat application in ad-hoc wireless network.

Chapter 3

Background

3.1 Scenarios of the Chat Application

Peer-to-peer applications in ad-hoc wireless networks can be used in some situations such as meetings in which persons wish to quickly share information; emergency search-and-rescue operations in battlefield, disaster rescue places; data acquisition operations in places without network infrastructure; places where need to communicate but can not speak out or so noisy that can not hear to each other clearly.

The peer-to-peer chat application designed and implemented in this dissertation can be typically used in some scenarios as described in the following:

"Talking" during the meeting

When several people of one team attend a meeting in their company or in a client side, they have prepared a project to propose in the meeting. During the meeting, from other person's speech, one people find something new or something reminds him that his team's project must be modified. They need to discuss quickly on some important figures or issues to make consistent. But at that time, they cannot whisper or speak out in the conference room; nor can they postpone the delivery to another day. In this situation, they can use the chat application to chat silently on their pocket PCs or laptops without letting other persons' notice and thus make a quick decision.

Chat in the library

In a library, it is not allowed to talk. But when two or more students need to discuss some issues about their assignments or they want to select a time to leave for dinner together, they can communicate through their computers or other devices just like in an Internet chat room without disturbing other students. They need not to depend on any network infrastructure in the library to connect each other. With wireless LAN cards installed on the devices, they can form an ad-hoc network easily.

The chat application can also be used in some other situations. The advantage of the chat application is that the users can join a chat room and always make the application open even when they are moving or want to leave for a rest. The application can accept all messages in the chat room except for the private chat messages between other two users. It can also provide the users the latest member list in the chat room automatically.

The chat application is designed for mobile devices so it is very efficient and the execute file is very small, only 31K. It will not occupy a lot of resources of the device it is running on.

3.2 IP Multicast

Multicast is a receiver-based concept, as described in the white paper of IP Multicast Initiative (IPMI), " receivers join a particular multicast session group and traffic is delivered to all members of that group by the network infrastructure. The sender does not need to maintain a list of receivers. Only one copy of a multicast message will pass over any link in the network, and copies of the message will be made only where paths diverge at a router. Thus IP Multicast yields many performance improvements and conserves bandwidth end-to-end. " [22]

IP multicast is efficient for application that one or more sources need to send to multiple receivers such as video/audio conference. Compared to Broadcast that sends one copy of the message to all nodes of the subnet, multicast only sends the copy to the group who will be interested in the message.

Multicast group is identified by IP address, as described in RFC1112 [24], “IP multicasting is the transmission of an IP datagram to a "host group", a set of zero or more hosts identified by a single IP destination address. A multicast datagram is delivered to all members of its destination host group with the same "best-efforts" reliability as regular unicast IP datagrams, i.e., the datagram is not guaranteed to arrive intact at all members of the destination group or in the same order relative to other datagrams. The membership of a host group is dynamic; that is, hosts may join and leave groups at any time. A host may be a member of more than one group at a time. A host need not be a member of a group to send datagrams to it.”

Host groups are identified by class D IP addresses that have "1110" as their high-order four bits. In Internet standard "dotted decimal" notation, host group addresses range from 224.0.0.0 to 239.255.255.255. The address 224.0.0.0 is guaranteed not to be assigned to any group, and 224.0.0.1 is assigned to the permanent group of all IP hosts (including gateways). This is used to address all multicast hosts on the directly connected network. There is no multicast address (or any other IP address) for all hosts on the total Internet. [24]

In order to support IP multicast, the nodes and the network infrastructure for example routers, among the nodes must be configured to enable multicast. Most implementations of TCP/IP on major operating system platforms support IP Multicast [23]. The operating systems we used in the implementation of the projects such as Linux, Windows CE, Windows XP, Windows 2000 can all be configured to support IP multicast. One advantage of IP multicast is that it is easy to extend scale but not increase the bandwidth or load of the network. For example, more users join a multicast application will not occupy a corresponding amount of bandwidth.

In addition, a single multicast group address may bind to multiple port numbers forms multiple sockets for data streams. So multiple applications may share the same multicast IP address.

3.3 Java on Windows CE

With the development of wireless technology, more and more people equipped with mobile devices such as PDAs. People normally use PDAs (Personal digital assistant) to store addresses, phone numbers, appointments and memo etc. But actually they can do more than these, people can use them access the Internet to send email, or play games and music etc. In addition to the basic functions, there are thousands of applications available for PDAs used in hospital by doctors, used by policemen and workers working in the open air, used by drivers to track locations etc.

Pocket PCs are arguably the most advanced PDAs on the market today. With fast processors, much more add-on options, and the powerful Pocket PC operating system from Microsoft, the Pocket PC has been more and more used as mobile productivity solution. Pocket PC is fast emerging as the platform of choice for enterprises to provide applications for their mobile employees, partners and customers as an extension of their intranets.

The PDA that runs Microsoft Windows CE 3.0 is called Pocket PC. Pocket PCs are bigger and powerful than mobile phones but smaller and less powerful than the desktop and laptop computers. There are many Pocket PC manufactures exist such as HP, Compaq, Casio, Nodia and so on. The Compaq iPAQ is the typical Pocket PC device.

Figure 3.1 Compaq iPAQ H3970

In this project we use Compaq iPAQ H3970. The iPAQ Pocket PC H3900 Series offers more versatility and functionality. The transflective display technology combines transmissive and reflective technologies with higher contrast and color saturation providing crisper images and clearer text. The 400 MHz Intel X-Scale processor focuses on accelerating multimedia and security applications. The iPAQ Backup utility is improved for quicker backup/restore and more options. Power stand by settings has been added to allow users to customize the standby time when the main power is low. Additionally Universal Remote Control for controlling home and office electronics is new to the H3900 series. Another new feature is the iPAQ Image Viewer, this helps to keep your favorite photos with you, and display them as a slide show taking advantage of brilliant new screen. [25]

Java can be used to develop mobile and wireless applications that take advantage of the powerful processors, large memory and graphical capabilities of devices based on Windows CE and the new Pocket PC 2002 platform. There are many products offer Java support on Pocket PC such as PersonalJava of Sun, Jeode of Insignia, SavaJe XE of SavaJe. iPAQ H3970 is preinstalled with Insignia Jeode. Since 2001, Compaq, Sun and Insignia have announced inclusion of Java technology with every iPAQ Pocket PC H3800. Jeode is a JVM for Windows CE device, adding a PersonalJava environment to the device.

Recognizing that "one size doesn't fit all," Sun has grouped its innovative Java technologies into three editions: Java 2 Platform, Micro Edition (J2ME^{TM} technology), Java 2 Platform, Standard Edition (J2SE^{TM} technology), and the Java 2 Platform, Enterprise Edition (J2EE^{TM} technology). Each edition is a developer treasure chest of tools and supplies that can be used with a particular product. [28]

The Java platforms family is shown as Figure 3.2.

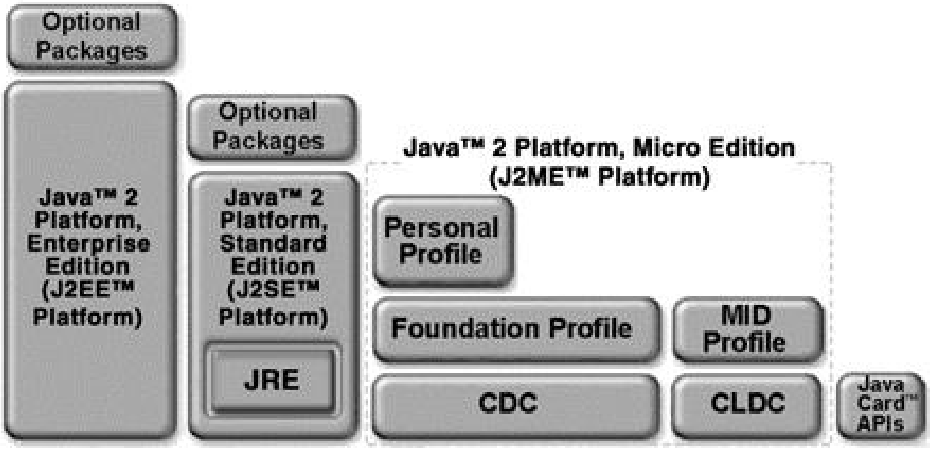


Figure 3.2 Java platforms [28]

Sun Microsystems announced PersonalJava prior to J2ME. PersonalJava defines a different subset of the core Java APIs for use within reduced memory footprint devices such as tabletop screen phones, high-end PDAs and set-top boxes. The next generation PersonalJava environment is becoming the Personal Profile Specification for J2ME. This essentially defines a third J2ME profile for PersonalJava devices that includes a set of device requirements and a set of APIs to expect from libraries. This will keep PersonalJava alive from a product perspective, but will move it into the J2ME space from a marketing perspective. [27]

