uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Dawei Ning

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Systems Science)

GRADE / DEGREE

Systems Science

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

A Novel Wireless Based Architecture for Museums

TITRE DE LA THÈSE / TITLE OF THESIS

A. Boukerche

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Morad Benyoucef

Wail Gueaieb

Gary W. Slater

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

A Novel Wireless Based Architecture For Museums

by

Dawei Ning

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.Sc. degree in
Systems Science

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

Your file Votre référence
ISBN: 0-494-14934-5
Our file Notre référence
ISBN: 0-494-14934-5

Abstract

Wireless networks were first developed in the 1970s, and people's interest in them has been growing ever since. A broad range of differing capabilities is covered by wireless technologies and is oriented toward different uses and needs. In this thesis, we propose a novel application architecture based on wireless networks and designed for museums to allow visitors, especially group of visitors, to have an attractive tour experience. While current museum visiting's approaches are largely interactive but have a number of disadvantages, this application utilizes mobile devices as facilities to help visitors benefit not only from visiting a real museum environment but at the same time enriching their ability to search for additional material. Moreover, as it provides powerful functions, the application will enable visitors to establish a virtual community platform by which they will be able to communicate and share information through wireless networks.

The following publications are part of this thesis:

Conference:

1. Azzedine Boukerche, Dawei Ning, Regina B. Araujo, UARTP - A Unicast-based self-Adaptive Reliable Transmission Protocol for Wireless and Mobile Ad-hoc Networks, ACM PE-WASUN 2005 October 10-13, 2005 Montreal, Quebec, Canada.

2. Azzedine Boukerche, Dawei Ning, A Novel Wireless Based Architecture For Museums, to be submitted.

Acknowledgements

I wish to express my gratitude to Dr. Azzedine Boukerche for his financial aid and because he gave me the opportunity to develop a really interesting project. Dr. Azzedine Boukerche constantly gave me helpful suggestions during each phase of the development of the project so that I could finish my thesis successfully.

Moreover, I wish to thank my mother sincerely for her unconditional support and constant encouragement. I will never forget her as well as her helps in my life. I would also like to thank my wife. She gave me two greatest gifts during my studying period: Alwin and Oswin. These two lovely little boys always gave me a lot of enjoyment when I felt frustrated.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Do we need a complex system for our museum tours? Few people have considered whether they need a complex approach for museum tours. Most people think that there is no other way to visit a museum except to buy a ticket, enter the museum, and begin their tours. However, with the popularity of web techniques, people have another option for visiting museums. They can visit a virtual museum through the Internet. A number of successful websites that provide virtual tours using corresponding web technologies exist, but if we examine these features more closely, we find that these approaches have their respective drawbacks.

1.1.1 The Conventional Visiting Approach

The conventional approach for visiting museums is the most comprehensive and simple. People buy tickets, enter museum exhibits, and stop in front of whichever paintings or other artifacts interest them in order to examine them further. However, it is impossible for people to obtain sufficient information concerning the items they are interested in when the descriptions of these items are printed on a small label. If visitors wish to obtain further information about the paintings or artifacts, they must go to the information desk or seek help from museum guides. This problem applies particularly to group tours of museums, where group instructors must coordinate students' activities according to schedule. All instructors have a similar experience in that when they make a presentation or announcement, certain students are distracted by other museum artifacts or paintings and pay attention to these instead of to the instructor and item he or she is talking

about. In addition, students may chat with others, further disrupting the group tour. To make matters worse, instructors need to spend a large amount of time looking for people who are lost. This always results in some messy situations, especially when large groups are concerned (See Figure 1.1). Therefore, instructing classes within museums as opposed to traditional classrooms is challenging for instructors.

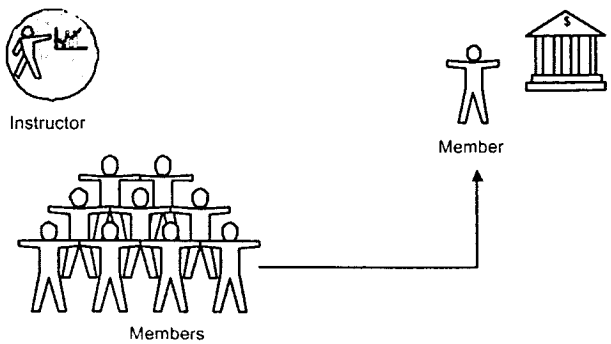

Figure 1.1: The Conventional Visiting Approach

1.1.2 Web-based Visiting Approaches

The web-based visiting approach for museums emerged after the Internet gained popularity. The Internet and its various applications have transformed our lives. We can shop, take classes, and meet friends online. In addition, we can visit museums, even as a group.

Virtual museum visiting has numerous advantages. Not only can people visit online museums anytime, but they are not constrained by schedules, weather, parking, accessibility limitations, or other factors. Furthermore, during instructor-student visits, both the instructor and students can remain within their classroom in front of computers instead of having to venture inside a real museum as a group. Students who are lead by an instructor can enter the virtual museum environment accompanied by video and audio features from the Internet. The instructor, as a central controller, can completely monitor and manage the status of his ro her students. This web-based visit approach efficiently solves the problems that emerge in the conventional museum-visiting approach (See Figure 1.2). In the web-based approach, all activities take place in a virtual environment. However, most people prefer to interact with an outside environment instead of remaining indoors sitting in front of a classroom computer.

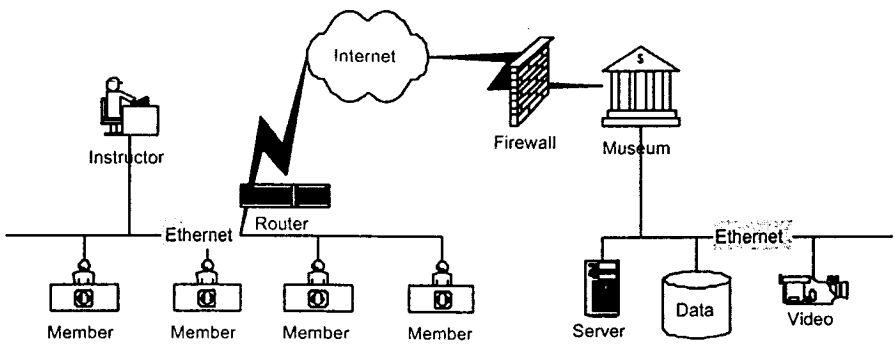

Figure 1.2: The Web-based Visiting Approach

	Flexible	Environment	Information	Controllable†
Conventional Approach	Poor	Real environment	Limited	Not efficient
Web-based Approach	Good	Virtual environment	enrichment	efficient

†For instructors to manage their members.

Table 1.1: A Summary of Two Approaches

Table 1.1 shows the advantages and disadvantages of these two approaches. Our purpose is to find a new, feasible approach that benefits from the advantages of both visiting styles.

1.2 Our Proposed Approach

There is a new idea emerging: there exists an approach which integrates traditional and modern visiting approaches so that people can share the enormous amount of information and flexible managing functionality of advanced mobile devices while touring in a real museum environment. This was our motivation for designing new application architecture. In this thesis, we propose a novel approach which can be summarized as follows:

- First, imagine that you are an instructor from a school. You are going to lead students during a field trip to a local art museum. When you and your students arrive at the museum, you purchase tickets, then enter the museum. A staff member from the museum deploys a PDA with wireless connectivity to everyone. A specific

application is mounted on the PDA, and the application mounted on your own PDA is slightly different from that of your students'. After receiving a brief introduction about the museum as well as some instructions, your group begins its tour. The PDA and the application on the PDA perform as a virtual museum guide. You and your students follow a route recommended by the application. Under your direction, students begin to use a scanner embedded in the PDA to scan the bar codes beside paintings and other exhibits. By using bar code scanners, PDAs can obtain an identifier of the painting. Using this identifier as a search word, students can obtain information about the museum items from a database. Because it is not suitable for there to be a lot of talking in museum environments, students can use the chat function provided by the application to discuss paintings with one another.

- Sometimes you must to give a short presentation to your students. However, you know form experience that not all students focus on your speeches. This problem is solved by the PDA. Before starting your presentation, you freeze some of functions of students' PDAs. For example, you can shut down the search and chat functions. During your presentation, you can send information or pictures to students through the application and wireless network in order to help them have a clear idea about your presentation. After finishing the presentation, you can resume the functions of the students' PDAs that you shut off. In order to ensure that students remain on track even after your presentation, you also have a channel to monitor students' PDA activities. Every PDA has a history of its users' activities. Therefore, you have the ability to know what your students are doing at all times.

- Coordinating students' activities is also a big challenge that instructors must face during a field trip. When classes move from one room to another in museum, instructors often lose students and must constantly look for them. However, this is not an issue for classes equipped with PDAs. By using a location awareness mechanism, you can immediately know the location of each of your students. Therefore, you can ask the lost student to rejoin the group or tell him or her where you are planning on going next. Because of this function, no one will get lost.

- File or information sharing is another important function we must mention. This function provides a good communication channel between students and the instructor. Under current conditions, we cannot ensure that our networks and/or servers

run in perfect order all of the time. Sometimes the network may have a serious traffic jam. In traditional networks, visitors cannot obtain information from network through PDAs. However by using the sharing mechanism, your group still can benefit from our application. For instance, let us say that student A wishes to obtain information about a particular painting. However, the network develops certain problems, so student A cannot obtain information through the network. However, student B has a copy of this painting on his PDA. Therefore, student B can share this information with student A as though the network never broke down. This function is suitable to another situation in that when your group moves to an outside building that the network does not cover.

- All of the above-mentioned focuses on the activities of one group. However, our application does not exclude communication between two groups. It is possible for there to be two or more groups in a museum at the same time. In addition, these groups may be required to exchange information. Such intercommunication cannot take place without the authorization of both you and the instructor of the other group.

Therefore, by using this proposed approach, we can obtain benfits from both the conventional approach and the web-based approach to visiting museums. Museum visitors are able to obtain as much information about the museum as a website offers, while experiencing a real museum environment as opposed to a virtual one.

Chapter 2

Museum Application Architecture

The first step of an application development is designing its architecture. Good architecture can get twice the results of average architecture with half the effort. On the other hand, poor architecture will result in more difficulties in the development phase or even the failure of the entire application. Application architecture is composed of two parts: the network platform and the application framework. Roughly speaking, in the field of application software development, there are two types of application framework: the centralized client-server application framework and the distributed peer-to-peer application framework. In the field of network platforms, there are two types: the traditional wired network platform and the wireless network platform. The wireless network includes the infrastructure-based wireless network and the mobile ad hoc wireless network. Although these technologies are different, we cannot say which one is better or more advanced than others. Today, each technology is applied widely in corresponding and suitable situations. The background of these technologies will be provided as follows.

2.1 Analysis of the Wireless Network Platform

2.1.1 Infrastructure-based Wireless Network Platform

Wireless networks were first developed in the 1970s, and interest has been growing ever since [31]. Two kinds of wireless network platforms exist. The first and most popular of these is the so-called infrastructure-based wireless network, which is built on a pre-existing fixed wire network. The wired network is the backbone and infrastructure of this kind of network environment. At certain sites where a wireless environment is needed, some devices, called access points, are distributed and mounted on the infrastructure.

Access Points are used as a bridge between wireless mobile devices and the backbone network. The wireless mobile devices connect to the network through these access points. The range of access points needs to cover an entire area in which wireless mobile devices can roam and connect to the network (See Figure 2.1). If the wireless mobile devices move outside the range of access points, the connection between these devices and the network is broken. Communication between two wireless mobile devices also requires that both of these devices be within the range of access points.

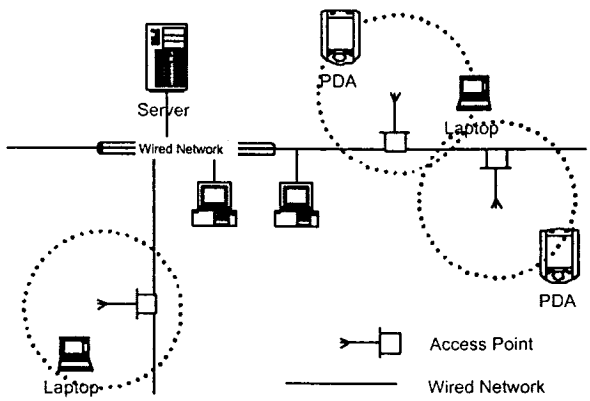

Figure 2.1: Infrastructure-based Wireless Network

2.1.2 The Mobile Ad Hoc Wireless Network Platform

Another kind of wireless technology is called mobile ad hoc network technology (MANET). A mobile ad hoc wireless network is a peer-to-peer network that allows a set of wireless mobile devices to communicate with each other and keeps connectivity without any support from a pre-existing fixed wire network. Compared to the infrastructure-based wireless network, if any two wireless mobile devices support ad hoc mode and are within radio range of each other, an ad hoc network is formed. When two wireless mobile devices outside of radio range need to communicate with one another, they can do so only via other roaming devices that act as routers [18]. This is because the mobile ad hoc network is built on arbitrary and temporary network topologies in which wireless mobile devices can self-organize freely and dynamically [22]. Devices can communicate in this network without any pre-existing physical network devices. A wireless mobile device can freely join and leave the network topology and also anticipate data transfer activities

(See Figure 2.2). Due to the distinguishing features of the network, the globally arbitrary mobility of devices, and self-organizing network topology, mobile ad hoc wireless networks have huge application potential, particularly in military, battlefield, rescue, and sporadic happenings coverage, where it is impossible to set up an infrastructure-based network because of hostile natural environments or matters of great urgency.

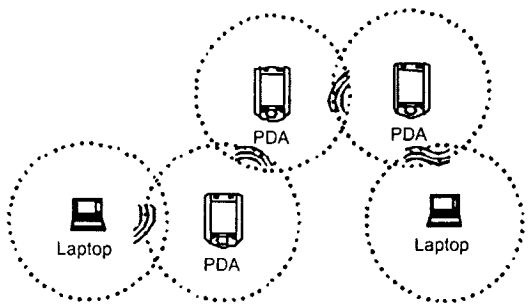

Figure 2.2: The Mobile Ad Hoc Wireless Network

2.1.3 The Difference Between Two Wireless Networks

An infrastructure-based wireless network is an extension of the traditional wire network. A fixed wired infrastructure is the pre-requirement for this network. Therefore, wireless mobile devices can also benefit from the stability of wired networks, including features such as robustness, stability, and reliability in performance. However, due to dependence on infrastructure, the effective area where wireless mobile devices can operate is constrained by the radio range of the access points. Another important issue is that two wireless mobile devices cannot communicate directly. All communications must be processed via the access points.

In contrast, the mobile ad hoc wireless network is built on temporary and dynamic topology. It is a self-creating, self-organizing, and self-administering network without any infrastructure or administrative support [39]. As mentioned above, if wireless mobile devices are within radio range of each other, an ad hoc network is formed. Generally speaking, ad hoc networks have better flexibility than infrastructure-based networks. However, since the topology of mobile ad hoc wireless networks changes rapidly and unpredictably, the stability and reliability of the network are affected. We can reduce those effects through adopting suitable protocols and algorithms in software architecture.