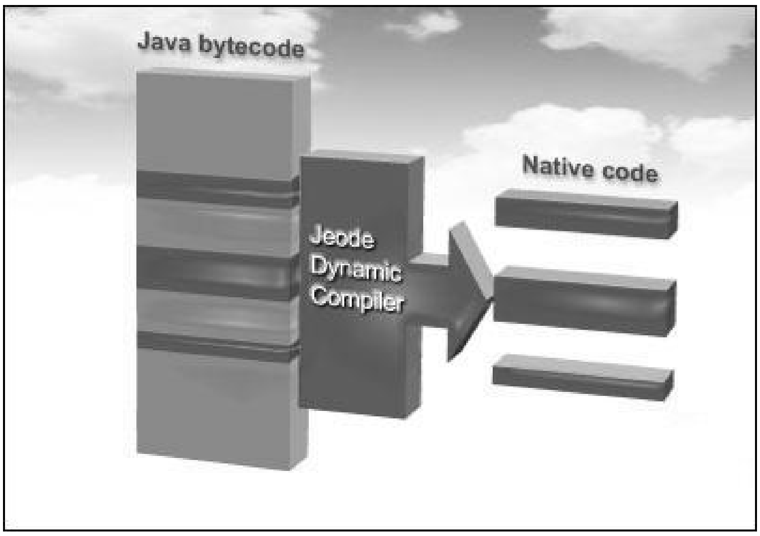


Figure 3.3 how Jeode works [26]

Jeode platform is complete compatibility with the PersonalJava and EmbeddedJava specifications. Jeode Embedded Virtual Machine (EVM) runtime engine was the first independently developed virtual machine to pass the extensive PersonalJava platform and EmbeddedJava technology compatibility test suites, ensuring full compliance with the latest PersonalJava and EmbeddedJava specifications. As a result, the Jeode platform earned the distinction of "Sun Authorized Virtual Machine." [26]

Because Insignia is a Sun Microsystems licensee and adheres strictly to the PersonalJava and EmbeddedJava specifications, the Jeode platform has been certified and is fully compliant with Sun's PersonalJava 1.2 and EmbeddedJava 1.0.3 specifications, which is roughly equivalent to JDK version 1.1.8. The Jeode platform includes all the class libraries in the specifications, including:

- java.applet (PersonalJava only)
- java.awt and sub-packages
- java.beans
- java.io
- java.lang and sub-packages
- java.lang.reflect
- java.math
- java.net
- java.rmi and sub-packages
- java.security
- java.sql
- java.text
- java.util and sub-packages

The java.net class contains the DatagramSocket method and MulticastSocket method to support the unicast and multicast that will be used in the implementation of the chat application.

3.5 Summary

This chapter introduced the IP multicast and Jeode which will be used in the implementation of the peer-to-peer chat application. IP multicast is used a lot in the discovery and message exchange of the chat application. Jeode offers very good support to run Java on Pocket PCs. Some typical scenarios of the using of the chat application are also introduced. The next two chapters will introduce the details of the design and implementation of the chat application.

Chapter 4

Design

4.1 Introduction

The aim of this chapter is to present a design for a multi-hop peer-to-peer chat system in an ad-hoc environment. This system will be designed and implemented as an example to understand and analyze the architecture of mobile ad-hoc peer-to-peer applications, what kinds of problems should be considered and how to solve them.

4.2 Requirements

The requirement of the application is to design a peer-to-peer chat system on mobile devices such as Pocket PCs and laptop computers with wireless LAN cards working in ad-hoc mode.

This chat system has the following features:

- Pure Peer-to-Peer architecture

  In other words, there will no computers that act as any kinds of servers in this system.

  Every peer acts as a both server and client as well as router and must discover the other peers in its local range and multi-hop ranges and maintain the peers list dynamically. Devices communicate directly with each other, without the use of an access point or any other network equipment.

- Ad-hoc network with the challenge of mobility

  Ad-hoc network is useful for establishing a network where wireless infrastructure does not exist or not required. All devices can communicate directly each other with wireless links. In an ad-hoc networks with mobile devices, any of the nodes may move arbitrarily and frequently. Mobility causes a dynamic and variable network topology and there is no infrastructure to depend on, so every peer should detect the location and availability of other peers dynamically. Wireless links are unreliable and easily to be influenced by the environments and get disconnection and reconnection frequently. Mobile devices always have the less power and capabilities than desktop computers. All these characters will be considered in the design of the system.

- Multi-hop Routing

  Wireless networks have a limited range of direct communication. Mobile devices that can communicate directly are called in one range. Otherwise they are called in different ranges. The peers that are not in a peer's local range are called in its remote ranges or multi-hop ranges.

  Each of the mobile devices should know who are also in the chat room no matter whether they are in its local range or other multi-hop ranges.

  No peer will know other peers in advance. They detect other peers themselves.

  The peers in the intersection of more than one range will act as routers to automatically forward messages between different peers.

  When a peer moves out of range and moves back again, it can join the chat again without restarting the application.

- Message Exchanging

  Chat is an instant messaging service system.

  All peers can chat in public or communicate with another peer without letting others know. In the public chat way, the message sent out by one peer will be shown to all other peers.

  In the private chat way, the message is only sent to the peer chosen by the sender.

- People may create different chat rooms using this application.

4.3 Overview of Design

According to the requirement, the overview of the design can be shown through the use case diagram in Figure 4.1.

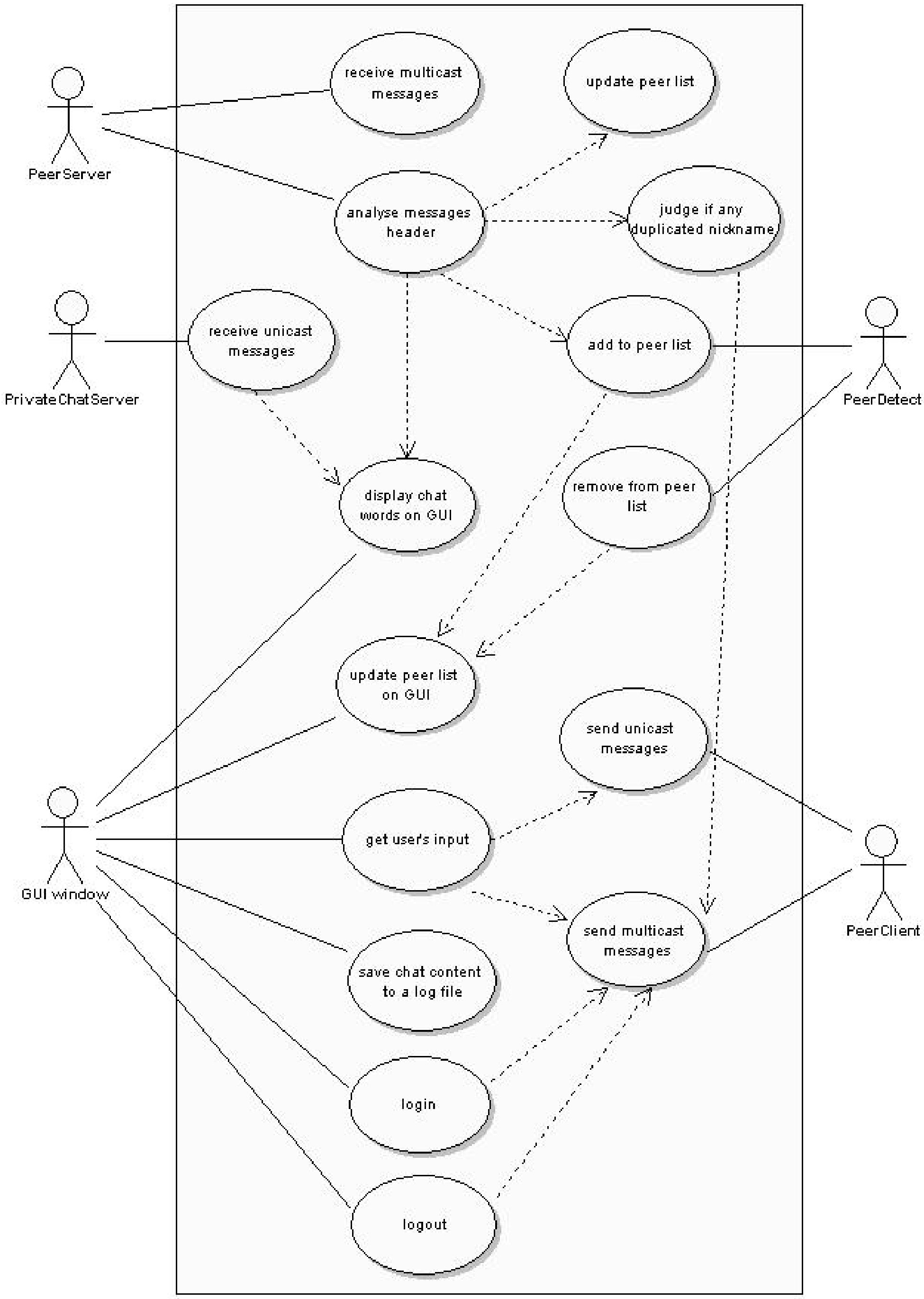


Figure 4.1 the use case of the chat application

According to the requirements of this project, it can be designed and implemented by taking advantage of the network features themselves like IP multicast, unicast and network socket.

The system is designed to use five components to implement all the functionalities. They are PeerServer, PeerClient, PeerDetect, PrivateChatServer, GUI. The use case shown in Figure 4.1 shows how these components interact together. More details about each of the components are discussed in Chapter 5.

The main five design issues implemented in the peer-to-peer chat system are:

- Egalitarian peer
- Auto-Self-Discovery
- Multi-hop routing
- Message exchanging
- Different chat rooms

The details about each of these five issues are discussed in the following.

4.3.1 Egalitarian Peer

A typical scenario of an ad-hoc peer-to-peer environment is described in Figure 4.2:

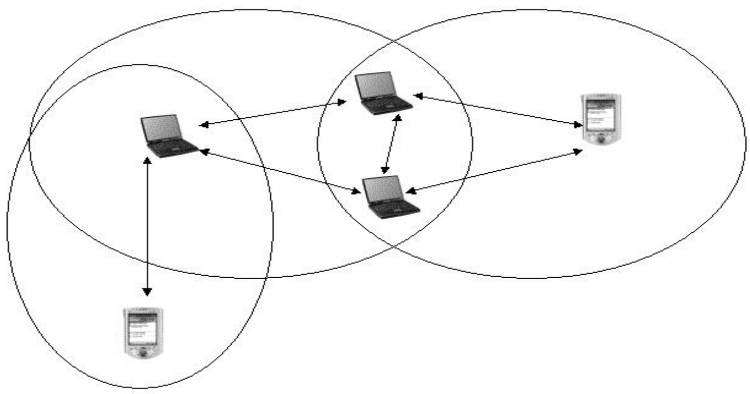


Figure 4.2 peer-to-peer computing in ad-hoc network

In this pure peer-to-peer chat system, every peer needs to make requests for connections with other peers and send messages to them. At the same time, it needs to respond to the requests for connections and receive messages from others. Finally, each peer has to forward messages because there is no specific router in the network. It is similar to Gnutella. Gnutella SERVENT (SERVer + cliENT) application lets users search for, download, and upload files. In the chat system, each peer integrates the function of server, client and router simultaneously.

As a server –

- Listening on socket to receive messages

  Every peer sets up two sockets to listen to connections for messages coming from the network. One is a multicast socket used to receive multicast messages sent to the multicast group with the same IP address and port number of the peer. The other is a unicast datagram socket used to receive messages sent to it only.

  As shown in Figure 4.3, peer C opens two sockets waiting for connection requests and receiving messages. One is for multicast. Another one is for unicast.

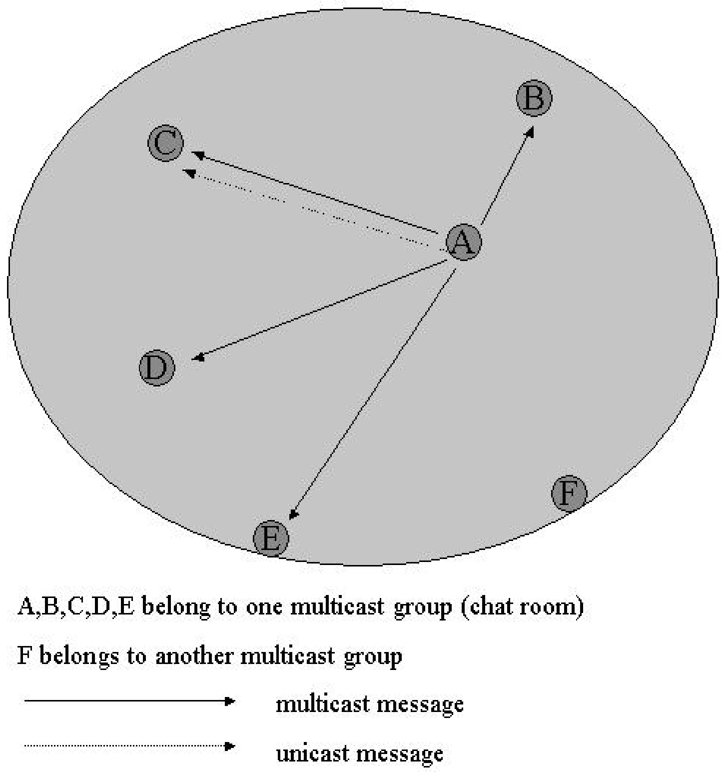


Figure 4.3 messages transfer

- Analyzing messages

In the chat system, all information is transferred through messages. Different types of messages carry different kinds of information. There are two types information. One is network status information dealing with peer status and membership. Another is chat information carrying chat content. There are six types of messages used to maintain the network status and get chat words for each peer in the network. Each message has a message header to identify its type. As a server, the peer will analyze every message received and carry out the corresponding operation.

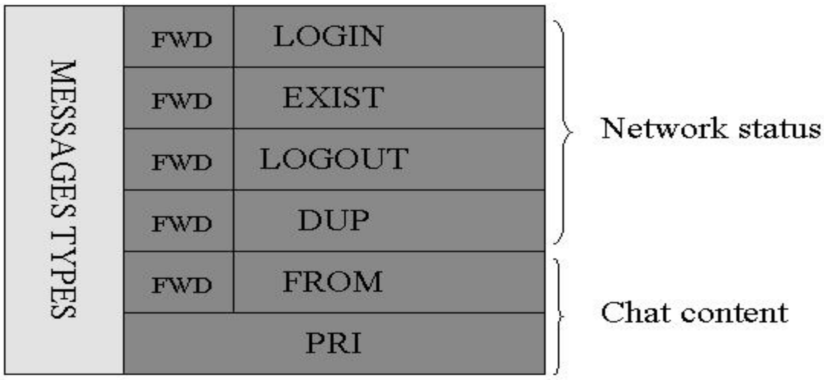


Figure 4.4 messages types of the chat application

More details of the six types of messages are discussed in the part of 4.3.4.

- Maintain network status

  As a network application, it is important to maintain the network status correctly. For an ad-hoc network application, all nodes can move independently, frequently and arbitrarily. So it is necessary to track the network topology dynamically such as how many peers exist in the network? Which peer newly joined the network? Which peer left the chat room? Which peer became not available? Finally, because this is a chat system, this information should be known by all nodes almost real-time, or we say, with very short time delay.

  This is implemented by Auto-Self-Discovery, which will be discussed in part 4.3.2.

As a client –

- Announcing existence

  The fist thing after a peer comes into the chat room is to announce its existence to the members in the chat room (multicast group).

  It announces its existence periodically so that every peer knows that it is alive and available. In the chat system, the interval is 2 seconds. How long the interval needs to be is a trade-off. A short period of time can detect the network changes almost in real-time, but sends out many more messages. Long periods of time can have better performance but may not respond to changes of the network very quickly. Through several tests and balance, for this ad-hoc chat system, several seconds is ideal. A peer should detect another peer is still alive or not frequently so that it will know the changes of network in time. A peer as a client should announce its existence several seconds and it need time to propagate this message to the entire network.

- Request for connection and send out messages

  As previously described, a peer needs to send out messages to show its status as well as chat words obtained from the GUI components, where the users have input them. Most of the messages are multicast messages sent to every member of the chat room (group). Other messages are unicast messages of private chat. When a peer needs to communicate with others, first, it sends out a request for connection to a network socket, once a connection is set up then it starts to send messages.

As a router –

- Forwarding messages to neighbor peers

  In this chat system, routing is implemented through message propagation throughout the entire network.

  In order to propagate messages to all peers in the whole multi-hop network, each peer forwards every multicast message it receives.

- Using unique sequence number to avoid loop

  Each message has a unique sequence number to identify it and avoid loop. More details will be discussed in part 4.3.3.

4.3.2 **Auto-Self-Discovery**

Finding and discovery is one of the most important components of peer-to-peer applications. Most peer-to-peer applications are not pure peer-to-peer systems. For example Napster, which has a central server as a user register server. List of peers as well as resources on them can be obtained from the server. As some other peer-to-peer systems, for example in a wired network, all the peers are stationary. So it is easy to know which peers are in the network and their locations in advance. So pre-configured files can be set on each of them. They need not to discover other peers themselves.

In a pure peer-to-peer system in a wireless ad-hoc network, it is not possible to know in advance who will take part and when. Each node has to directly discover others automatically and atomically.

The most feasible way is to use IP multicast to discover other peers. Because IP multicast only sends a single message to a selected group of the network instead of sending each of the receivers a copy of the message, it saves bandwidth especially in a wireless LAN. Finally, because of the uncertainty of how many peers will log in to the system, a network protocol like IP multicast is suitable, because the sender will send messages to every member having joined the multicast group no matter how many nodes there are and where they are.

The peers should discover:

- A peer joining the chat room.
- A peer exiting the chat room.
- A peer that is not available any more.

  It may be because of moving out of range, or there may be a network problem.

- Peers that already exist in the chat room.
- Whether a peer is in its local range or in a remote range.

  This can provide the user the change of his network position and relationship with other peers.