Therefore, mobile ad hoc wireless networks are one of the most popular subjects today.

2.2 Analysis of Application Frameworks

2.2.1 Centralized Client-server Framework

2-Tier Client-server Framework

Client-server framework is the second generation of application architecture and is also the first network-based application architecture to become commercially viable and prevalent [17]. Client-server framework allows us to enable communication between two applications, with one application acting as the server and the other acting as the client [17]. The server and clients form a request-response structure to complete their interactions. Clients give service requests and wait for responses from the server. Therefore, this framework is based on a centralized model in which one server provides services to many clients. In this centralized model, the client-server framework provides a mechanism for the distributed process and centralized storage of resources. Modern database applications are one example of client-server framework, in which the primary functions of the server are to store data and perform business logic. However, in clients, besides providing multiple user interfaces for different users, client applications perform pretreatment of raw data. If the server is not accessible, clients may temporarily cache data in a local memorizer and synchronize data with the database when the server becomes available.

n-Tier Client-server Framework

As client-server framework evolves, computing framework further separates in structure. Business logic and data pretreatment are separated from the server and clients to form new tiers, the application servers, which are deployed between the server and the clients. These new middle tiers provide computation services of business logic accessed by the clients (See Figure 2.3). The distribution of business logic between the server and the clients enables the clients to focus on data rendering and user interface. The advantage of the separation of components is to make the components in each tier independent. For instance, we do not need to be considered the communication mechanisms when we design the code structure of applications. Therefore, in such a framework, we benefit from scalability and distribution.

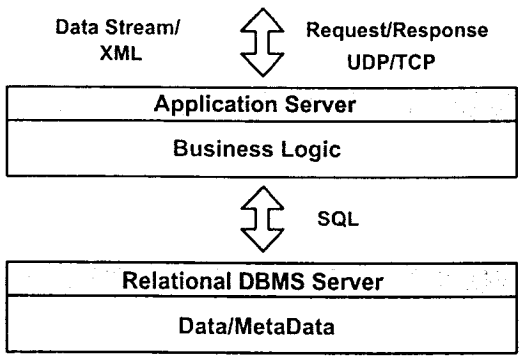

Figure 2.3: Client-server Framework

Web-based Client-server Framework

With the evolution of the Internet, web technology has become one of the most popular technologies today. The Web is essentially a client-server technology where the clients and server communicate through HTTP (Hyper-Text Transfer Protocol) [17]. In the proposed application, clients are replaced by the web browser, which is the third party application of the various platforms. The server, now the so-called web server, wraps the data in markup code, such as HTML, XML, and scripts. HTTP is a connectionless and stateless transmission protocol based on TCP/IP. When an HTTP client, namely the web browser, connects to the web server, the web browser sends an HTTP request. The web server, after receiving this request, returns an HTTP response with respective data wrapped in HTML or XML format. The Web browser then renders these HTML or XML markup codes into webpages, which are then provided to the final user (Seeing Figure 2.4). Since web-based framework provides users with a standard client interface to users, users can use applications without any further training.

Brief Summary of Client-server Framework

Because data and business logic are deployed in the centralized server, it is easy to synchronize data and manage business logic in the centralized client-server framework. All clients share the same resources. However, the centralized client-server framework requires that all users be connected to the server. Another important issue we need to mention is that if one client has the data that another client needs, there is no way to

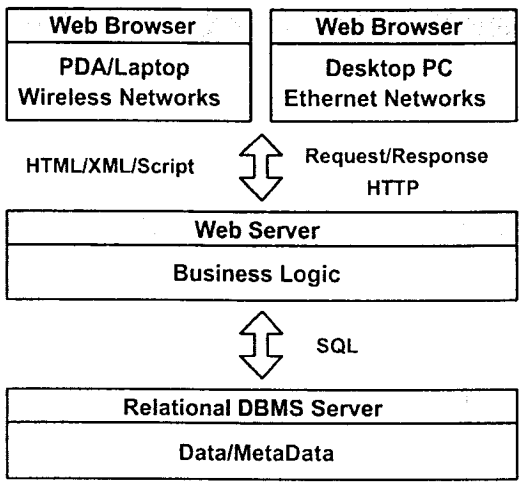

Figure 2.4: A Web-based Framework

let the two clients exchange the data directly. Moreover the clients are not a stand-alone application; they have to cooperate with the server in order to perform tasks. This characteristic results in poor performance in fault tolerance: if the centralized server has a problem, the whole system fails. Another drawback of the centralized client-server framework is that the server does not initiate interaction with clients. In other words, the server does not give information to the clients. The common way to obtain information is for clients to pull information from the server.

2.2.2 Distributed Peer-to-peer Framework

Peer-to-peer framework is defined as distributed systems without centralized control or hierarchical organization where the application running at each node has equivalent functionality [28]. Unlike the centralized client-server framework, in distributed peer-to-peer framework, there is no necessity for a centralized server. All participants in the system, who are referred to as peers, are equal participants and play two roles at the same time: server and client. Therefore, this framework provides a mechanism through which a peer can access any other peers, although it is not a requirement that all peers have the ability to connect to each other. In such a framework, we efficiently solve the problems presented in centralized client-server frameworks (See Figure 2.5).

- Peer-to-peer framework does not require connection to the server. Furthermore,

there is no requirement that all peers be connected to each other. Therefore, peer-to-peer framework provides better fault tolerance than centralized client-server framework.

- Due to the distribution storage mechanism, users can maximize the utilization of resources. If a peer needs information, other peers may be able to satisfy his or her requirement and can set up an information-sharing channel in order to do so. Therefore, each peer has access to the resources of all other participants.
- Communications among peers are bidirectional and more flexible. Peers can either push data to or pull data from others.

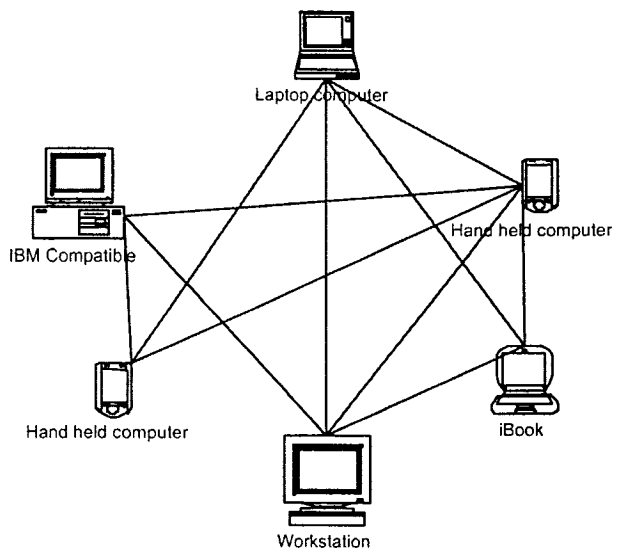

Figure 2.5: A Distributed Peer-to-peer Framework

2.3 An Effective and Efficient Solution for Our Application

2.3.1 Our Objective

We define our application as a mobile application based on wireless networks. This application is designed for group visitors in museums. By using advanced wireless mobile

devices, people can obtain an attractive tour experience when visiting museums. This attractive tour experience, generally speaking, allow people to obtain sufficient information using an efficient communication mode and effective managing mode when visiting real museum environments.

2.3.2 The Hybrid Network Platform

Since our application is based on the wireless network and allows visitors to use wireless mobile devices when they are moving, we first need a wireless network environment in the museum by which wireless mobile devices can access the network and exchange information with servers, and can communicate with each other. From the pervious section, we know there are two kinds of wireless networks we can choose from. In infrastructure-based wireless networks, we can benefit from stability and robustness. However, museum visitors often wander away form their group and can therefore not participate in organized group activities. It is difficult to limit their territory. Additionally, we cannot ensure that all wireless equipments is in good condition at all times. However, the self-creating, self-organizing, and self-administering networks built on a temporary and dynamic topology characterize the mobile ad hoc wireless networks. Any wireless mobile devices with ad hoc mode can form a network without any infrastructure support. Therefore, in our case, we need to find a hybrid wireless network platform to benefit from the stability and reliability of infrastructure-based wireless networks as well as the flexibility and self-organization of mobile ad hoc wireless networks. Moreover, we need to integrate the wireless network with the traditional wired network so that all wireless mobile devices can obtain channels to access the server and the database.

In our application, we have three heterogeneous networks: the traditional wire network, the infrastructure-based network, and the mobile ad hoc wireless network. However, due to the characteristics of the infrastructure-based wireless network, the wire network and the infrastructure-based wireless network can be considered together. Therefore, the main challenge we need to face is the integration of these two kinds of wireless networks when they use different communication channels.

Many papers address hybrid wireless network issues. Chen *et al* present a mixed-mode WLAN that is the integration of ad hoc mode with wireless LAN infrastructure

so that mobile devices can dynamically switch between infrastructure mode and ad hoc mode according to the instruction of the access points [29]. Miller *et al* propose a hybrid network implementation to extend infrastructure reach that focuses on dynamic IP assignment and routing protocol in the hybrid wireless network [?]. In [29], authors present an AP-oriented scenario where access points are in charge of managing mobile devices which need to switch to the respective mode. Each mobile device needs to report its request for remaining in the ad hoc mode. If access points have not received a report after a certain amount of interval time, the mobile device will return to infrastructure-based mode. A challenge existing in [29] is how to switch a device's mode when it is not within transmission range of the access point, where this device also cannot send a report to the access point to claim the request for remaining in ad hoc mode. In addition, many control messages are delivered to switch mobile devices' modes. It is unavoidable that there are a large number of overheads produced in the wireless network. In a bandwidth-limited mobile ad hoc wireless, too many overheads result in poor performance.

In our case, we adopt a static pre-assigned IP scenario instead of a dynamical IP scenario. Therefore, we do not need to focus on the implementation of the DHCP in the ad hoc mode. Firstly we need to let all mobile devices operate in ad hoc mode. The key equipment in our hybrid wireless network is the access point. The access point will play the role of a gateway which connects two subnetwork platforms: the mobile ad hoc wireless network and the wire network. The access point anticipates all activities in ad hoc mode with all mobile devices; for example, the ad hoc routing discovery and routing maintenance. On the other hand, we need to allow mobile devices to access wire networks. In this hybrid mode (See Figure 2.6), we have the following characteristics:

- In the transmission range of the access point, according to the the shortest path rule of ad hoc routing protocol, all the mobile devices are only one hop away from the access point. Therefore, all transmissions between the mobile devices and resources in the wire network are conducted via this relay. This transmission mode is the same as the infrastructure mode.

- Since all mobile devices are in ad hoc mode, if certain mobile devices in the range of the same access point want to communicate with each other, they can do so directly without the access point's participation. Therefore, we can balance bandwidth utilization in access points.

- By using this hybrid mode, we can extend the coverage of the network. A mobile

device outside the range of the access points can also connect to the infrastructure network via a mobile device that is within the range of one access point. In museums, for example, there is usually a limited range of coverage with respect to the entire area. There may be access points inside museum buildings, but mobile devices just outside the buildings cannot access these. However, if there are other mobile users, or dedicated relays, in the lobby of the building, such devices may serve as routers to extend the range of the access point to outside mobile devices.

- Communication can also be set up between two mobile devices which are in the range of different access points. As a relay, data is transferred from one access point to another through the wired network.
- If a group of mobile devices move outside the range of access points because, they are still in connected. They still can communicate with each other even if the infrastructure network or some of the access points fail.

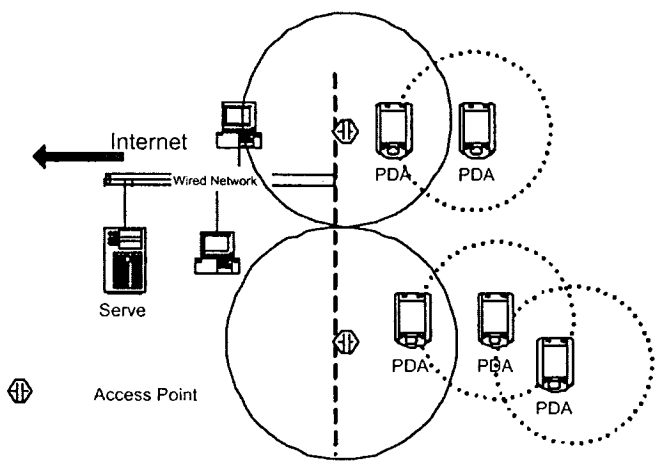

Figure 2.6: A Hybrid Wireless Network

2.3.3 A Framework Based on the Application Proxy

Our application framework is divided into two parts: the internal part and the external part. Internal framework is the internal application system, which includes database, security management module, and other administrating modules. The main administrators

and operators are the museum staff. For example, in this part, administrators update the database, maintain the system, analyze the status of visitors in real-time, and perform other museum functions. Therefore, we use the traditional client-server application framework based on the wire network platform in this part. We can adapt more mature security mechanisms (there are many mature security products in today's market). In addition, more reliable transmission performance is provided by the wire network, and more stable application framework is characterized by the client-server framework. Another reason for us to adapt this framework is that users in the internal framework have characteristics of relative fixity. Most of them do not need to move around while operating their machines. There is a exception for mobile administrators. Sometimes, administrators in the museum can use mobile devices to maintain these facilities from one site to another. They also need to communicate with the server or database directly with respective authorizations.

Another part is the external framework provided to visitors. According to the analysis of activities of visitors, besides searching for information on the database, visitors also need to communicate or exchange data with each other either when the infrastructure-based network is available or when it is unavailable. A distributed peer-to-peer framework allows visitors to benefit from distributed storage, data sharing, and self-organization. Therefore, visitors can share resources with each other in any situation.

Hence, the current challenge is how to integrate these two subsystems into one application framework when the client-server framework and peer-to-peer framework are two heterogeneous application frameworks with two different implementation mechanisms. We introduce a popular concept, application proxy, which is used as a relay to process data moving between two heterogeneous application frameworks. Also known as application gateways or application-level proxies, they are an application program that runs on a system between two different application frameworks. An application proxy can integrate two or more heterogeneous application frameworks seamlessly into a hybrid system (See Figure 2.7). When a mobile client belonging to the external frame wants to establish a connection to a service belonging to the internal framework, it connects to the application proxy first. The mobile client then negotiates with the proxy server in order to communicate with the destination service. In effect, the proxy establishes the connection within the internal framework and acts on behalf of the mobile client. This creates two connections: one connection between the mobile client and the proxy

server, and another connection between the proxy server and the server. In external peer-to-peer framework, the application proxy is a common peer which anticipates the resource share by providing the respective resources which are from the internal frame. To internal frame, the application proxy is a proxy for resource queries and a relay that forwards resource to the querist. An analysis of the application proxy has the following primary advantages:

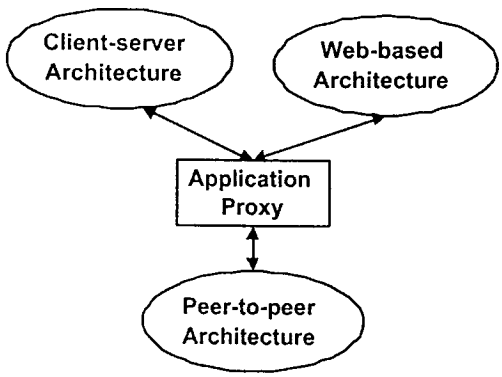

Figure 2.7: An Application Proxy

- A complex application usually needs to satisfy several different requirements. It is difficult to match all different application environments by providing only one single frame solution. A complex application is organized by several different application frames. An application proxy provides a coordinated mechanism to integrate these heterogeneous frames into an entirety.