Discovery Mechanism:

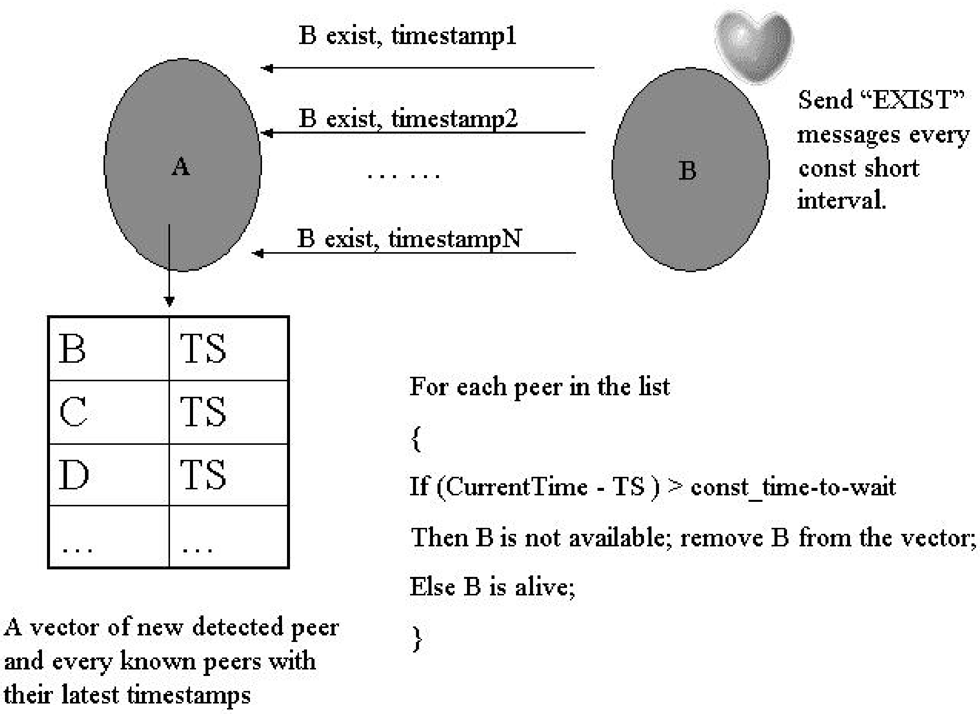


Figure 4.5 peer's discovery and maintenance mechanism

- Use of timestamp

  In order to trace every other peers' status, a timestamp is used. Each peer sends out "Exist" messages periodically with current local time.

  The peer who receives this message will update the timestamp of the sending peer.
- Heartbeat message based mechanism

  Each peer constantly sends out "EXIST" messages to announce its existence.

At the same time, each peer maintains a vector of known peers and their latest timestamps.

On receiving a message with a new peer name, it is inserted into the vector with the peer name and the timestamp.

On receiving a message that is already in the vector, then the peer's timestamp is updated.

There is a detect process always running in the background. This process periodically checks every peer's timestamp. If it is old enough, then the peer must no longer be available. The peer's name will be removed from the vector.

4.3.3 Multi-hop Routing

The routing protocol used in this chat system is a propagation multicast routing protocol. Each peer maintains two route tables for all available members in the chat room. One is for local range peers. One is for remote range peers. The route tables contain the IP address of each member.

Each peer maintains route tables locally and independently. Considering the characteristics of ad-hoc networks, it is hard and not worth to maintain routing information other than locally.

- High mobility
- Dynamical topology
- Any peer can be end-node as well as router

This is also a simple routing protocol. Because we have to keep in mind the characteristics of mobile computing which has low bandwidth with mobile devices that have limited computing power and capabilities.

Multicast protocol is used because of the following considerations:

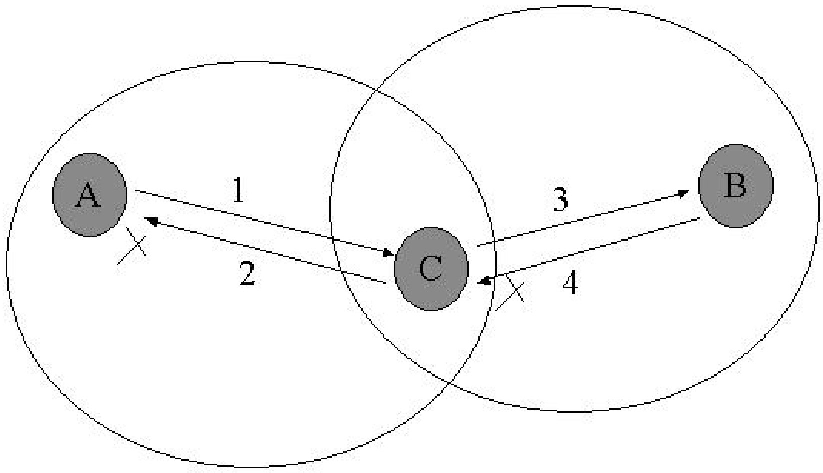
- Multicast protocol is efficient.
- Messages sent to the multicast group can be transparently routed to all

members of the multicast group.

- Multicast is suitable to the group-oriented application
- Multicast is a feature of network itself and is supported by almost all platforms

.

As shown in figure 4.6, Messages 1, 2, 3, 4 have the same sequence number, they are actually the same message just sent out or forwarded by different peers. In order to be loop-free, each peer has a list of sequence numbers of which messages it has sent out or forwarded.



A send message 1 to C.

C forwards the message to A(2), B(3)

A discards message 2 because of the same sequence number as message 1.

B accepts message 3, and forwards to C.

C discards message 4 because of the same sequence number as message 1.

Figure 4.6 routing mechanism

The existing ad-hoc network routing protocol such as AODV cannot be used in this chat application. Firstly, the protocols like AODV do not maintain the information of the list of all nodes in an ad-hoc network so a user of the chat application cannot get the member list in the chat room. Secondly, AODV does not support multicast, which is used a lot in the application. So we use the multicast propagation routing protocol described in Figure 4.6.

4.3.4 **Message Exchanging**

As shown in Figure 4.4, in order to meet the requirements of the application architecture and system, six types of messages are used in this application:
(SN: sequence number)

A) LOGIN message:

Message header		Nickname of the peer
LOGIN	**SN**	**A string of letters and digits**

When a peer first logs in to the system, it sends a multicast "LOGIN" message to existing peers, so everyone knows there is a new peer coming to the chat room. If a peer doesn't log in, it can neither send nor receive messages. Because every peer maintains its own peer list, the information about the new peer will be added to everyone's peer list. But how does the new peer know how many peers already exist? That depends on the "EXIST" message.

B) EXIST message:

Message header		Nickname	Timestamp
EXIST	**SN**	**A string of letters and digits**	**The current system time**

Because there is no registration server for peers, each peer has to maintain their own peer list dynamically. This means that one peer must know whether other peers are active or not, even if they send no messages but just listen. Secondly, a late joining peer needs to know how many peers belong to the system and their information. Thirdly, one peer must announce its existence to other peers and detect if other peers are still active and can be communicated with. All peers must update their peer list periodically and dynamically. A peer may move out of range or have network or system problems so that it is not available to others. When it is available again, no matter whether it recovers from the fault or moves back into the range again, all the other peers will detect it and obtain its new status in a certain amount of time. This message will also provide some security, as anyone who has logged into the system and starts to listen to others chatting will be known by other peers, no matter whether he "speaks" or not.

C) FROM message:

Message header		Nickname	Chat words
FROM	**SN**	**A string of letters and digits**	**Any String**

This type of message is used to transfer chat data in multicast messages. After receiving such a message, the chat contained in the message will be displayed on the GUI component along with who has sent these messages.

D) DUP message:

Message header		Nickname of the peer
DUP	**SN**	**A string of letters and digits**

Again because there is no server for the system to maintain information centrally, there is always a possibility that two peers use the same nickname when they login. The latest one to use the nickname cannot detect that duplication in time, but, after all the peers exchange information, they will realize that. It is hard to avoid this kind of thing happening, but it can be detected and all the peers informed and a warning message displayed on all peers' GUI windows.

As discussed previously in chapter 3, the usage scenarios involving this system are happening to people who normally have known each other and just need to communicate in situations that there is no suitable infrastructure or where it is not suitable to talk aloud. The possibility of nickname duplication will be low. Another method to avoid the duplication of nicknames is to use the computer name as the nickname, but many people use elusive name as their computer name and it is hard to match the name with the person.

E) LOGOUT message:

Message header		Nickname of the peer
LOGOUT	**SN**	**A string of letters and digits**

Peers use this kind of messages to log out of the system. A multicast message will be sent to all peers to inform them about a peer's exit.

F) Private chat message to remote peers:

Message header	Nickname	Nickname		Chat words
PRI	**Source**	**Destination**	**SN**	**Any String**

This type of message is used in private chat situations, when a peer only wants to talk with a remote peer and doesn't want its chat words displayed on everybody's GUI window.

The message is sent out through multicast, until the destination peer get it.

The private chat between two peers that are not in the same range cannot use unicast because they cannot communicate directly. So multicast is used to forward a private chat message to the destination that can accept it through analyzing the "destination" information in the message.

G) Private chat message to local peers:

Message header	Chat words
Nickname of the peer	**Any String**

It is a unicast message. Because every peer maintains a route table of every other peer's IP address, so in the local range, this type of message is sent just to the datagram socket of the destination peer.

4.3.5 Different Chat Rooms

Different chat rooms can be established by using different multicast addresses and port numbers. Users can input different valid multicast addresses and port numbers when they log into the application.

The valid range of multicast address has been discussed in Chapter 3.

4.4 Functionalities

The term "peer" in this thesis means any mobile device, like a laptop computer or PDA, a Pocket Pc running an instance of the peer-to-peer chat application, can communicate to each other directly or indirectly using ad-hoc mode network interface.

As described previously, it is a pure peer-to-peer architecture, so every peer should support functionalities as following:

1) Multicast Messages Receiver

   Join a multicast group and listen to the socket.

2) Unicast Message Receiver

   Listening to the unicast socket.

3) Messages Sender

   Get information from GUI and send the messages to the multicast group or a specific peer.

4) GUI
   - Join a chat room.
   - Exit a chat room
   - Log the chat content to local files and load logs to display
   - Receive user input
   - Show chat room members and chat contents to the user
5) Detector

   To check those known peers are still available.
6) Messages Analyzer
   - Discovery of new joining and leaving peers
   - Update peers' latest information
   - Update GUI information to user.

As shown in the use case diagram, these six functionalities can be implemented with five components that will be discussed detailed in Chapter 5. The relationship of the functionalities and system components are as following:

Functionalities	System Components
Multicast Messages Receiver	PeerServer
Messages Analyzer	
Unicast Messages Receiver	PrivateChatServer
Messages Sender	PeerClient
GUI	GUI
Detector	PeerDetect

4.5 Summary

In this chapter, the requirements and the main system design issues were discussed. The peer-to-peer chat application was designed in order to deepen the understanding the architecture and problem issues in a pure mobile ad-hoc peer-to-peer circumstance. The requirements set out in part 4.2 have been fulfilled in full. The system has been divided into six main components, and in the next chapter, the detailed implementation will be discussed.

Chapter 5

Implementation

5.1 Introduction

The functionalities and components of the peer-to-peer chat system have been discussed in Chapter 4. Mobile devices can be Pocket PCs, laptop computers, handheld computing devices etc, which use wireless LAN cards working in peer-to-peer workgroup mode to compose an ad-hoc network. Since these mobile devices may use different operating systems like Linux, Windows CE, windows and so on, Java is a good choice for development, because Java is platform independent and "Write once, run everywhere" [29].

The five components of this application are as shown as Figure 5.1.

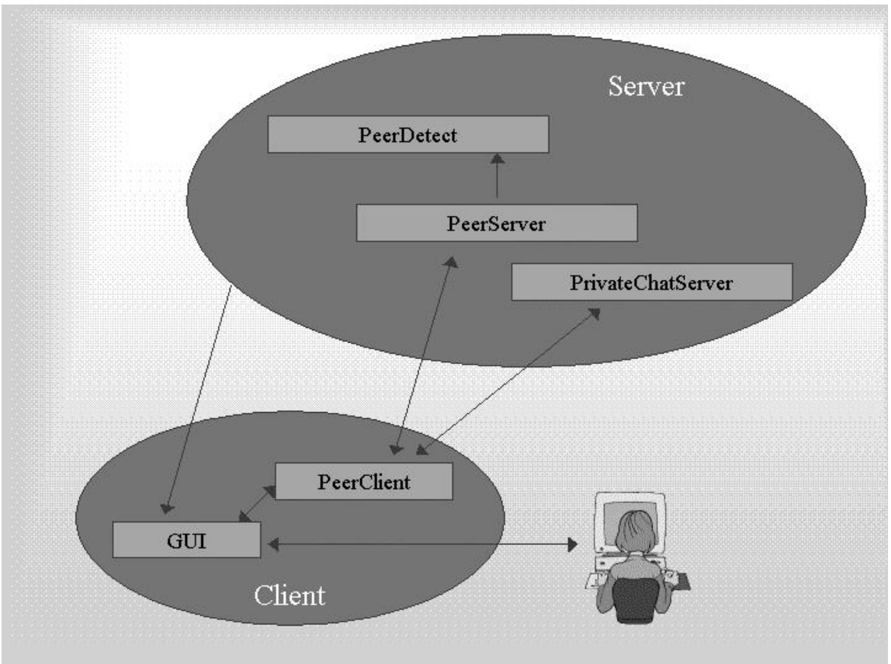


Figure 5.1 components of the chat application

These five components are implemented through two java classes. The Server class contains three inner classes as threads to implement the functionalities of PeerServer, PeerDetect, and PrivateChatServer. The Client class implements the graphical user interface with an inner class PeerClient running as a thread. As shown in Figure5.1, the GUI is the only component that interacts with users. The relationship of the GUI and the other four components is shown as Figure 5.1 and Figure 5.2. After login to the system, the other four components start to run as threads. When the “Logout” button is pressed these four threads will stop and the GUI is closed. Using threads to implement the four components is dependent on their functions being independent; they can execute concurrently to be more productive and they need to execute concurrently to act in the roles of both server and client.

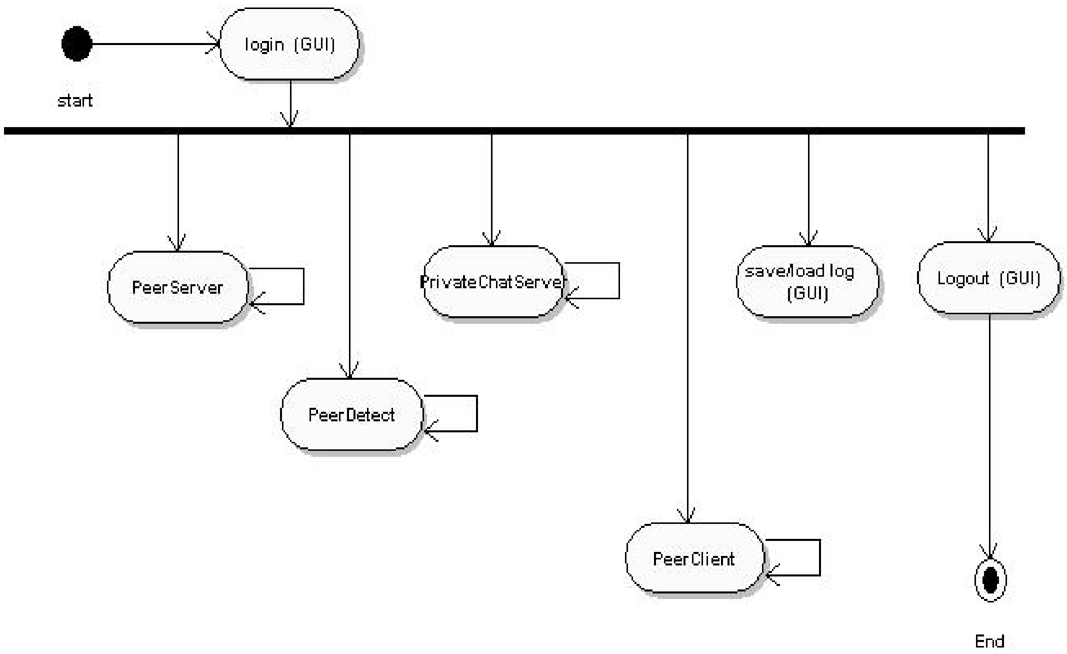


Figure 5.2 the activity diagram of the components

The details of the five components are discussed as the following.

5.2 Main Components

5.2.1 PeerServer Component

The PeerServer component is the most important and complex component of the application.

In this component, two of the six functionalities of the application are implemented.

➢ Multicast Message Receiver

As shown in Figure 5.4, once the user logs in to the chat room, the PeerServer thread starts running. It creates a multicast socket and starts to listen to the socket. Every peer in the chat room has a socket with the same multicast IP address and port number. A peer must join the multicast group so that it can receive every multicast message sent to the group.

➢ Messages Analyzer

What the PeerServer will do depends on what type of message received. The types of messages have been discussed in part 4.3.4 of Chapter 4. From the figure 5.3, the main two operations of a PeerServer are:

- Update the peer list to user

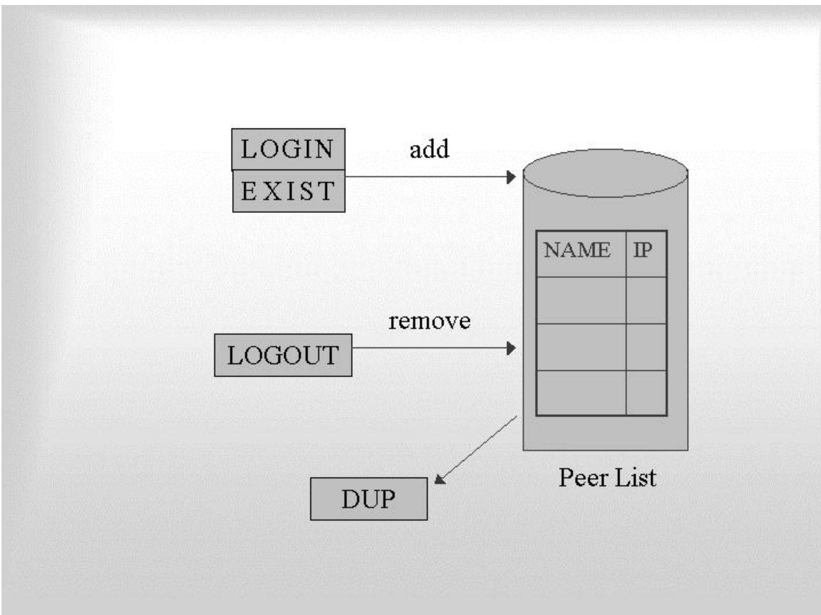


Figure 5.3 how PeerServer maintains the peer list

When PeerServer receives a “LOGIN” message, it means a new peer has entered the chat room. The peer’s nickname and IP address will be added to the peer list.

When PeerServer receives an “EXIST” message, it means that a peer exists in the chat room. PeerServer then transfers the information to PeerDetect component to see if this peer is already in the peer list. If it is not, the peer’s nickname and IP address will be added to the peer list. The PeerServer component will also detect if any two peers have the same nickname but different IP addresses, which means their nicknames are reduplicated. PeerServer will then send out a “DUP” alert message.