- The application proxy also provides a security mechanism to the application. Since the whole application is divided into two parts: internal frame and external frame, which are connected by the application proxy, we can use a filter unit in the application proxy to control accesses coming from the external frame. For example, in the external frame, besides visitors, administrators also use mobile devices to maintain the system in the museum. And both visitors and administrators need to access the server or the database but have different access authorities. By using the filter unit of the application proxy, we can protect internal resources from unauthorized access.

Reliable Transmission Mechanism

Although we adopt the hybrid wireless network as our network platform, we still mainly focus on the ad hoc mode. The reason for this is that all the mobile devices and access points perform in ad hoc mode. In fact the mobile nodes and access points form a mobile ad hoc wireless network. Therefore, mobile devices and access points have all the characteristics of ad hoc mode. In addition, since mobile ad hoc wireless networks are characterized by low and highly variable bandwidths, highly variable latency, and unpredictable topology changes. these particular characteristics bring affect network performance. We know that in such an unstable wireless network environment, it is impossible to push the wireless network into a wider practical level if there is no mechanism to provide a desired transmission performance and to guarantee reliable network communication. In the data transmission layer of this application, Unicast-based self-Adaptive Reliable Transmission Protocol, which we refer to as UARTP, is proposed for such wireless networks. The basic strategy of this protocol is to use the combination of Automatic Repeat reQuest (ARQ) [9] and probabilistic Forward Error Correction (FEC) [4], namely a combination of retransmission and probabilistic erasure code mechanism, to help set up an effective protocol to reduce packet loss and improve transmission efficiency while keeping a reasonable overhead and simple algorithm for implementation. A detailed description of this process is presented in Chapter 3.

The Mobile Peer-to-peer System

While the peer-to-peer (P2P) system has attracted a great deal of interest from various research and commercial organizations, mobile implementation of p2p application is still in its first step. In this application, a mobile peer-to-peer system architecture, referred to as MAHP2P, is introduced. This architecture enables dynamic adaptation among a triple-layer hierarchy while preserving available network connectivity and efficient load balance. A mobile-agent-based look-up system introduced by MAHP2P also provides approaches to reduce network traffic (See Figure 2.8). The detailed analysis of this look-up system is presented in Chapter 4.

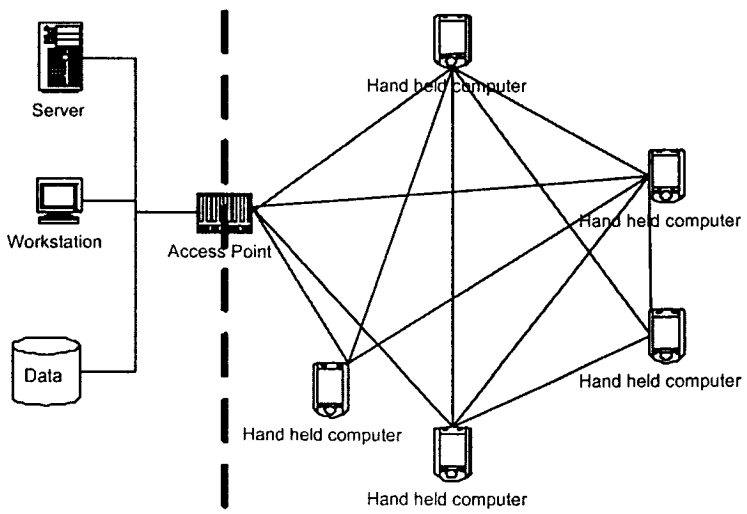

Figure 2.8: A Peer-to-peer System

Ad Hoc + Peer-to-peer in one Architecture?

Many people confuse the ad hoc network with the peer-to-peer system. Some people think that ad hoc and peer-to-peer are the same. In fact, although both have similarities concerning their routing discovery and maintenance schemes, they are different systems. Mobile ad hoc networks and peer-to-peer systems are established on a different OSI layer. Mobile ad hoc wireless networks are based on the network layer, while peer-to-peer systems are based on the transport layer. We know that there are two wireless network mode: infrastructure-based and ad hoc. In addition, there are two application frameworks: client-server and peer-to-peer. We can design a client-server application based either on the infrastructure wireless network, or on the ad hoc wireless network. On the other hand, we can also build a peer-to-peer either on the infrastructure wireless network or on the ad hoc wireless network. To further illustrate this, we present the following characteristics of both techniques.

- An ad hoc wireless network is mostly user-driven, while the peer-to-peer system is data-driven [36]. The purpose of an ad hoc network is to discover and communicate with other mobile nodes, and the purpose of a peer-to-peer system is to redundantly distribute resources across the system. However, in our proposed application architecture, we need to use the synergetic effects of these two techniques to satisfy both members' discoveries, resource look-up, or sharing.

- the peer-to-peer system is separated from the physical network. The system topology of a peer-to-peer system is completely different from the topology of the physical network. Hence, it is very suitable for the "group" concept, which plays a key role in the instructor-student mode of our application architecture. In contrast, the topology of an ad hoc network is the same as the topology of a physical network. Therefore, it is easy for us to estimate the physical distance of one resource or mobile node. This is very important for our application.

- Peer-to-peer systems have their own routing protocol, such as Gnutella [1], Chord [27], and so on. Since peer-to-peer systems are built on the application layer, their routing protocols have lower performance than routing protocols built on the network layer. Thus, discovering mobile nodes, namely routing protocol, must be implemented on the network layer. However, in order to keep higher performance and other reasons, network layer routing protocols cannot provide an efficient mechanism in the field of security. But the usage of security mechanisms such as public key infrastructure (PKI) in peer-to-peer systems is easier than in mobile ad hoc wireless networks.

As one of the most popular network techniques built on a temporary and dynamic topology, mobile ad hoc wireless networks are characterized by the ability to self-create, self-organize, and self-administer. Meanwhile, due to the characteristics of the peer-to-peer system-the dynamic topology, balancing bandwidth, and distributed storage-this system is the best choice for a mobile application that is built on the mobile ad hoc wireless network.

Generally speaking, our proposed application architecture includes four layers. On the bottom, a wireless network platform is built; the median layer is composed of a transmission mechanism to provide reliable and efficient data delivery; a mobile peer-to-peer system is implemented on top of these two layers. The data description layer is the user interface (See Figure 2.9).

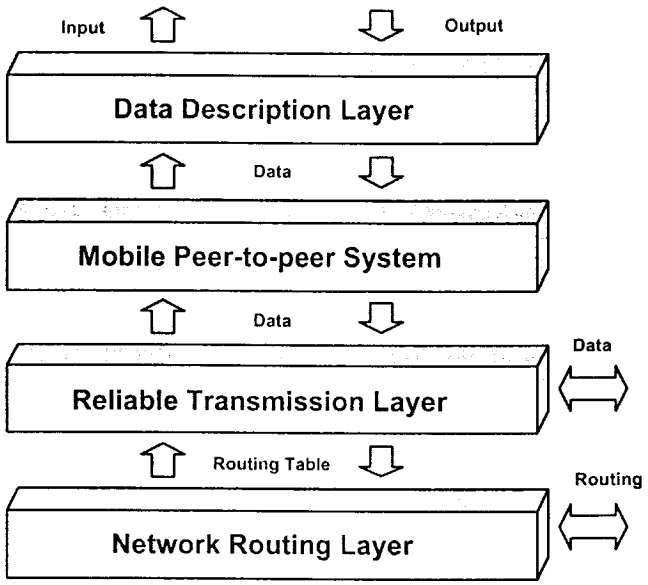

Figure 2.9: The Hierarchy of Application Architecture

2.4 Implementation

2.4.1 Wireless Mobile Device

We use PDAs (Personal Digital Assistants), a lightweight handheld devices, as visitors' terminal devices that function as a combination guidebook, virtual guide, map, and wireless communication tool. In our implementation, we adapt the HP Corporation's iPAQ rx3100 series PDA HP's iPAQ rx3100 unit is a very powerful mobile device with integrated Wi-Fi (802.11) and Bluetooth. Its specifications are shown in Table 2.1. The expansion slot allows us to use a SDIO in-hand barcode scanner supported in the PDA with SDIO slot. We use barcode scanners as optional input tools to interact with PDAs (Seeing Figure 2.10).

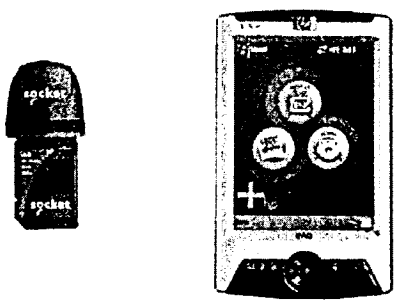

Figure 2.10: HP iPAQ rx3100 Series PDA with SDIO in-hand Barcode Scanner

Table 2.1: The Specifications of the HP iPAQ rx3100 series

CPU	Samsung S3C 2440 processor 300MHz
Display Screen	3.5" Transflective QVGA TFT Display with 64 K Colors
Operating System	Windows Mobile 2003 for Pocket PC
Memory Capacity	56MB Memory available to user
Slot	Built-in Slot Supporting SD/MMC and SDIO Cards
Network	Integrated Wi-Fi (802.11b) and Bluetooth

2.4.2 Interface

Member Registry

Since these wireless mobile devices are used by different visitors every day, it is impossible for visitors to remember all of the IP addresses of other members when they want to communicate with each other. Thus, we need a member registry mechanism to allow visitors to have a member list. By using a member list, visitors can communicate with each other. Creating a registry of instructors is the first step in implementing this program. In the first window of the application, there are two options for visitors to select: instructor and student. An instructor merely selects the instructor option, inputs his or her name, and clicks the button to complete the registry process. Then students can select the student option, ask for the instructor list (usually, there is more than one group in the museum), select an instructor, input their name, and click on the button. After the student registers as a member under his or her instructor's group, both instructor and student obtain a member list (See Figure 2.11).

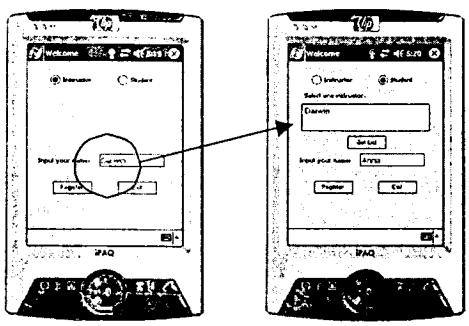

Figure 2.11: Registry Window of Instructor and Student Version

Information Search

Information search is the main function of this application. However, the mobile device or PDA is usually a small device with a very small screen. Moreover, the mobile application is usually used by mobile people. People use PDAs when they are walking or doing something else. Therefore, it is not convenient for people to input operations by using regular input methods; for example, soft keyboards and touch screens. In this application,

we use bar code scanners as an another input method. Every painting in the museum has an unique bar code to identify it. Visitors can use bar code scanners to get the IDs of the paintings. Through this ID, the application can obtain information from the database (See Figure 2.12).

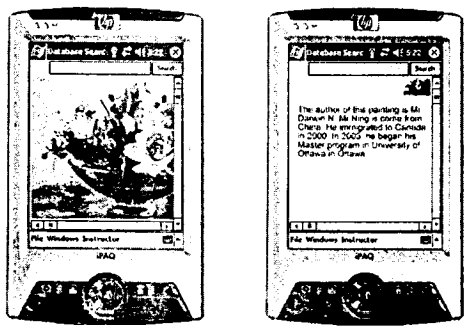

Figure 2.12: Information Search

Chat Room

We provide an online chat function to allow visitor to have group discussion online. Usually, group discussion are not permitted in the museums. Group visitors can use this function to discuss the museum, tour, or paintings. We can select members from the member list to chat with on a one-on-one or group basis (See Figure 2.13).

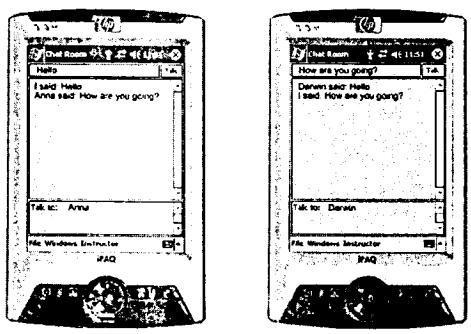

Figure 2.13: Chat Room

Control Module

This function is designed for instructors. Instructors can use this to control and monitor students' PDAs. If more than one student is selected from the member list, the window shows the student list with function switches. The instructor can use these switches to control the functions of students' PDA. If only one student is selected from the member list, the window also shows the history of the student's chatting (See Figure 2.14).

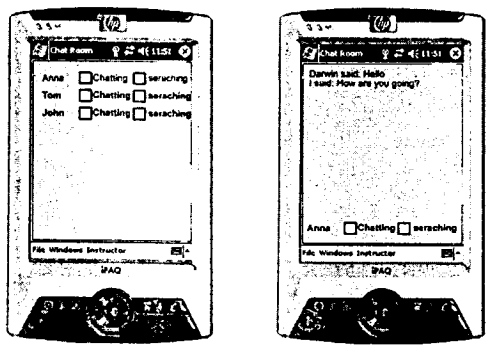

Figure 2.14: Control Module

Quiz module

We provide a function to allow instructors to send small quizzes to students when groups visit the museum. By using this function, instructors have a channel by which to test students. After receiving quizzes, the students answer them, then send their answers to the instructor. The instructor can then review the students' answers(See Figure 2.15).

Server-side Services

The server-side application is a provider of general information. Museums sometimes send general information or notices through a broadcast system. These notices usually disturb visitors' activities, are sometimes not clear, and are often difficult to understand. Through PDAs, the museum can silently send information and notices to all visitors by broadcast, or send information to select students by unicast. The server-side application also can send short quizzes, thereby allowing visitors to have an entertaining tour.

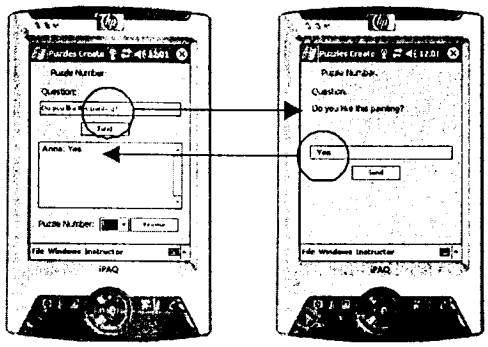

Figure 2.15: Quiz Module

2.5 Related Works on Mobile Application Architecture

There have been several attempts to support wireless application with mobile devices. However, all of them focus on a single field. Ohshima et al [44], for example, use PDAs as a lightweight method of providing a combination guidebook, map, and content for theme park guests. This application adapts the client-server framework based on the infrastructure wireless network. The main drawback of this system is that it does not provide a mechanism for visitors to communicate with each other directly. Another drawback is that if the access points or the server fail, guests cannot obtain any benefit from the PDAs. Flick Software [21], an Ottawa-based mobile software solution provider, developed a portable interactive guide to make museum visits an interactive experience. Flick Software's small interactive device with its mobile interactive guide allows users to learn about museum artifacts, find exhibits on a locator map, check IMAX schedules, and even play museum games. However, these system still focus on the individual visitor. Therefore, they do not allow interaction between mobile devices.

Chapter 3

A Unicast-based self-Adaptive Reliable Transmission Protocol

3.1 Introduction

Mobile ad hoc wireless networks are networks composed of peer-to-peer mobile nodes that communicate with each other without any pre-configuration, fixed infrastructure, or router support. Thus, no server or traditional router is used to manage network operations or communications among nodes. Furthermore, nodes continue roaming at random within these networks. They join in arbitrarily when other users leave the network without any announcement. If two nodes are within radio range of one another, an ad hoc network is formed. Hence, the routing between two nodes may consist of many hops. This kind of routing is called multi-hop routing. Due to the distinguishing feature of these networks, the globally arbitrary mobility of nodes and duple-function of nodes (acting as mobile-end device and network router), mobile ad hoc wireless networks have huge application potential, particularly in military, battlefield, rescue, and sporadic happenings coverage.

Disadvantages always come with advantages. In mobile ad hoc wireless networks, mobile nodes usually work in groups to participate in network activities. As they move relative to one another, the wireless link may be broken when other users are established. Therefore, the topology of mobile ad hoc wireless networks changes rapidly and unpredictably. This affects the stability of network routing. As we know, an unstable routing makes it impossible to provide perfect network transmission performance. Un-

stable radio signal transferring, used in wireless networks to deliver data, often suffers disturbances from other signal sources and worsens network performance. In such unstable network environments, it is unavoidable for a large amounts of packet loss and transmission latency in the data delivery process. Hence, there is an inevitable need for reliable transmission. However, due to the salient characteristics of mobile ad hoc wireless networks, it is difficult to overcome these problems. In this chapter, we present a desired solution to this challenging problem in order to make mobile ad hoc wireless networks practical and widely deployed with perfect transmission performance.

Many papers address reliable transmission issues. Some, such as the Stop and Wait Protocol [12], the Sliding Window Protocol [9], and the Selective Repeat Error Recovery Protocol [10], have already been successfully implemented in traditional wire networks. The typical representative of the Selective Repeat Error Recovery Protocol is the Transmission Control Protocol (TCP) [2], which is the main delivery protocol used in the Internet. However, TCP has not received the same good reviews as it received in traditional wire networks when it was implemented in mobile ad hoc wireless networks. Here, TCP is used as an important comparison objective to be analyzed.

Since arbitrary topology changes are rarely experienced in traditional wired networks, there are more factors to be considered when designing a new protocol for mobile wireless networks. Our proposed reliable transmission protocol, which is the Unicast-based and self-Adaptive Reliable Transmission Protocol that is denoted UARTP, in some ways resembles certain traditional techniques derived from wired data transfer protocols (i.e. the sliding window [9], the acknowledgement mechanism, and erasure code [35] [43]). However, it also has particular characteristics that are more suitable for the mobile ad hoc wireless network environment. These particular characteristics are discussed in detail in later sections.

3.2 Description of UARTP

Reliable transmission is defined as when "Data is accepted at one end of a link in the same order as was transmitted at the other end, without loss and without duplicates." [8] This implies four constraints:

- Transmission in the original order
- No data loss
- No data duplicates
- Data being received within a reasonable time interval [8]

Thus, an efficient and effective transmission protocol must satisfy these four constraints. The main motivation for this work is to provide a reliable transmission approach for dynamic, unstable mobile ad hoc wireless networks, which match the following requirements simultaneously: high delivery rate, reliability, low latency, and low overhead. Although several solutions will be discussed in this chapter, these solutions were initially designed for traditional wire networks and do not have the desired transmission requirement when applied to wireless networks. Strictly speaking, the basic idea of this new proposed transmission protocol, UARTP, is a combination of two basic techniques: Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC). A hierarchical structure efficiently integrates these two techniques together and makes this new protocol more suitable for mobile ad hoc wireless networks. This structure has two levels: the link layer that uses hop-by-hop transmission mode, and the transport layer that uses the mode of end-to-end transmission. The following paragraphs provide further analysis concerning these two modes.

- From the viewpoint of the transmission process, we know that there are two transmission modes: the hop-by-hop transmission mode and the end-to-end transmission mode ("end" refers to the end system). In the end-to-end transmission mode, there is no intermediate node participating in reliable transmission disposal. In other words, data error detection and data recovery are implemented only at the final destination node (see Fig.3.2). Hence, there is a possibility that the data packet, after transferring $n-1$ hops successfully, is lost at the n^{th} hop. In this case, all the previous $n-1$ transfers are a wasted effort [40] as we need to deliver this packet again through the same n hops. Additionally, since intermediate nodes do not participate in reliable disposal, the probability that packet loss will occur by the time the hops reach the final receiver is greater, especially when a large number of hops are involved [40]. However, hop-by-hop mode uses storing-and-forwarding mechanism. Data detection and recovery are implemented in all nodes

participating in this delivery process (see Fig.3.1). Therefore, the system tries to maintain high reliability at each hop, and also has higher reliability in reaching the final destination node than end-to-end mode. Therefore, the hop-by-hop mode is a more effective choice when protocol uses ARQ technique as data detection and recovery techniques. UARTP uses hop-by-hop mode as its transmission mode to improve the delivery rate at each hop. This is applied in the ARQ technique.

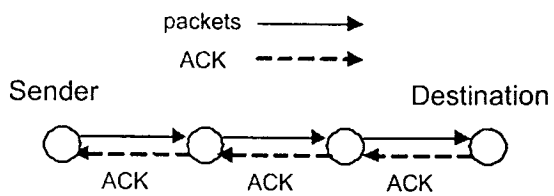

Figure 3.1: Hop-by-hop

- As a data redundance technique, FEC can be implemented either in hop-by-hop mode or in end-to-end mode. However, FEC has encoding and decoding processes that are usually implemented in the application layer, it is unavoidable that it will produce a large number of delays resulting from these processes. Furthermore, the delay is salient as the number of hops increases (see Fig.3.2). UARTP implements FEC in end-to-end transmission mode as an important approach to avoid retransmitting. Through mathematic calculation, we have estimated the reliability of the transmission for each packet through the probability of loss in the delivery process. The probability of packet loss is the important factor for UARTP to do further disposal, which is applied in FEC.

Another important issue is that UARTP classifies packet loss in two categories: random packet loss and traffic congestion. Each type of loss has different origins and characteristics. Due to these differences, UARTP uses different approaches to dispose of these. We discuss these differences as follows:

- *Random Packet Loss.* The definition of random packet loss is that data packets are lost during the transfer process for no particular reason. In fact, random

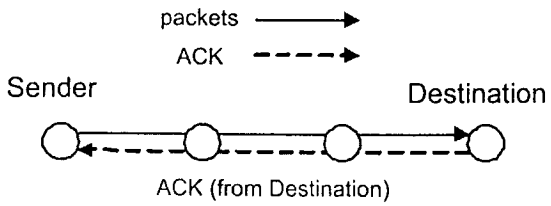

Figure 3.2: End-to-end

packet loss is unavoidable. The only thing that we can do is try to reduce the packet loss rate instead of eliminating random packet loss completely. In mobile ad hoc wireless networks, random packet loss also has relevant salient characteristics. Ad hoc networks are based on the IEEE 802.11 standards. However, radio signal have very poor transmission reliability compared to signals transmitted through wire networks. When travelling on the air, radio signals can easily suffer various disturbances from unexpected signal sources or other physical phenomena. These disturbances are the main cause of degradation of transmission performance. In this environment, packet loss is arbitrary; we cannot know which packet will be lost, and when or at which hop it will be lost. We call this random packet loss or non-congestion packet loss.

- *Traffic Congestion* The definition of traffic congestion is: an effect on networks when routers are unable to cope with the amount of traffic and start discarding packets, which then have to be retried [14]. Wireless networks have a lower bandwidth than traditional wire networks. There are no fixed or specific routers in such networks. The mobile node, as a common mobile device, must play the role of router as well. In addition, mobile devices have low performance, low channel throughput, and limited forwarding buffer. Under such a limited bandwidth and channel capacity network environment, if certain devices receive a large number of packets that exceeds its buffer capacity and the rate of incoming packets exceed the device's rate of processing, this device will start discarding these new arrival packets. Furthermore, according to the retransmission mechanism, senders keep sending packets if no acknowledgement is received (no packet received from the sender, no acknowledgement sent to the sender). In this situation, it is very probable that traffic congestion will occur, particularly when a large number of active

mobile devices co-exist. Sometimes the worst consequence occurs: networks black-out permanently. Therefore, when congestion occurs, we need to try to find a way to lower the sending rate in time in order to reduce packet delivery and thereby alleviate congestion.

As mentioned above, UARTP is a hybrid protocol of ARQ and FEC. However, the characteristics of UARTP, derived from ARQ and FEC, possess greater adaptability to mobile ad hoc wireless networks. UARTP uses retransmission and acknowledgement of ARQ, as well as erasure code of FEC, to solve random packet loss problems, and uses traffic congestion detection and probe mechanisms to solve traffic congestion problem. In order to provide a clearer view, we abstract the hierarchical structure of the architecture of UARTP contained in Fig.3.3 and Fig.3.4. The next few sections describe the main characteristics of UARTP.

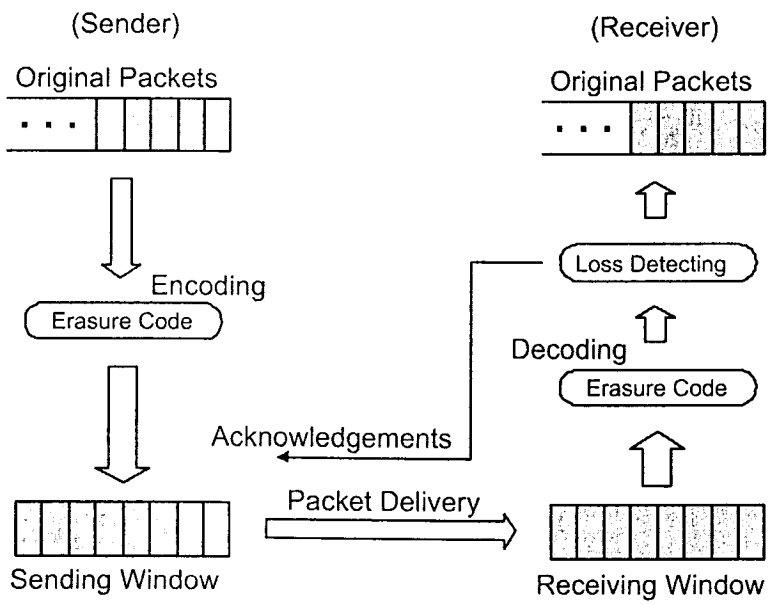

Figure 3.3: Architecture of UARTP

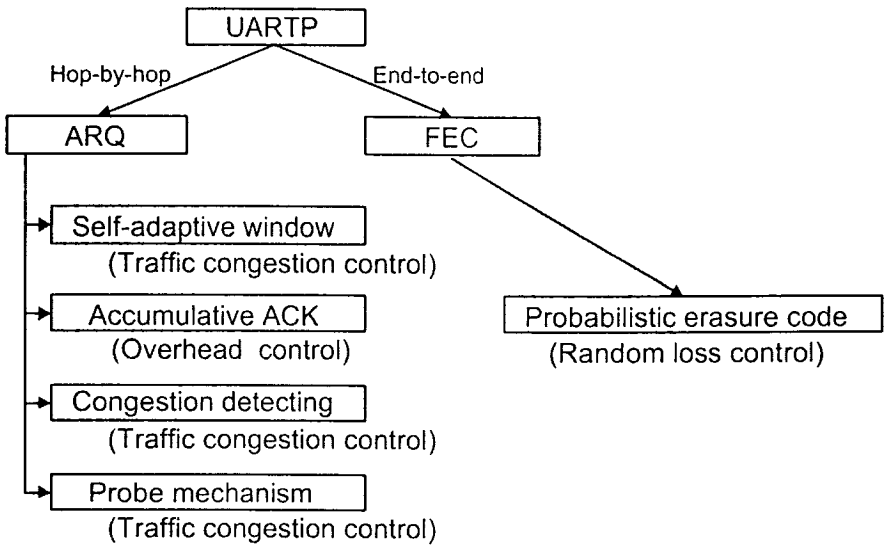

Figure 3.4: The Architecture Tree of UARTP

3.2.1 The Accumulative-acknowledgement-based Mechanism

UARTP uses the window concept of the Sliding Window Protocol [9] but has different implications. In UARTP, each node has a sending window to monitor the data packet queue, in which data packets wait to be sent. We call this data packet queue the waiting queue. The sending window considered as a whole and the packets in the sending window are sent as a batch-bulk. In other words, bulks of window-size packets are transferred to the receiver in one timeout. The window size is adaptive. We use packet loss rate as a important metric to estimate the level of loss. We record all packet loss rate according to each transmission acknowledgement. Through the probability theory, we can obtain a range of packet loss rates. If the current packet loss rate has fallen within this range, the current packet loss is acceptable. If the current packet loss rate is greater than the highest bound of this range, it is unacceptable and need to require a decrease in the size of the sending window. However, if the current packet loss rate is smaller than the lowest bound of this range, we need to consider to increase the size of the sending window. Through altering the window size, we can control the sending rate efficiently. We know the low packet loss rate and the efficiency of delivery are two opposite factors. Here UARTP tries to keep a reasonable packet loss rate in order to obtain a desired efficiency of delivery.

Recall that there is one acknowledgment for each respective data packet loss event in most traditional transmission protocols. In a limited bandwidth wireless network, such a large number of acknowledgements travelling on the air will cause the make network to become overloaded. This is another main reason for traffic congestion. In UARTP, we use an accumulative acknowledgement mechanism to avoid a large number of transmissions of redundant acknowledgements. We use only one acknowledgement for window-size bulk packet transmissions. After a timeout delay on the receiver end, the receiver places all acknowledgements together as a combined acknowledgement and sends this to the sender (see Fig.3.5). By using this mechanism, overhead is effectively reduced.