When PeerServer receives a “LOGOUT” message, it means that a peer has exited the chat room. The peer’s information will be removed from the peer list.

- Display chat words to user

If the header of a message is “FROM” or “PRI”, PeerServer will pick up the chat words and send to GUI component to display to user if appropriate.

“PRI” is used for private chat that two users are not in the same range. It is a multicast message but not unicast, because these two peers cannot set up connection and communicate directly.

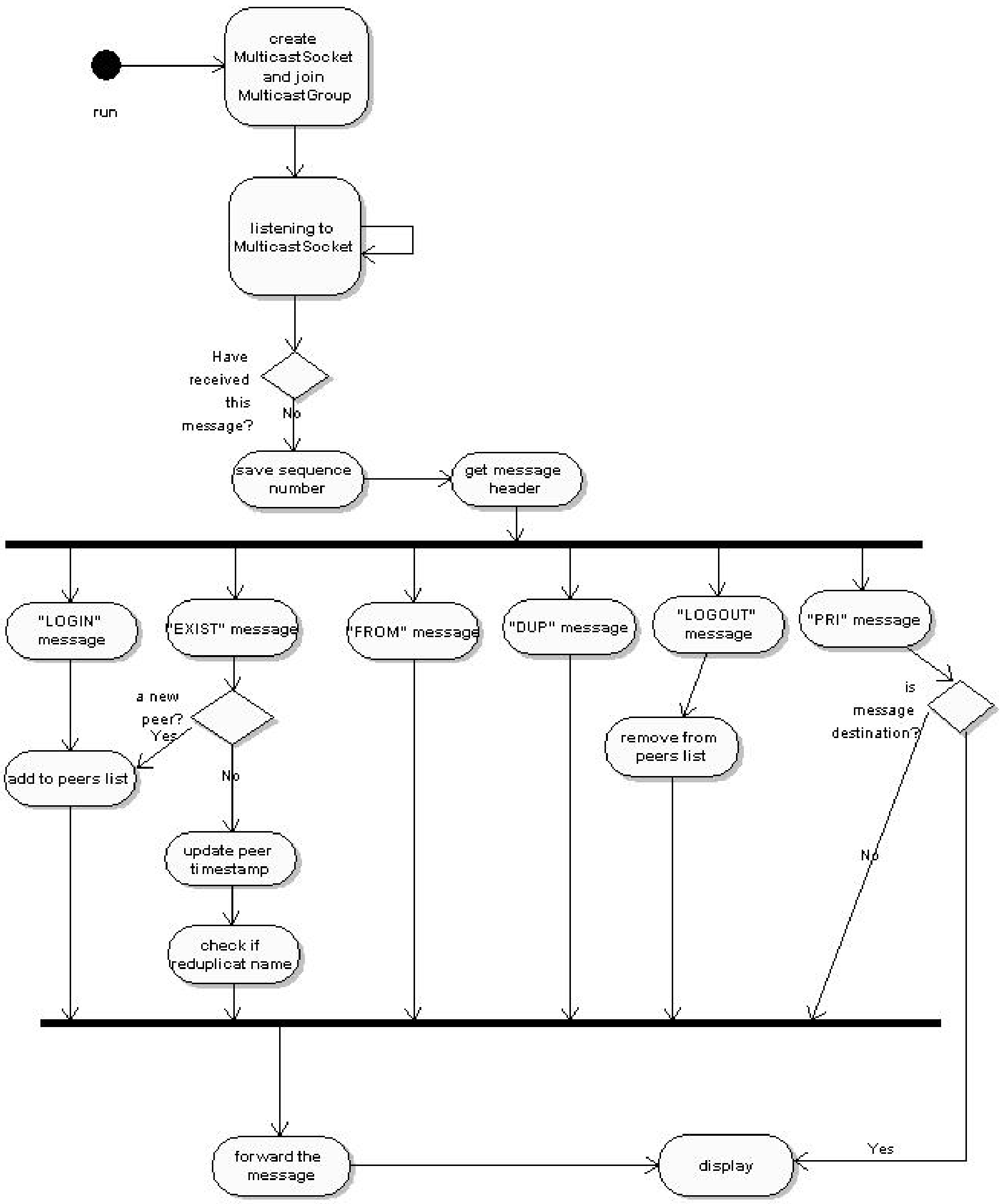


Figure 5.4 the activity diagram of the PeerServer component

5.2.2 **PrivateChatServer Component**

The function of the PrivateChatServer component is to receive private chat messages coming from another peer. So it will know that another peer won't want to have the chatting contents displayed on every peer's GUI window.

The PrivateChatServer component implements one functionality of the application:

the “Unicast Messages Receiver”.

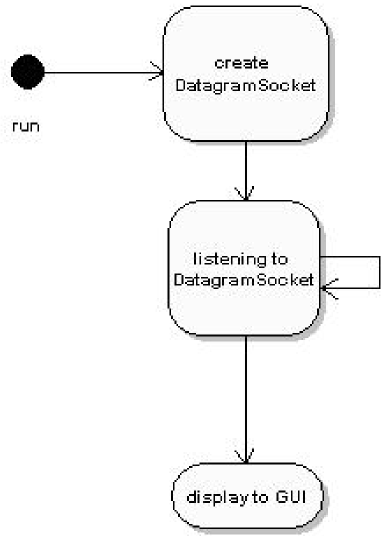


Figure 5.5 the activity diagram of the PrivateChatServer component

5.2.3 **PeerDetect Component**

The discovery mechanism of this peer-to-peer chat system has been discussed in detail in Chapter 4.3.2. The PeerDetect component maintains a vector of nickname and latest timestamp. When PeerDetect gets a new “EXIST” message transferred from PeerServer, it gets a new timestamp of a peer. In this way, the vector is always refreshed. The process is shown as Figure 5.6.
This component implements the functionality of “Detector” of the application.

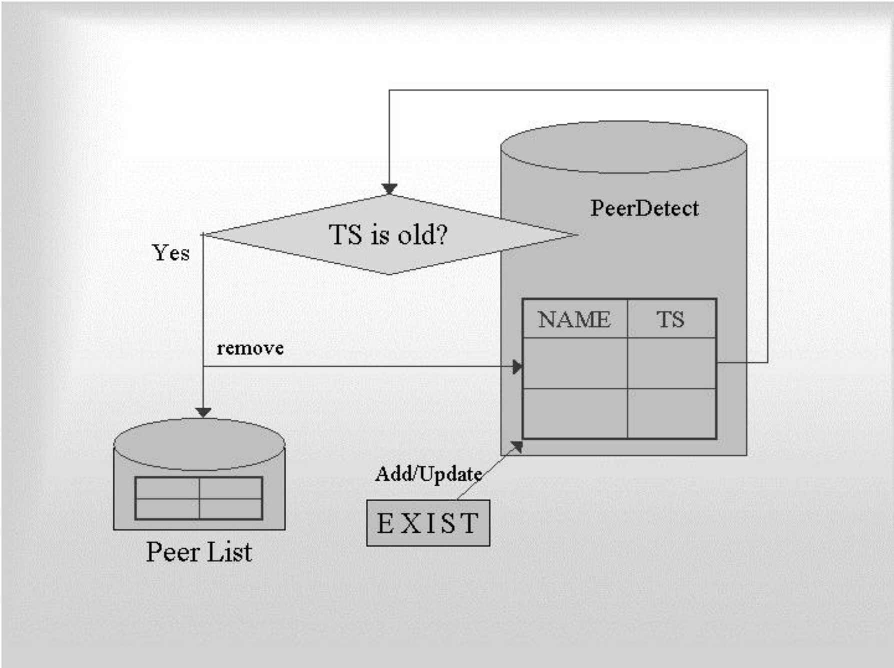


Figure 5.6 how PeerDetect works

The main function of the PeerDetect component is to continuously check every peer in its vector if it is alive using the algorithm described in the following.

```
For each peer in the list
{
If (CurrentTime - TS ) > const_time-to-wait
Then B is not available; remove B from the vector;
Else B is alive;
}
```

Actually, the PeerDetect component maintains two vectors in the application, one is for local range peers, and one is for remote peers. Two vectors can provide users the information about their locations and the relative relationship with other peers while moving.

The value "const_time-to-wait" used in the application for local peers is 20 seconds and for remote peers is 50 seconds.

The value "const_time-to-wait" used in the application for the local vector is 20 seconds. It means that if has not received messages from a peer for 20 seconds, then this peer may no longer be available in the chat room. Because every peer announces its existence every 2 seconds, so 20 seconds is long enough to make sure of its status. The value of "const_time-to-wait" can be adjusted according to the situation of the network. After several tests and comparison, in this ad-hoc peer-to-peer application, we use the value of 20 seconds. The reason to use 50 seconds for remote peers is the same.

5.2.4 **PeerClient Component**

The function of the PeerClient component is to send messages out to the PeerServer and PrivateChatServer components of other peers. The PeerClient component implements the functionality "Messages Sender" of the application.

There are two sources of messages it will send.

- Chat words obtained from the GUI

  Every time the user inputs his chat words in the GUI, the GUI will send them it is send to all members in the chat room or just send to a specific member. As shown in Figure 5.7, if the local-list and remote-list are both "all", then this message should be sent to all members. It is a "FROM" message. Otherwise, this message should be sent to the member whose name is chosen in the local-list and/or remote-list. If it is to a local member, it is a unicast message. If it is to a remote member, it is a "PRI" message.