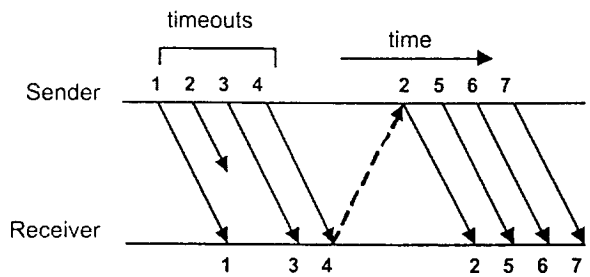

Figure 3.5: UARTP

3.2.2 Traffic Congestion Detection and Probe Packet Mechanism

According to the analysis of traffic congestion in the previous section, we know that traffic congestion occurs when the amount of incoming packets exceeds the available capacity of the receiver. The general idea solving this problem is to terminate the sending process or reduce sending frequency. Firstly, we need to know how to detect the occurrence of traffic congestion. If the sender does not receive acknowledgement within several successive timeout periods, or the current packet loss rate exceeds a specific percentage, UARTP protocol thinks that a severe traffic congestion is occurring. UARTP then enters probe status. Therefore, it terminates all sending processes. The sender uses the probe packet mechanism to continue sending probe packets to the receiver until it receives a probe response from the receiver. After receiving a probe response, the sender resumes

the sending process.

3.2.3 Probabilistic Erasure Code Mechanism

Erasure coding [35] [43] [37] is a technique of FEC for achieving high availability and reliability in communication systems using redundant packet technique. The scheme of erasure code is to encode m data packets into n fragments with an erasure code on the sender end. The receiver can reconstruct m original packets by receiving any m out of n code words ($n > m$). The value of n is the key factor needed to consider. If we select a large n comparing the packet loss rate, we can achieve high reliability without retransmission. However, the large erasure code n may result in a high number of redundant packet transmissions. A small n will result in a large number of retransmissions.

In mobile ad hoc wireless networks, if we use FEC in all sending process, particularly when FEC is used to solve random packet loss problems, we have at least $\frac{M \times (n-m) \times (1-\bar{x})}{n}$ redundant packets to be sent, where M is the amount of packets to be sent, and $\bar{x}$ is the average rate of packet loss. Hence, it is not advisable to use FEC on all sending packets to solve the problem of random packet loss. If we only dispose of those packets with the most probability to be lost during the delivery process, the amount of redundant packets will be markedly reduced.

UARTP uses the probabilistic erasure code mechanism to solve the problem of over-redundant packets. Through analyzing the features of random packet loss, we know that the number of random packet losses in interval time is poisson distribution, and each interval time between two successive random packet losses is independent and exponential distribution. Another important assumption is that each hop in random packet loss is independent. Hence, through the probability theory we can estimate probability of being lost. We use this estimation of packet loss to determine whether it is necessary to be encoded using erasure code. Using this mechanism, UARTP efficiently decreases the amount of redundant packets while still benefiting from the FEC technique.

Table 3.1: Notation

Term	Meaning
$Q_W(N)$	The waiting queue of node N in which packets are waiting to be sent
$Q_S(N)$	The sending queue of node N in which packets are being sent
$L_N(Q)$	Length of queue Q at node N
R_N	The record set of packet loss rate at node N with elements $\{r_1, r_2, ...\}$
U_R	The upper-bound range of the acceptable random packet loss rate
L_R	The lower-bound range of the acceptable random packet loss rate
p	The probability of each packet loss on each one-hop delivery
P_L	The probability of each packet loss when reaching the final destination node

3.3 The Analysis of the UARTP Model and Algorithm of UARTP

In this section, we analyze the feasibility of this protocol. First, we propose a mathematic model to study the rationality of this new protocol based on certain assumptions. Through proving these assumptions of this model, we discuss the general algorithm for the mobile pattern. We summarize the notations in Table 4.1, which we provide in this chapter.

3.3.1 ARQ

Given a source node S and a final destination node D, we assume that $p_1, p_2, , p_n$ denote packets waiting to be sent in the source node. In each node, there is a waiting queue that stores all the data packets waiting to be sent. We denote $Q_W(N)$, the waiting queue of node N. Each data packet in this queue is assigned a unique incremental sequence number. We use another queue to represent the sending window, denoted $Q_S(N)$. We assume that the initial value of the window size is equal to I. The timeouts, $T.O.S$, are equal to $RTT + K$, and the RTT is the current round trip time from the sender to the respective receiver. K is a constant delay.

At the source node N, if node N finds that there are some packets in the $Q_W(N)$ wait-

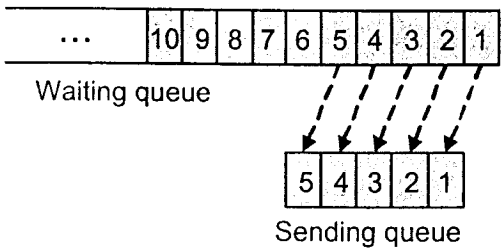

Figure 3.6: Waiting Queue and Sending Queue

ing to be sent, this node begins to continuously move packets from $Q_W(N)$ into $Q_S(N)$ until $Q_S(N)$ is full (Fig.3.6). Therefore, there are $L_N(Q_S)$ packets in $Q_S(N)$ waiting to be sent. Then this transmission mechanism sequentially transmits these $L_N(Q_S)$ packets buffered in $Q_S(N)$. After sending these packets, UARTP waits for an acknowledgement. The waiting time is equal to timeouts, $T.O.S$. If no acknowledgement is received after one timeout, UARTP retransmits these packets. If it receives an acknowledgement, UARTP updates the $Q_S(N)$. The acknowledgement from the receiver usually notifies which packets the receiver has received. UARTP removes the packets from $Q_S(N)$, which have already been received by the receiver, and fills up with new packets (see Fig.3.5).

Meanwhile, UARTP protocol records the respective packet loss rate r_i according to i_{th} acknowledgement. We have a record of random packet loss rate R_N at node N, where $R_N = \{r_i, i = 1, 2, 3, ...\}$. Here we formulate one definition and one lemma.

Definition 1 In a one-hop delivery, the occurrence of random packet loss event P is *an independent random variable* of the discrete type that satisfies *bernoulli distribution* with the probability p. The random variable P also satisfies the following properties:

1. $P(success) = 0$ and $P(failure) = 1$.
2. The expected value $\mu = p$.
3. The variance $\sigma_P^2 = p(1-p)$.
4. The standard deviation $\sigma_P = \sqrt{p(1-p)}$.

Lemma 1 The random packet loss rate R is *an independent random variable* with

1. Expected value $\mu_R = \mu_P$
2. Variance $\sigma_P^2 = \frac{\sigma_P^2}{n}$

where μ_P is the expected value of random packet loss event P, σ_P^2 is the variance of random packet loss event P, and n is equal to $L_N(Q_S)$, the number of packets in sending window Q_S.

Proof. Assume that n packets are being sent. From **Definition 1**, we have $P_1, P_2, ..., P_n$ which are n independent random variables P with the same expected value μ_P and variance σ_P. Therefore, we have the packet loss rate

$$R = \frac{P_1 + P_2, ..., + P_n}{n}$$

From Theorem 6.2-3 in [41], we know that $R = \sum(\frac{1}{n}P_i)$ is also the independent random variable with

1. $\mu_R = \sum(\frac{1}{n}\mu_P) = \mu_P$
2. $\sigma_R^2 = \sum(\frac{1}{n})^2\sigma^2 = \frac{1}{n}\sigma^2$

From **Lemma 1**, we know that the value of random variable R is dependent on the size of sending window. However, the size of sending window is adaptive. Therefore, we need to re-calculate the acceptable range of the random packet loss rate after the size of the sending window is adapted. There are two phases to calculate this range:

1. The initial phase. Since there is no record about the random packet loss rate, we obtain the expected value of the random packet loss μ_R and the variance of the random packet loss σ_R^2 from **Lemma 1**, where probability of packet loss p is the average value of random packet loss. Then we obtain the two bounds:

$$L_R = \mu_R - \sqrt{\sigma_R^2} = \mu_R - \sigma_R$$

$$U_R = \mu_R + \sqrt{\sigma_R^2} = \mu_R + \sigma_R$$

Table 3.2: Analysis of Random Packet Loss Rate

Value	Analysis
$r_i > U_R$	The new loss rate is not acceptable. Thus it is thought of as traffic congestion, the size of the sending window is decreased by 1
$L_R \leq r_i \leq U_R$	The new loss rate is acceptable and is recorded
$r_i < L_R$	Consider increasing the sending rate

2. The operating phase. In this phase, we already have a series of records of random packet loss rate. Here, we use the sample mean and sample variance concepts to calculate the range of sample variance, s^2. Thus the sample mean is:

$$\bar{r} = (r_1 + r_2 +, ..., +r_n)/n = \frac{1}{n}\sum r_i$$

and the sample variance is:

$$s^2 = \frac{1}{n-1}\sum(r_i - \bar{r})^2$$

where $i = 1, 2, ..., n$, n is the amount of records of random packet loss rate at the current value of $L_N(Q_S)$.

The two bounds are:

$$L_R = \bar{r} - \sqrt{s^2} = \bar{r} - s$$

$$U_R = \bar{r} + \sqrt{s^2} = \bar{r} + s$$

Now, we can compare the current random packet loss rate r_i with these two bounds. There are three cases analyzed in Table 3.2:

A question emerges: If the size of the sending window is tuned too frequently, a large number of probabilistic calculations are needed. Such a mass of calculations will result in the reduction of efficiency. However, we know that this window adaptability mechanism is used to control random packet loss. In a stable external environment, random packet loss has a relatively stable rate that remains within a small range.

Another question is how to determine which packet loss rate is an acceptable random packet loss and which packet loss rate constitutes unacceptable traffic congestion. In UARTP, we use two criteria to estimate traffic congestion:

1. UARTP does not receive acknowledgement for successive transmission rounds, and
2. The current packet loss rate exceeds 50 percent.

If UARTP matches either of these criteria, it enters probe status. First, UARTP freezes all sending processes. Then UARTP repeatedly sends a probe packet to the receiver and waits for a probe response. If the UARTP receives a response from the receiver, it resumes the regular transmission process. Through traffic congestion detection mechanism, we can keep the packet loss rate below 50 percent. This is also the important characteristic of UARTP.

3.3.2 The Probability Model of UARTP

From **Definition 1**, we know that the occurrence of random packet loss event P is *an independent random variable* of the discrete type that satisfies *bernoulli distribution.* Therefore, we have another lemma as follows.

Lemma 2 The occurrence of random packet loss is an *approximate poisson process.* The number of random packet loss events that occur in a given continuous interval is a *poisson distribution* with λ.

Proof. First, we assume that the packet delivery is a transient process, which is implemented on a one-by-one basis. Then, we suppose that one and only one packet is delivered in a sufficiently short interval. Each interval time is non-overlapping and independent. According to **Definition 1**, the event of random packet loss is *bernoulli distribution.* Therefore, from Definition 3.5-1 in [42], we have **Lemma 2.**

Lemma 3 The interval T between two successive random lost packets is *exponential distribution.*

We have p.d.f. of *exponential distribution*

$$f(x) = \frac{1}{\theta} e^{\frac{-x}{\theta}}$$

where x is the sequence number of packet; $\frac{1}{\theta}$ is the number of occurrences of random packet losses per a basic unit; and the basic unit is the average random packet loss rate, which is equal to the total number of packet losses divided by the total number of packets.

From this function, we can obtain the random loss probability of packet, denoted p, in one-hop delivery. According to the end-to-end characteristic mentioned in the **Description of UARTP** section, we need to estimate the probability of random packet loss for whole transmission process, namely from the source node to the final destination node. First, we summarize the known conditions:

1. The probability of one-hop random packet loss p
2. Each packet delivery is independent
3. Each hop delivery is independent

Using these three conditions, we obtain another lemma.

Lemma 4 In mobile ad hoc wireless networks, packet delivery in the multi-hop delivery process can be represented by a homogeneous *Markov chain.*

Proof. We consider each mobile node as a state. The packet is delivered through several states, from state i_1 to state i_n. Since each hop delivery is independent, the probability of one one-hop random packet delivery, $q = 1 - p$, also is independent. Therefore, we obtain

$$P(X_n = i_n | X_{n-1} = i_{n-1}, X_{n-2} = i_{n-2}, ..., X_1 = i_1) = P(X_n = i_n | X_{n-1} = i_{n-1}) = K_{i_{n-1}, i_n}$$

According to Definition 5.1.1 in [33], this lemma is proven.

We give a further analysis on packet delivery represented by the *Markov chain.* Given a source node S and a destination node D, let $N_1, N_2, ..., N_k$ denote the path between S and D. Each hop has a different delivery success probability q_i, $q_i = 1 - p_i$ (see Fig.3.7). The probability transition kernel of the *Markov chain* is shown in Fig.3.8. From [33], we obtain the element of transition kernel

$$K_{ij} = P(X_n = j | X_{n-1} = i) = \frac{P(X_n = j, X_{n-1} = i)}{X_{n-1} = i} \ (*).$$

From the previous analysis on multi-hop delivery and Fig.3.7, we know that a successful multi-hop delivery can be represented using probability formula as follows:

$$P(\text{successful multi-hop delivery}) = P(X_n = D, X_{n-1} = N_k, ..., X_3 = N_2, X_2 = N_1, X_1 = S).$$

According to formula (*) and item 3 (Each hop delivery is independent), we have:

$$P(\text{successful multi-hop delivery}) = K_{N_k,D} \cdots K_{N_2,N_3} K_{N_1,N_2} K_{S,N_1}$$

where $K_{N_k,D} \cdots K_{N_2,N_3} K_{N_1,N_2} K_{S,N_1}$ are the respective successful delivery probabilities of each hop along the path from the source node to the destination node. Finally, we can obtain the probability of each packet loss in multi-hop delivery, which is equal to $1 - P(\text{successful multi-hop delivery})$, denoted P_L. Now we use P_L to compare with the average random packet loss rate, which is equal to the total number of packet losses divided by the total number of packets. If only P_L is bigger than the average random packet loss rate, this packet will be disposed of using the erasure code mechanism.

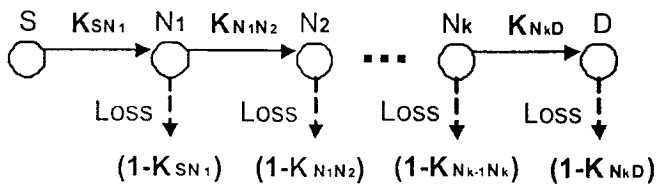

Figure 3.7: Multi-hop Delivery

State	S	N1	N2	. . .	Nk	D	Loss
S	0	KSN1	0	. . .	0	0	(1-KSN1)
N1	0	0	KN1N2	. . .	0	0	(1-KN1N2)
N2	0	0	0	KN2N3 . .	0	0	(1-KN2N3)
.							.
.							.
.							.
Nk	0	0	0	0 . .	0	KNkD	(1-KNkD)
D	0	0	0	0 . . .	0	1	0
Loss	0	0	0	0 . . .	0	0	1

Figure 3.8: Transition Kernel of Markov Chain

3.3.3 Probabilistic Erasure Code Encoding Mechanism

The general idea of erasure code is to use a $n \times m$ matrix, denoted A, as the encoding generator. Let $X = x_1...x_m$ be the vector of original packets, and $Y = y_1...y_n$ be the vector of the encoded code words of the original packets. The encoding process can be represented by a matrix-vector multiplication, $AX = Y$, or by an encoding function, $A(X) = Y$. The decoding process can be represented by $A^{-1}Y = X$ or $A^{-1}(Y) = X$, which A^{-1} is the inversion matrix of A. A simple type of erasure code is linear code. Encoding matrix A has the property that $A(X_1) + A(X_2) = A(X_1 + X_2)$. According to the principle of FEC, we need to obtain any subset of m code words to recover the m original packets. Therefore the generator, matrix A, should have at least m linearly independent rows so that the linear equation system $A^{-1}Y = X$ has a unique solution $X(x_1...x_m)$, which represents the unique original packets [40]. The core problem is how to build the matrix A and how to select the value of n. First, we need to keep the matrix A with rank m. This means that any set of m rows forms a non-singular matrix, or the determinant of the matrix is non-zero. There are a great deal of erasure code algorithms that have distinct characteristics. In UARTP, we use the *Reed-Solomon Code* [40]. In the Reed-Solomon Code, the element of matrix $A(i,j)$ is equal to x_i^{j-1} where x_i is non-zero and different from all other variables(see Fig.3.9).

$$\begin{vmatrix} 1 & X_1 & \cdots & X_1^{m-1} \\ 1 & X_2 & \cdots & X_2^{m-1} \\ \vdots & \vdots & & \vdots \\ 1 & X_{n-1} & \cdots & X_{n-1}^{m-1} \\ 1 & X_n & \cdots & X_n^{m-1} \end{vmatrix}$$

Figure 3.9: The Reed-Solomon Matrix

Since UARTP uses the probabilistic and selective erasure code encoding mechanism, only packets which have a specific probability of loss rate will be encoded. We use the double-encoding mechanism. For example, if we have m packets that need to be en-

coded, we use a matrix with $n \times m$ where $n - m = m$. In addition, we use a half-window fault-tolerating mechanism. For example, if the size of the sending window is equal to $L_N(Q_S)$, and through calculations we know that probability model of packet p_i needs to be encoded, we encode the packet p_i as well as $L_N(Q_S)/2-1$ packets which are around the packet p_i. So we have packet $(p_{i-(L_N(Q_S)/2)+1}, \cdots, p_{i-1}, p_i, p_{i+1}, \cdots, p_{i+(L_N(Q_S)/2)-1}, p_{i+(L_N(Q_S)/2)})$ encoded, which is the total of $L_N(Q_S)/2$, half the size of the sending window packets that need to be encoded. After encoding, the code words become n vector where n is equal to $L_N(Q_S)$, the size of the sending window (based on the double-encoding mechanism). The receiver, if receiving more than any subset of m code words, the receiver can recover m original packets (see Fig.3.10). Hence, in one window-size sending process, if the packet loss rate does not exceed 50 percent, no retransmission occurs. But if the current packet loss rate is more than 50 percent, according to the criteria of traffic congestion mentioned in the ARQ section, UARTP will enter probe status. Therefore, using the double-encoding mechanism and the half-window fault-tolerating mechanism, we can guarantee that there is no retransmission occurring in these erasure codes. If two packets need to be encoded, and the distance between them is less than the window size, these two packets will be encoded together.

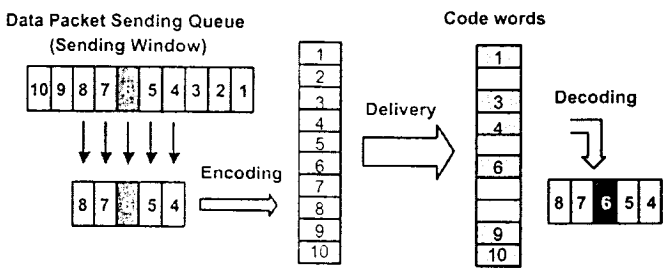

Figure 3.10: The Double-encoding and Half-window Fault-tolerating Mechanisms

3.4 The Mobility Characteristics of UARTP

Mobile ad hoc wireless networks are composed of mobile nodes that roam randomly in wireless network environments when certain intermediate nodes within the networks act as routers to forward data to their destination. Therefore, a successful transmission process needs a source node, a destination node, and the cooperation of a number of

intermediate nodes. Since the movement of nodes is random, we are unable to predict the next mobility-position of each node. When such nodes leave the network or for an unknown reason cannot provide service correctly, we must update our routing table immediately and obtain the newer routing information on time in order to guarantee the completion of this transmission. However, routing discovery and path repair are usually time-consuming processes that sometimes fail. During this period, there is no routing information existing, or incorrect routing information is used by these nodes. Under these circumstances, packets are generally discarded or lost during the transmission process. Therefore, the more rapidly the nodes move, the more rapidly the routing change.

In addition, when the frequency of mobility is increased, packets may use different paths to reach the destination. Conditions of these paths are various depending on the diverse environments and the nodes' diverse performance.

Due to these mobile characteristics of the mobile ad hoc wireless network, we need to make more considerations to guarantee that data packets reach desired destination nodes completely and successfully. Therefore, it is indispensable to establish certain new rules for the UARTP to acclimatize mobile network environments.

- *Rule 1.* Retransmissions are always sent to the next node that is oriented to the destination node instead of to the lower node in the last transmission process. In the mobile ad hoc wireless network, due to the mobility of nodes, these nodes may be different.

- *Rule 2.* Acknowledgements are always sent to the next node that is oriented to the original source node instead of to the upper node in the last transmission process. Only the acknowledgement from the final destination node is transferred to the original source node. And only after receiving the acknowledgement from the final destination node can the original source node update the sending queue.

- *Rule 3.* Congestion detection is only available for neighborhood nodes. If one neighborhood node, which is locked due to the congestion, moves out, and more than one hop is involved, the locked status is cancelled.

- *Rule 4.* After sending a bulk of data packets, a node is in waiting status and waits for an acknowledgement. If no acknowledgement is received, the node checks whether the receiver is still one of its neighborhood nodes. If it is, this node performs its regular process and retransmits data packets. If not, the node restarts its retransmission process to the next node that is oriented to the final destination node.

- *Rule 5.* After receiving an acknowledgement, an intermediate node will check the local buffer to see if there are certain packets that the receiver wishes to have. If there are, the intermediate node retransmits these packets to the sender of the acknowledgement. If no packets are desired, the node sends an acknowledgement to the next node that is oriented to the original source node and cancels former related transmission operations in which this node participated before.

- *Rule 6.* The probabilistic erasure code estimates the probability of packet loss through the number of hops. However due to the characteristics of mobile ad hoc wireless networks, the hops between any two nodes are variable. So if the packets determine that the number of remainder hops is zero when reaching an intermediate node, this packet needs to re-estimate the probability of packet loss with the number of hops from this intermediate node to the final destination node.

In the following paragraphs, we discuss the mobile characteristics of this algorithm in detail by addressing a simple example. In this example, we assume that wireless topology remains unchanged during timeouts. This assumption guarantees that all packets in the sending window are kept in the same node.

- *Initial state.* Suppose that there are 5 nodes in our ad hoc network (node S, node 1, node 2, node 3, and node D). The initial topology is shown in Fig.3.11. Now node S has 5 packets in the sending window that must be sent to node D. We denote S_i $(i = S, 1, 2, 3, D)$ as a set of packets waiting to be sent. So in initial state, $S_S = (p_1, p_2, p_3, p_4, p_5)$, and $S_1, S_2, S_3, S_D = 0$. We know that in this state $S_S >= S_1 >= S_2 >= S_3 >= S_D$

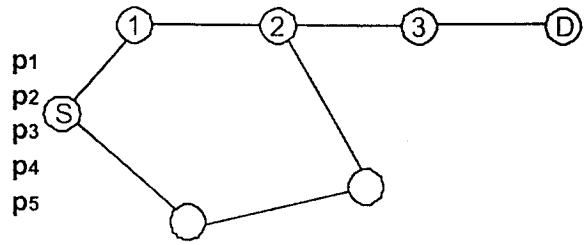

Figure 3.11: Initial State

- *State 1.* During a timeout, node S sends all packets in its sending window $P_S(p_1, p_2, p_3, p_4, p_5)$ to node 1, which is the next node oriented to destination node D. Node 1 receives only some of them (p_1, p_2, p_3). Node 1 buffers these packets and transfers them to the next node, node 2, before the end of the timeout of node 1. The sets of each node are shown in Fig.3.12.

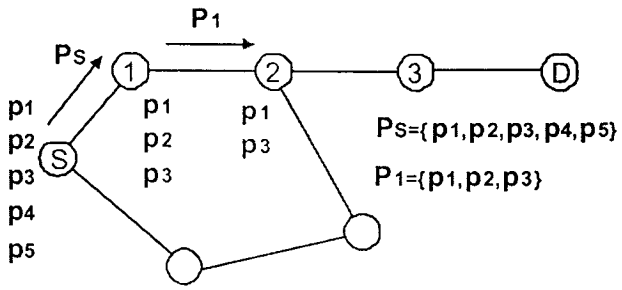

Figure 3.12: State 1

- *State 2.* A transient move occurs. The new topology is shown in Fig.3.13. After a timeout, each node that has received data packets from source node S begins to send an acknowledgement to source node S (according to rule 2). When the acknowledgment (ACK 1 (1, 2, 3) of node 1 passes node 2, it finds that node 2 has already acquired a retransmission from source node S by sending its acknowledgement (ACK 2 (1, 3)). According to rule 5 (namely $S_1 > S_2$), the retransmission requirement of node 2 and other related transfer processes is cancelled, and ACK 2 is replaced by ACK 1. (Shown in Fig.3.14)

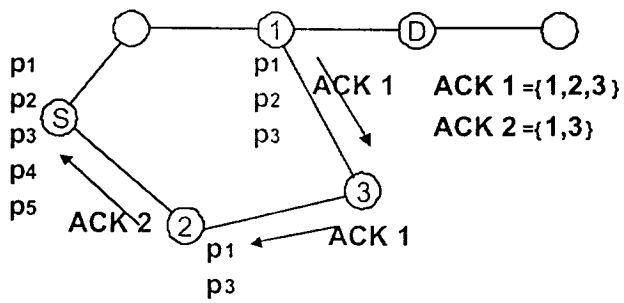

Figure 3.13: State 2-a

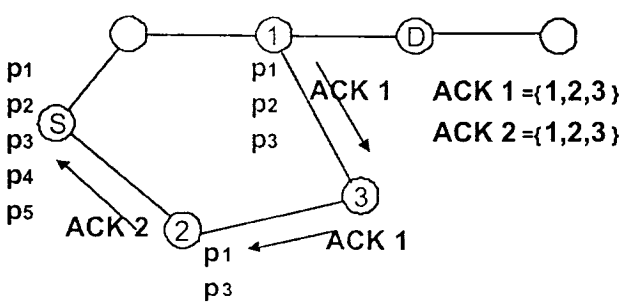

Figure 3.14: State 2-b

- *State 3.* According to rule 1, node 1 transfers current data packets $P_1(p_1, p_2, p_3)$ to destination node D. We assume that destination node D receives packets p_1, p_3 from node 1. Then, according to rule 2, destination node D sends an acknowledgement to the source node along the path (node 1, node 3, node 2). Meanwhile, source node S retransmits packets $P_S(p_4, p_5)$ to node 1 along node 2 and node 3. Node 2 and node 3 also send acknowledgements to the respective nodes. (Shown in Fig.3.15)

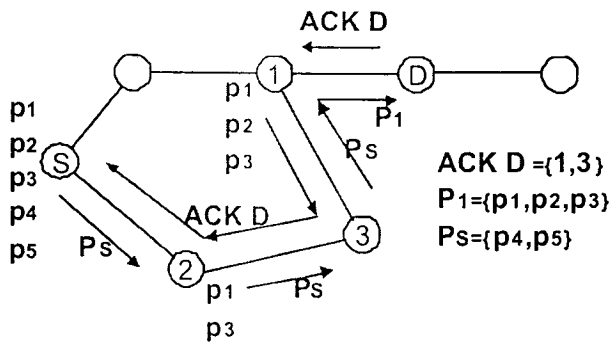

Figure 3.15: State 3

- *State 4.* According to rule 2, after source node S receives acknowledgement from node D, ACK D (1, 3), it updates its respective sending queue and starts a new transmission round. The new sending window is $P_S(p_2, p_4, p_5, p_6, p_7)$. (Shown in Fig.3.16)

- *In addition.* Before receiving acknowledgement from node 1 (mentioned in state 2), node 2 attempts to perform the same process as node 1. Node 2 transfers these packets it has received, $P_2(p_1, p_3)$, to the next node, node 3, then node 1. When these packets reach node 1, node 1 discovers that the packets already exist in the local buffer. So node 1 discards these data packets. And when node 2 sends an acknowledgement, ACK 2 (1,3), to source node S, source node S retransmits the respective packets. But after receiving the acknowledgement from node 1, node 2 terminates all related processes.

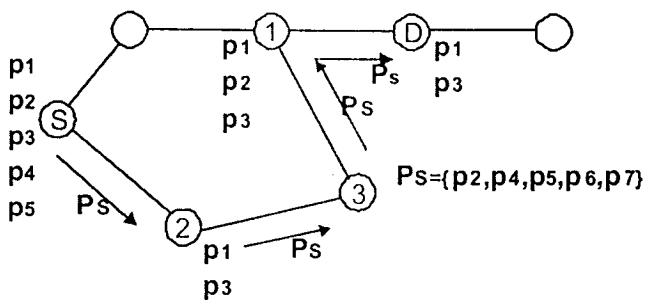

Figure 3.16: State 4

3.5 Simulation Result and Performance Evaluation

The simulation is built om Network Simulator Version 2 (NS2) platform [15]. We use DSDV as the mobile ad hoc routing protocol. The simulation area is $500m \times 500m$. The transmission range of each mobile node is $250m$. A specific data packet source continuously unicasts 512-byte data packets to a specific destination node when the other 50 percent of nodes send packets at random. Network bandwidth is 2 Mbit/sec. MAC protocol is 802.11. We simulate 10 situations in which there are 10 nodes, 20 nodes,...,90 nodes, and 100 nodes operating in this environment. The simulation time is 900 seconds. We analyzed three protocols: UARTP, TCP, and UDP, and obtained respective data, which is shown in Fig.3.17. From Fig.3.17, we have very good performance of delivery rate. We can keep a minimum $0.68 percent$ reliable transmission rate even in 100-node mobile ad hoc wireless networks. Fig.3.18 shows the delivery delay of each protocol. UDP is the best, but its delivery rate is poor, especially in large scale wireless networks. Through analyzing, we find that the main reason for the delay of UARTP results from the probe status; namely, traffic congestions occur. So UARTP waits for relieving a traffic congestion. In general, UARTP has a very salient performance of delivery rate with a stable and acceptable delay.