- Announcing existence messages

  As shown in Figure 4.5, every peer periodically sends out "EXIST" messages to the group to announce its existence. It is similar to the heartbeat of a human. This kind of message will be processed by the PeerDetect component of every other peer.

5.2.5 **GUI Component**

The GUI component is where the user interacts with the application; from where the user inputs his chat words, operates through pressing buttons and gets information about other users' chat words and the members in the chat room. What it looks like is shown as Figure 5.7.

The operations that a user can do are:

- Join a chat room by pressing the "Login" button.
- Exit a chat room by pressing the "Logout" button.
- Log the chat content to a local file by pressing the "Log to file" button.

  As described in Chapter 3, this chat application is useful for people who need to communicate in some certain kinds of situations. So it may be important for them to log the chat content to local file as a memo.
- Load logs to display by pressing the "Load from file" button.

  This is to load a log file's content to the display window to user.
- Send some selected chat content to a specific user by pressing "Send content" button.

  When a user just joins the chat room, he may need to know what people have chatted about. Especially in a scenario like a necessary nonverbal meeting or communication in a certain place, a late joining person may need to catch up with others. Or when a person missed some information because of disconnection, other people may send any selected contents in the display window to him.

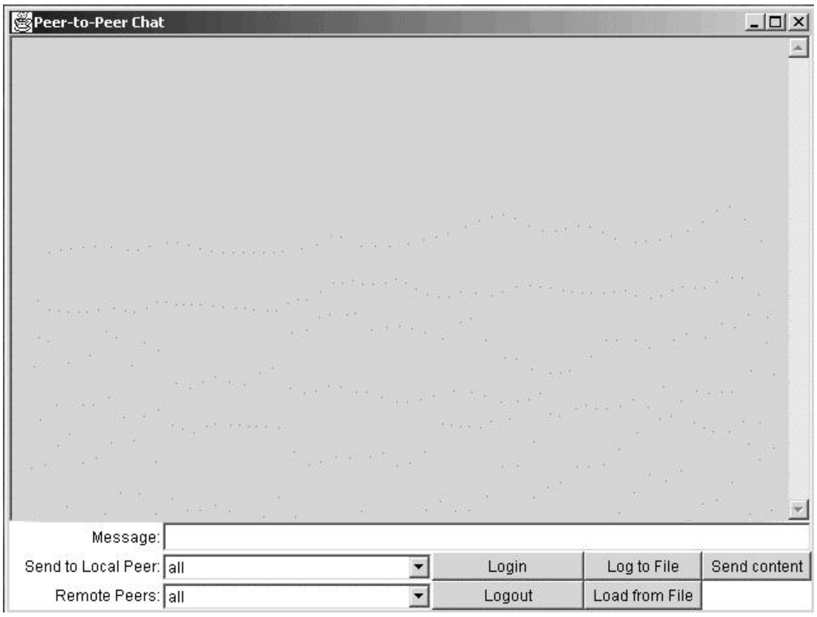


Figure 5.7 GUI of the peer-to-peer chat application

An example of the using of the application is shown as Figure 5.8.

```
center2 comes into the chat room. Welcome! (Fri Aug 30 15:50:43 GMT 2002)

Detected a new peer peter in this chat room! (Fri Aug 30 15:50:45 GMT 2002)

center1 comes into the chat room. Welcome! (Fri Aug 30 15:51:20 GMT 2002)
center1: hi all, this is center1
center2: hi, this is center2

Detected a new peer ipaq in this chat room! (Fri Aug 30 15:52:56 GMT 2002)

ipaq: hi,allthis
center1: hi,all

ipaq has no response, seems has left the chat room... (Fri Aug 30 15:55:49 GMT 2002)
```

Figure 5.8 an example of the chat

5.3 Test

In order to test the design and implementation of the application, several mobile devices were configured and act according to three basic networked scenarios.

5.3.1 Environment

The mobile devices used in the test are laptop computers and Compaq iPAQ Pocket PC.

	Operating system	JVM compatibility
Laptop computers	Red hat Linux Windows XP Windows 2000	Sun JDK 1.3.1 Sun JDK 1.4 Sun JDK 1.1.8
iPAQ Pocket PC	Windows CE 3.0	Insignia Jeode Personal Java 1.2 JDK 1.1.8

All devices are equipped with wireless LAN cards which compliant with IEEE802.11b standard and work in ad-hoc mode at 2.422GHz frequency with the same network ID.

Figure 5.9 platforms of mobile devices used in the tests

As shown in Figure 5.9, various devices and operating system were used. The only requirement is that they have a Java virtual machine with compatibility to JDK 1.1.8. This is because Personal Java 1.2 is compatible with JDK 1.1.8. Since Java is platform independent and “write once, run everywhere”, the chat application can be easily deployed to a device without any changes. In order to communicate each other, the wireless LAN cards must be compliant with IEEE 802.11b standard and configured to work in ad-hoc mode at 2.422GHz frequency with the same network ID.

The application was packaged to a .jar file that was only 31K. The .jar file was deployed on several kinds of mobile devices such as iPAQ Pocket PC and laptop computers. The application occupied very small resources of these devices.

5.3.2 **Networked Scenarios**

In order to test the functionalities as well as mobility, three different networked scenarios are designed.

- One range test

This test is done using four peers. All of the four peers work in one range.

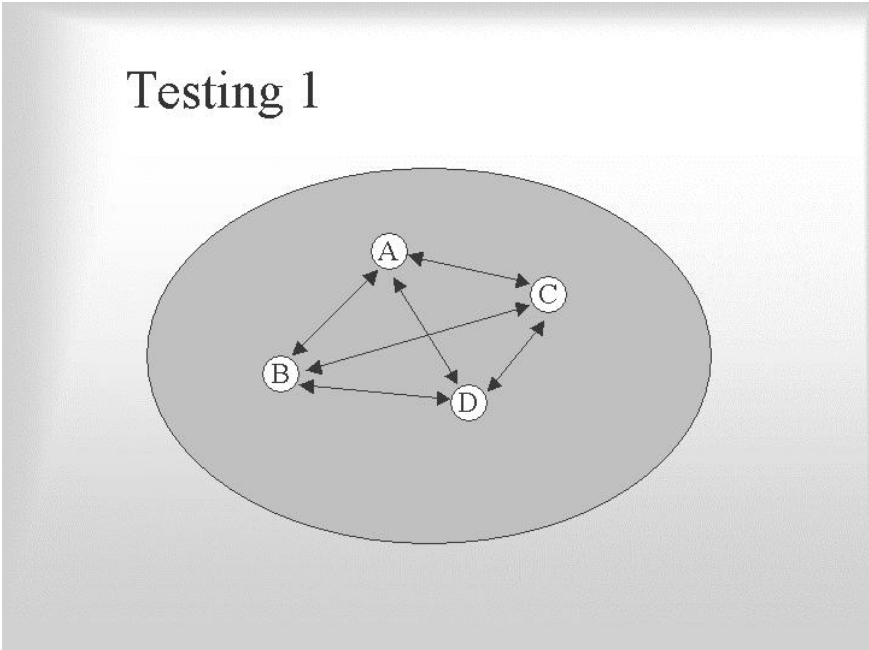


Figure 5.10 one range test

This test is mainly to check the functionalities of the application except for multi-hop routing. Through this test we confirmed that the design and implementation of the pure peer-to-peer architecture works well.

- Two ranges test

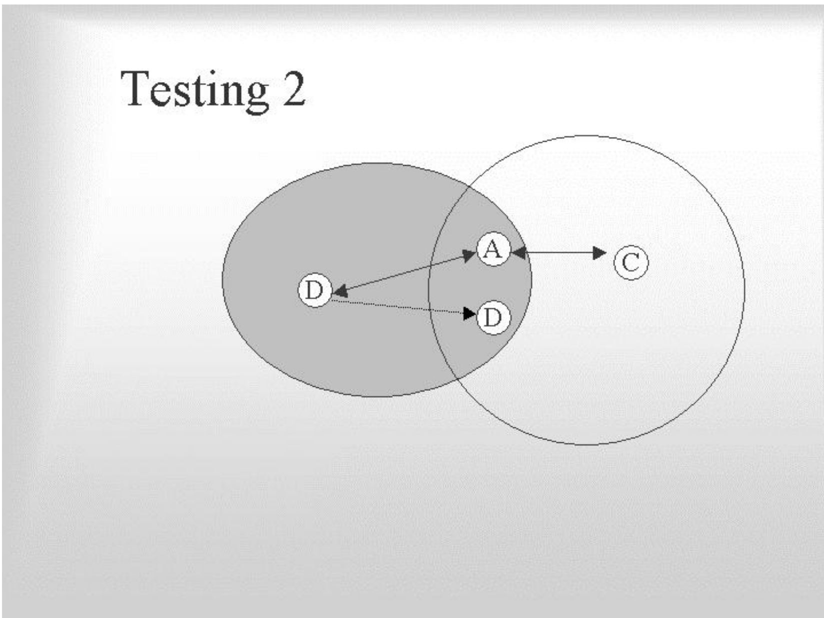


Figure 5.11 two ranges test

This test is completed by three peers, which are deployed to two ranges. In order to test the adaptability to mobility, one peer D as shown in figure 5.11, moves into the same range of A and B. In this way, both A and B detect that a remote peer has gone but a new local peer joins.