3.6 Related Work

Roughly speaking, a reliable transmission protocol provides the ability to detect errors and correct them in the shortest possible time interval. This ability to detect errors and

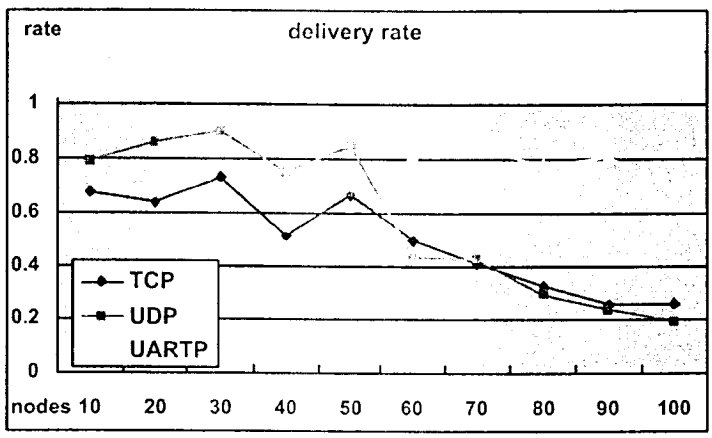

Figure 3.17: Delivery Rate

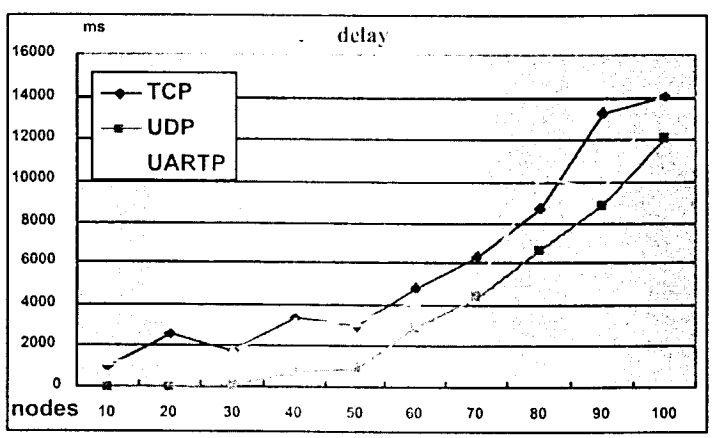

Figure 3.18: Delivery Delay

correct errors has classically been provided via two methods: Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC) [32].

3.6.1 Automatic Repeat reQuest (ARQ)

Most unicast protocols are usually accomplished by using the Automatic Repeat reQuest mechanism (ARQ) and the data retransmission mechanism. ARQ is a receiver-oriented reliable transmission technique that involves receiver packet loss notification to senders and explicit requests for retransmission of lost packets [32]. ARQ has two important components: acknowledgements and timeouts.

An acknowledgement (ACK) is a small control packet that is sent by the receiver to inform the sender that the data packet previously sent by the sender has been received. The reception of acknowledgement indicates that there has been either a successful packet transmission or a successful round trip. Through sending acknowledgement packets, receivers obtain a channel to communicate with senders. Senders continue retransmitting packets that are queued in the sending queue in accordance with timeouts or missing packets that are not listed in the acknowledgements received from receivers. An extended version of ACK is negative acknowledgement (NACK), which explicitly informs senders which packet is the next one that receivers are to expect.

A timeout is defined as an event that occurs when one network device expects to hear from another network device within a specified period of time, but does not. The resulting timeout usually results in a retransmission of information or the dissolving of the session between the two devices [13]. How to select a reasonable value of interval time is an important concern. If timeouts are too large, there are too many delays between the two successive sending processes and such accumulative delays result in the low transmission efficiency (Fig.3.19). However if timeouts are too small, this will result in a redundant retransmission process. The reason is for this is that the timeouts expire and retransmission resumes before acknowledgement arrives (Fig.3.20). Usually, we select timeouts depending on round-trip-time (RTT). Round-trip-time (RTT) is defined as the time interval needed for network communication to travel from the sender to the receiver and back again[11].

We review two typical transmission protocols that use the Automatic Repeat reQuest

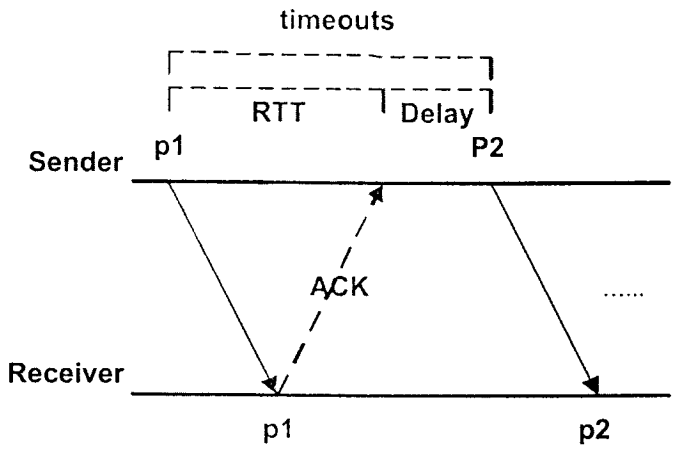

Figure 3.19: Large Timeouts

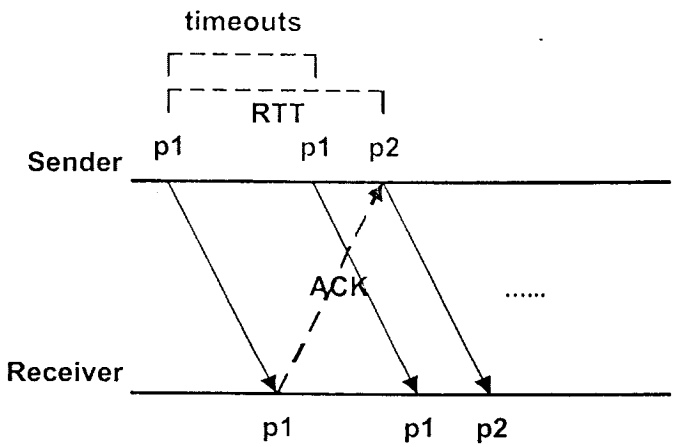

Figure 3.20: Small Timeouts

(ARQ) techniques: The Sliding Window protocol [9] and the Selective Repeat Error Recovery protocol [10], as follows.

The Sliding Window Protocol

The Sliding Window protocol uses the window concept and the sequence number mechanism. Each data packet is assigned a unique sequence number. At the sender end, all data packets are stored in the sending queue with the order of the incremental sequence number. A virtual window is mounted on the sending queue and continues sliding. The packets in this sliding window are those which are being processed currently; they need to be sent to the receiver. Two indices are used to determine the sender window size and position. One index, denoted FPS (First Packet Sent), presents the sequence number of the first data packet sent in the current sending queue. Another index, LPS (Last Packet Sent), denotes the maximum sequence number of data packets that the sender can transmit. By using these two indices, we form a virtual window or boundaries. The window size is equal to $LPS - FPS + 1$, and packets between these two indices are allowed to be sent to the receiver. The sending rate is equal to $RTT/(LPS - FPS + 1)$. We use window size to control the sending rate according to the current bandwidth. All the packets that the sender can transmit are the packets with sequence numbers between FPS and LPS. Each packet, whose sequence number belongs to this sending window, associates a respective timeout. When one packet has not received an acknowledgement after a timeout, this packet will be retransmitted. Only when the FPS receives an acknowledgement will the sender window move forward: $FPS = FPS + 1$ and $LPS = LPS + 1$.

At the receiver end, there are also two indices that we need to be mentioned: NPE (Next Packet Expected), the next packet sequence number that the receiver expects to receive; and LPA (Last Packet Acceptable), the maximum packet sequence number that the receiver can receive. These two indices are also bound to a receiving window. Therefore, the receiving window size is equal to $LPA - NPE + 1$. All packets the receiver can receive are those with sequence numbers between NPS and LPA. When receiving a packet, the receiver checks the sequence number of this packet to see if its sequence number belongs to the receiving window. If the sequence number does, the receiver buffers this packet in its receiving queue. If not, the receiver discards the packet. The receiver buffers all packets whose sequence number belongs to the receiving window, but only acknowledges the receipt of the NPE first, even if other packets with higher

sequence numbers have already been received. Afterward, the receiver window will be tuned: $NPE = NPE + 1$ and $LPA = LPA + 1$. The Sliding Window protocol is shown in Fig.3.21.

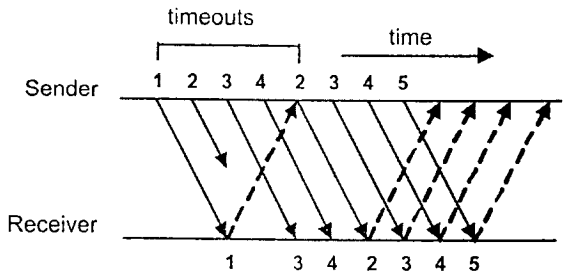

Figure 3.21: The Sliding Window Protocol

The Selective Repeat Error Recovery Protocol

The Selective Repeat Error Recovery protocol is one of the most complex transfer protocols. It uses Go-Back-N [6], Polling [7], and the selective error acknowledgement mechanism to improve transmission efficiency. When the sender has packets to be sent, each packet is assigned a unique incremental sequence number. Both ends (the sender end and the receiver end) have a buffer to store the packets to be sent or the receiving packets. At the sender end, all the packets are queued in the sending buffer and a steady flow of data packets are transmitted by the sender to the receiver [7]. At the receiver end, the receiver receives packets and stores them in the receiving buffer. If the receiver finds that the sequence numbers of packets it has received are discontinuous, it thinks that a loss is occurring. The receiver then sends a NACK to inform the sender which packets are missing and which packets it wishes to have. After explicitly receiving this acknowledgement, the sender terminates the current transmitting process and uses Go-Back-N to retransmit the missing packets and then resumes its transmitting operation of new packets until the retransmitted packets are acknowledged within the predefined time interval or another missing packet request is received [10]. If the receiver successfully receives the missing packets, the packets in the receiving queue are ordered in sequence. The receiver then sends an acknowledgement to the sender, which informs the sender that the missing packets have been received. The Selective Repeat Error Recovery protocol is shown in

Fig.3.22.

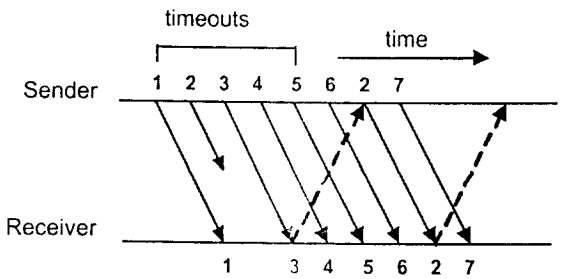

Figure 3.22: Selective Repeat Error Recovery Protocol

If in the predefined time interval the sender does not receive any acknowledgement of last retransmission, the sender are transferred to polling status and terminates the current transmitting process. Then the sender begins the transmission of a control packet indicating the current status, and requesting receiver status [7]. The control packet is sent periodically until the sender receives a response concerning the successful transmission of the control packet from the receiver. After receiving the control packet, the sender resumes its transmitting operation. Through the polling mechanism, the sender synchronizes its sending status with the receiver's receiving status.

3.6.2 Forward Error Correction (FEC)

Forward Error Correction (FEC) is a data redundant mechanism. It provides a method of obtaining error control in data transmissions by sending redundant data at the sender end and recognizing it at the receiver end [5]. The basic concept is that the sender prevents packet losses by encoding a certain amount of redundant information into the original packets using an encode generator, which allows the receiver to reconstruct the missing data packets using a decode generator without further communication with the sender [37]. Because the FEC does not require handshaking between the sender and the receiver, it is a unidirectional transfer mechanism. Therefore, FEC is attractive for multi-cast application. Since FEC works by adding redundant packets, the more redundant packets are added, the more errors are corrected, and the more overhead or control

packets are produced.

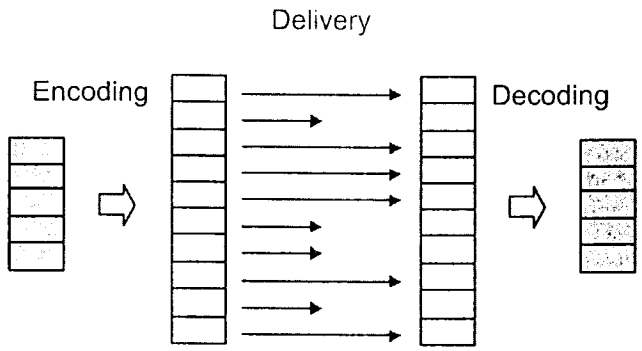

Figure 3.23: Erasure code

This FEC coding mechanism is usually called erasure code. Fig.3.23 shows a visual description of the erasure code mechanism. The key principle behind erasure code is encoding algorithms; namely, how to encode m original packets to produce n code words ($n > m$). Through such a way, any subset of m encoded code words suffice to reconstruct the original packets. Choice of n and m are also important issues. If n is too small (i.e. $n = m + 1$), there is only one redundant packet which is used as the sum of all the original packets. So it is only used for very low packet loss rate. If m is equal to 1, erasure code simply makes the same packet retransmit n times. This is another class of overhead. Therefore, we need to have at least $m > 1$ and $n - m > 1$ [37].

In summary, all of the transmission techniques mentioned above are the basic elements of popular data delivery protocols. However, these transmission techniques were initially designed for traditional wire networks. Therefore we need to use deeper consideration if we want to obtain satisfactory transmission performance in mobile wireless network environments. In this chapter we describe UARTP, a combination algorithm that uses window concept, accumulative acknowledgement, and probabilistic erasure code mechanisms. Compared with certain traditional techniques, UARTP has salient characteristics, which are analyzed as follows.

UARTP v.s. Selective Repeat Error Recovery Protocol with Respect to the Acknowledgement Mechanism

The UARTP and the Selective Repeat Error Recovery Protocol have different acknowledgement mechanisms. The Selective Repeat Error Recovery protocol uses the loss-event-based acknowledgement mechanism. When the receiver detects the packet with a discontinuous sequence number, it thinks that packet loss is occurring. The receiver then sends a missing packet request to query the missing packets (see Fig.3.22). In this protocol, if the network has a stable and perfect transmission performance and little packet loss is occurring, the Selective Repeat Error Recovery protocol has very good transmission efficiency and very low overhead because few acknowledgement packets need to be sent. However if this protocol is used in mobile ad hoc wireless networks, it may produce a number of acknowledgements because packet losses occur more frequently.