- Three ranges test

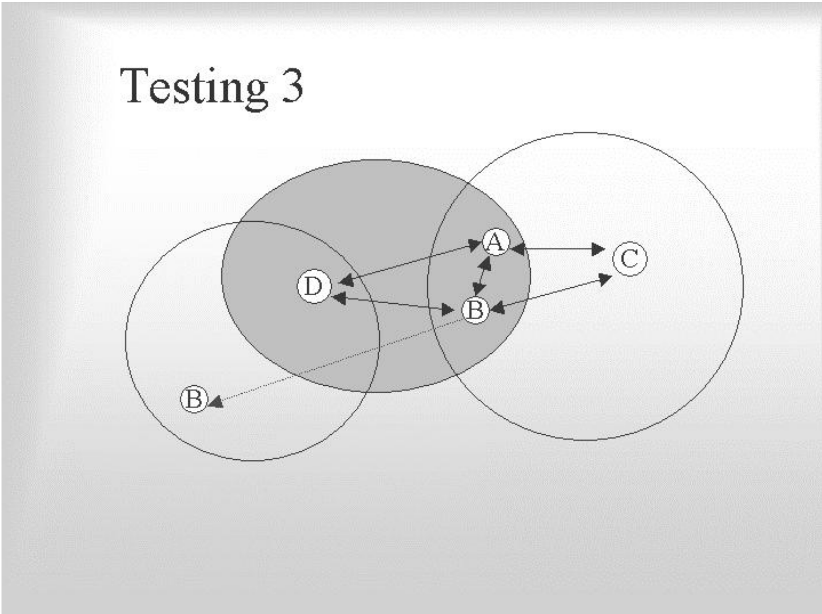


Figure 5.12 three ranges test

This test is completed by four peers, which are deployed to two ranges at the beginning and then changed to three ranges. Also in order to test the adaptability to the mobility, one peer B as shown in figure 5.12, moves into the third range out of A, C and D.

In this way, A finds a local peer B disappears and then finds a remote peer B.

C finds a local peer B disappears and then finds a remote peer B.

D does not find any changes.

B detects that both local peer A and C disappear and then turn to be remote peers.

No matter these peers are in one range or two, three ranges, after some adaptation, they can still communicate to each other.

Three different networked scenarios are described previously. In the actual tests, more tests have been done on the basis of three topologies such as extending the mobility out of range, or with two peers moving.

5.4 Summary

In this chapter, how the components of the ad-hoc peer-to-peer chat application are implemented was discussed in detail. How these components work together and how users can interact with the GUI of the application is also explained in detail. After many tests with different networked scenarios as described, the application was shown to meet the requirements successfully and achieve the goals of the project.

Chapter 6

Conclusion

6.1 Conclusion

As stated in the objective in chapter 1.2, many researches have been done in the fields of wireless technology, mobile computing, ad-hoc network and peer-to-peer system. These researches helped the design and implementation of the peer-to-peer chat application in ad-hoc wireless network. Because of the characteristics of mobility, mobile devices may frequently move out of communication range, so the situation of multi-hop communication is considered in this system. Nodes in an area can act as routers and forward messages among nodes that cannot communicate each other directly. Since it is a pure peer-to-peer application, the main design issues such as peer discovery and egalitarian roles of peers are introduced.

In chapter 4 and chapter 5, the details of the design and implementation of a peer-to-peer chat application in ad-hoc wireless network was introduced. We addressed the design issues that are important in ad-hoc peer-to-peer systems and analyzed the requirements of the chat application. The details of the solutions to each of the design issues were also described. After many tests using different test scenarios described in the section of chapter 5.3.2, the chat application was proved to be successful and efficient. It met the requirements and achieved the goals of the project. The performance on Pocket PC was satisfying. The whole .jar file deployed on mobile devices was only 31K.

The advantages of the chat application are:

- Pure peer-to-peer application

  This chat application is a pure peer-to-peer application, every peer acts as both client and server, as well as router.

Any peer can join or exit the ad-hoc network at any time.

- Multi-hop routing

  This chat application supports multi-hop routing using IP multicast.

- Mobility support

  Any of the peers can move arbitrarily during the chat. All the peers can get the latest information dynamically and automatically while moving without restarting the application.

Because the chat application is to be used in ad-hoc networks and based on IP multicast, so it has some inherent disadvantages:

- IP multicast is not reliable

  IP Multicast extends UDP and a sender sends messages to a multicast address and the receivers have to join that multicast group with the multicast address to receive them. The message transmission is unreliable, and there is no notion of membership.

- Ad-hoc networks are not reliable

  Because of the high mobility and wireless links, ad-hoc networks are not as reliable as wired networks or wireless networks with infrastructures.

The increasing popularity of real-time and multimedia applications over the Internet makes the emergence of reliable multicast services. Recent years, many researches have been done on reliable multicast protocols such as SRM (Scalable Reliable Multicast), RMTP (Reliable Multicast Transport Protocol), MTP-2 (Multicast Transport Protocol) and so on.

These are also many researches addressing the reliability and mobility in ad-hoc networks to deliver QoS within ad-hoc networks. Some multicast routing protocols such as MAODV (Multicast Ad Hoc On-Demand Distance Vector protocol), AMRIS (Ad hoc Multicast Routing protocol utilizing Increasing id-numberS) [13] have been proposed to take advantage of multicast in terms of reducing the routing protocols' overhead and improving the performance to deal with the high mobility and dynamic changes of multi-hop ad-hoc networks.

In order to design and implementation this chat application, a lot of research has been done on the fields of peer-to-peer systems, ad-hoc networks, mobile computing and wireless technologies. These fields are all ones of the most robust technologies at present. Some of the researches have been shown in chapter 2. If have more time, we could research more and understand more deeply on these technologies. Anyway, this project made us gain a lot of experiences and knowledge of these interesting fields.

6.2 Peer-to-peer Future

Peer-to-peer is not a new concept but draws a lot of attention recent years because of the emergence of more powerful computers, higher bandwidth but lower costs networks. Peer-to-peer brings us new approaches on file sharing, collaboration, distributed computing etc. Unlike client/server architecture, which must depend on the powerful servers running, peer-to-peer systems provide real non-stop services. Though peer-to-peer technology is under the pressure of security, privacy and management issues, it is still a robust technology that will bring us lots of changes to traditional system architecture with the treads to more collaboration and powerful computing capability on the basis of ubiquitous networks.

References

[1] Datamonitor report, http://www.datamonitor.com

[2] Finfacts Irish Internet Statistics Centre - Financial Services, http://www.finfacts.com/intfs.htm

[3] Wireless Financial Services, Cambridge, MA, USA, July 7, 2000 http://www.celent.com/PressReleases/20000707/wireless.htm

[4] C.K. Toh: "The Wireless Network Evolution." By Article is provided courtesy of Prentice Hall PTR. APR 12, 2002

[5] Paul Mclvor: "Third Generation Wireless - A Necessary Evolution". April 05, 2001

[6] The UMTS Forum, http://www.umts-forum.org

[7] Puneet Gupta: "Will WLANs cannibalize 3G? ", CNETAsia. May 28, 2002

[8] Mads Haahr: lecture notes of NDS 101 course, Trinity College http://www.cs.tcd.ie/courses/mscnds/handbook/nds101/

[9] Andy Oram: O'Reilly Network, Gnutella and Freenet Represent True Technological Innovation, May 12,2000

[10] Brendon Wilson: the book "JXTA", chapter 1, p12-14

[11] Brendon Wilson: the book "JXTA", chapter 3, p39-44

[12] Padmini Misra: Routing Protocols for Ad Hoc Mobile Wireless Networks

[13] C.W. Wu, Y.C. Tay: AMRIS: A Multicast Protocol for Ad hoc Wireless Networks

[14] Richard Koman: David Anderson: Inside SETI@Home, Feb. 15, 2001

[15] Albert Proust: A Bluetooth Primer, Nov. 03, 2000

[16] Craig C. Freudenrich: How SETI Works, http://www.howstuffworks.com/seti.htm

[17] Susan Breidenbach: Feature: Peer-to-peer potential, Network World, July 30,2001

[18] George H. Forman, John Zahorjan: The challenges of mobile computing, March 9,1994

[19] Infrared Data Association, Serial Infrared Link Access Protocol, version 1.1, 1996, http://www.irda.org

[20] Technical Summary of "IrDA DATA" and "IrDA CONTROL", http://www.irda.org

[21] IrDA: Background and Overview, http://msdn.microsoft.com

[22] How IP Multicast Works, An IP Multicast Initiative White Paper; Stardust Technologies, Inc

[23] IP Multicast Backgrounder, An IP Multicast Initiative White Paper; Stardust Technologies, Inc

[24] S. Deering: RFC1112, Stanford University, 1989

[25] FAQs- iPAQ Pocket PC H3900 Series http://www.compaq.com/products/handhelds/pocketpc/faq_H3900.html

[26] Jeode platform Sun-authorized Java Compatibility, http://www.insignia.com/content/products/otherjeode.shtml

[27] John Zukowski: "Think small: Java on Compaq's iPAQ", http://www.javaworld.com/javaworld/jw-10-2001/jw-1026-devices_p.html

[28] "What is Java Technology? ", http://java.sun.com/java2/whatis/

[29] "Write Once, Run Anywhere(TM): Why IT Matters; ", http://java.sun.com/features/1998/01/wora.html