In contrast, UARTP uses an accumulative acknowledgement mechanism that is based on the window technique. At the sender end, a bulk of window-size packets is sent at the current sending rate and position in the current window. When the receiver receives packets, it knows the start and end time of the current sending period from the sending rate and respective position of the arrival packets. Hence, the receiver can estimate when the current sending process ended. When the receiver estimates the sending process end time, the receiver sends an acknowledgement to the sender. The sender reorganizes the sending window, removes packets that the receiver has received and adds new packets until the window is full. Afterward, the sender begins the next sending process. So no matter how bad the networks are, there is always acknowledgement of window-size packets. In such unstable, low bandwidth mobile ad hoc wireless networks, UARTP efficiently reduces overhead compared to the Selective Repeat Error Recovery protocol.

UARTP v.s. the Selective Repeat Error Recovery Protocol with Respect to Traffic Congestion

UARTP and the Selective Repeat Error Recovery protocol have different traffic congestion detection and recovery mechanisms. The Selective Repeat Error Recovery protocol cannot distinguish random packet loss from traffic congestion. If the Selective Repeat Error Recovery protocol does not receive acknowledgement after retransmitting the missing packets, it immediately enters "polling" status [10] [7]. However, in mobile wireless

networks, many packet losses occur at random without any particular reasons. Hence, if all failures of packet retransmitting are processed using "polling", this may result in very poor transmission efficiency.

In UARTP, the reasons for packet loss are classified into two types: random packet loss and traffic congestion. If common packet loss occurs, UARTP simply uses the packet retransmitting mechanism. Packets are retransmitted periodically until an explicit acknowledgement is received from the receiver. In addition, UARTP uses its particular traffic congestion detection mechanism to find traffic congestions. If traffic congestion occurs, UARTP relieves the congestion through changing window size and by means of probe mechanism.

Chapter 4

The Mobile Peer-to-peer System

4.1 Introduction

The evolution of peer-to-peer system architecture comprises the development that took place during the transition from the centralized system (from the late 90s) to the decentralized system (from 2000) to the current mixture system (from 2002 to the present) [25]. Up to now, most peer-to-peer systems have been implemented successfully in the traditional wire network, particularly on the Internet, in applications such as Napster [3], Gnutella [1], Chord [27] [26], Tapestry [30] [16], and JXTA [34]. Generally speaking, the peer-to-peer system, commonly known as P2P, has moved away from the more traditional client-server mode to a network where each participating device acts as both client and server [19]. Here we refer to these devices as peers. In the peer-to-peer system, each peer has no more functions than the other peers in the system. This means that each peer has the same priority to join and leave the system or access the system source. Hence, an excellent system platform for distribution of data storage, file sharing, and network computing is provided. In fact, as a pioneer in network technology, Napster's efforts to provide a service for the exchange of MP3 music files spawned a number of academic and industrial projects aimed at this area.

In traditional wire network environments, P2P systems today have matured to the point where they show very good performance in either network stability or system efficiency. However, this applies only to network environments with wide-band Internet connections, relative peer reliability, and high processing and memory capacity [25]. With the widespread of wireless technologies, more and more applications are emigrating in

mobile environments, especially in mobile ad hoc mode. Since they are characterized by low and high variable bandwidth, high variable latency, and unpredictable topology changes, mobile ad hoc wireless networks need a novel system architecture to make them more practical and widely deployed with perfect implementation performance. When a fusion of two familiar technologies is considered, a new system architecture emerges: *Mobile P2P*. Unlike traditional P2P systems, mobile P2P is defined as a P2P system built on wireless network environments. In this chapter, we constrain these wireless network environments to mobile ad hoc mode.

As a fresh technology, Mobile P2P encounters a great number of new challenges brought about by the mobile environment that must be dealt with. After analyzing the benefits and drawbacks of existing P2P technologies, we present our proposed self-adaptive hybrid P2P system architecture based on the mobile agent,denoted MAHP2P. This architecture is a triple-layer hierarchy. Self-adaptability is implemented through the server-aware mechanism, which adapts respective system mode depending on the awareness of the central server existing in the system. The mobile agent, as an intelligent software entity, efficiently reduces network traffic and improves fault tolerance. Other characteristics are also taken into account to accommodate mobile ad hoc wireless networks.

4.2 Motivation

4.2.1 Mobile P2P

A peer-to-peer system architecture should be symmetrically distributed and should implement unconstrained volunteer participants. Three typical resources can be shared via a P2P system: computing power, storage capacity, and user presence. However, these three resources are the precise weaknesses of most mobile devices since current mobile devices such as PDAs, wireless sensors, or even mobile phones cannot provide powerful computing ability or mass storage capacity. Due to the mobility, arbitrariness, and instability of mobile wireless network environments, each peer in the network cannot provide a relatively stable connectivity presence to others. The main motivation for this work is driven by the need to provide a feasible architecture applied on top of the mobile ad hoc wireless networks to match all the following requirements simultaneously, from which the

popularity of mobile P2P applications benefits:

- **Fault tolerance:** The absence of a single peer, because of failure or voluntary disconnection, cannot affect the operation of the system. In other words, the P2P system has the ability to operate in dynamic network environments;

- **Load balance:** Due to the distributed storage of resources, network links towards resources can be considered as complete graphs, and the traffic load is consequently more homogeneously distributed [23];

- **The ability to harness idle storage and network resources:** Idle storage and resource can be estimated to be available during certain periods of time.

4.2.2 The Challenges of the Mobile P2P System

As the range of mobile applications increases, the design of mobile P2P architecture faces a number of challenges:

- **Choice between centralized architecture and decentralized architecture** Centralized architecture largely defeats the purpose of the peer-to-peer system and can be too restrictive if a centralized index server is not available [38]. Since all peers need to connect to a centralized server, centralized architecture provides an efficient mechanism to manage members. In decentralized architecture, there is no centralized server. Some peers can self-organize as one group and communication can generally take place between groups. Therefore, even very large systems are feasible. However, decentralized architecture requires a great deal of overhead to maintain relationship with neighbors.

- **Resource lookup** Centralized architecture again defeats the purpose of the peer-to-peer system when using a centralized index server to locate resources. But decentralized architecture needs to use multi-cast query to reach all of its neighbors. This results in a large amount of overhead.

- **Resource dispensation and location** If every resource is stored only in the local peer, namely the owner, it does not function in accordance with the characteristics of P2P: The ability to harness idle storage and network.

4.3 Description of MAHP2P

To overcome the challenges of the mobile P2P mentioned above, we explore integrating distributed hybrid architecture into our proposed peer-to-peer system. This system, denoted as MAHP2P, simplifies the design of peer-to-peer by addressing the following characteristics:

4.3.1 Virtual System Community (VSC)

The entire P2P system is composed of several virtual subsystems. We call these virtual subsystems virtual system communities (VSCs). Each VSC is an open system which every peer can join or leave freely. One peer acts as super-peer to maintain this VSC. Members within the same VSC can communicate with each other directly. If one peer wants to communicate with peers in another VSC, the peer needs authorization from two super-peers (one super peer from the sender's VSC and another from the destination peer's VSC). These two authorizations form a security mechanism. The subsystem is a concept from a virtual view instead of a physical location concept. So it is possible for VSC to overlap with others in the physical view. Fig.4.1 demonstrates the virtual view, and Fig.4.2 illustrates the physical view. We define that to any one peer of VSC A, if one peer of VSC B is reachable, all peers in VSC B is reachable.

4.3.2 Hierarchy

The first goal of MAHP2P is to establish a general P2P platform with super adaptability to complex network environments. MAHP2P is a combination of two traditional architectures: centralized mode and decentralized mode, which is based on the three-layer hierarchy: infrastructure-based index server, super peer, and peers. These architectures enable MAHP2P to have the ability of server-awareness. When peers perceive that there is an infrastructure-based index server existing nearby, peers will transit their mode to centralized mode. We know that infrastructure-based index servers are usually professional servers with very high computing power, sufficient storage capacity, and wide

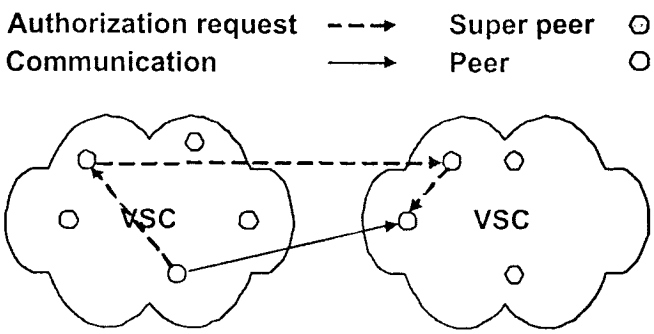

Figure 4.1: Virtual System Community

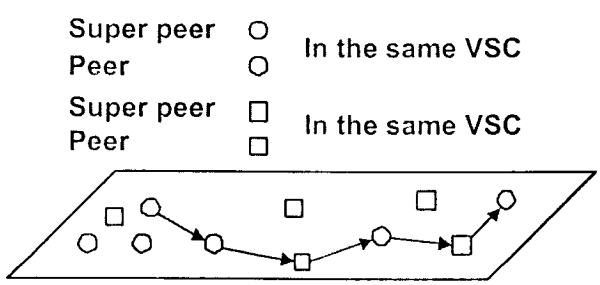

Figure 4.2: Physical View of P2P System

bandwidth. Hence, in this mode, peers can benefit from the stable centralized directory of object location and management mechanism provided by the index server. Moreover, this centralized system can reduce the probability of failure resulting from the arbitrariness of peers' mobility and instability of mobile ad hoc wireless network environments.

Although this server-centric architecture is poorly scalable for large-scale network environments, the collapse and load problems can be solved completely if the index server is available only to neighboring peers. If the index server fails or is not present, peers will transfer to semi-centralized mode. Within a semi-centralized mode, an index server also exists and is acted on by a peer. We call this super-peer. Usually, this peer is the common peer with sufficient battery power and relative stability. Therefore, this super-peer is not fixed. The main functions, such as index database or directory of the super-peer, are dynamically migrated among these peers. The super-peer also provides a mechanism to maintain and manage its members. The hierarchy of MAHP2P system is shown in Fig.4.3.

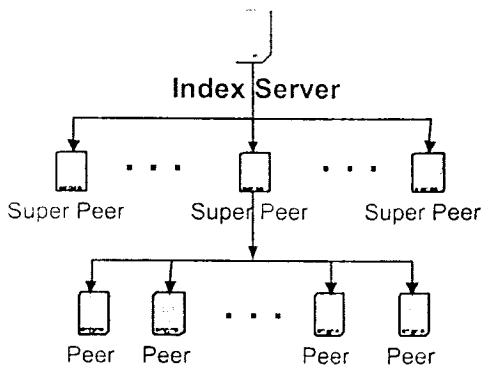

Figure 4.3: The Hierarchy of the MAHP2P System

4.3.3 Distributed Deployment of the Encoding Segment

Resource sharing and exchange is a very important function provided by the P2P system. In a pure peer system, all resources are stored in the respective owners. Therefore, only one copy of a resource exists in the system. This affects the reliability and efficiency of resource deployment. In the mobile P2P system, peers are inherently resource-constrained. Some mobile end devices, such as laptops, PDAs, and even mobile phones, are restricted

with respect to bandwidth, storage capacity, and processing power [38]. Moreover, the dynamic nature of the mobile P2P system makes it possible for peers to go off-line at any time due to system failure or voluntary disconnection. To avoid having to perform another search to locate an alternative peer, a peer can cache the original resource in the intermediate peer along the path to the destination peer. However, this scheme results in a large number of abundant duplicates dispersed throughout the system, and requires a great deal of workload in updating or maintaining these duplicates. This is also the main origin of network overhead.

MAHP2P uses another scheme which adapts a distributed deployment of encoding segments to overcome these challenges. As the owner of a resource, a peer first encodes the resource into code words using data abundant algorithm (e.g. erasure code). Through the first resource delivery, the code words are deployed to respective intermediate peers which possess specific properties, such as battery life or stability etc. These code words are unique in the system. This prevents duplicate delivery. When another resource query arrives, the code words which are distributed to certain peers are delivered to the destination peer asynchronously. According to the data abundant technique, the original resource can be reconstructed by receiving a specific length subset from code words. In this scheme, even if some peers fail or leave, the destination peer can still obtain the entire resource if certain specific length subsets are received. Therefore, using distributed deployment of the encoding segment, MAHP2P provides more efficient fault tolerance and load balance as well as the ability to harness idle storage.

4.3.4 Failure Recovery of the Super Peer

As a central point, the super peer plays a key role in the virtual system community. It manages all of the activities of peers. However, the super peer also acts as a common mobile peer with relative super power, and it still possesses all of the characteristics of a mobile peer: arbitrary dynamic mobility, voluntary leaving, or failure. Therefore, recovery of the super peer must be considered when we design such a hierarchy-based P2P system. Moreover, recovering the super-peer is the recovery of the virtual system community.

We employ the distributed hash table technique as a recovery mechanism. We partition the dynamic hash table to fixed length parts and deploy parts to certain peer

candidates. In the case of the super-peer failing permanently, we can recover the super peer from these candidate peers.

4.3.5 Publishing and Subscribing the Peer State

In MAHP2P, the publishing and subscribing of the peer state is a synchronization mechanism among peers. Each peer has a state indicator, which is an unique incremental sequence number. After each event occurs the state will be increased by one. In addition, each peer has a log file. Each activity or event will be stored in this log file. Thus we can review or recover all the events stored in the log file. This state indicator is advertised throughout the virtual system community. Moreover, each peer has the state record of other peers. This record shows that this peer knows the state which the corresponding peer advertised due to the occurrence of a certain event. If a peer, denoted $P_{i,1}$, can find that the state record of certain peers, denoted $P_{i,2}$, is smaller than that of the $P_{i,2}$ advertised. This peer know that some events are occurring in $P_{i,2}$ which are published by $P_{i,2}$. In order to maintain system consistency and synchronization, each peer needs to advertise this state by sending a subscription message. After receiving subscriptions, $P_{i,2}$ begins to push these event messages to the subscribers with the latest state indicator. After receiving these event messages, each subscriber makes relevant updates and also increase the state record of $P_{i,1}$ with this latest state indicator. Through this mechanism, changes to state of $P_{i,1}$ are deployed to the entire virtual system community.

4.3.6 Mobile Agents

A software agent is a computing entity that performs certain tasks on behalf of somebody or something [20]. A mobile agent is a software agent that has the additional property of not being bound to operate only in the system in which it began [24]. A mobile agent has the unique property that during its lifetime it can be halted, with its state and code moved to another computer on the same network, and then can continue to execute from where it stopped on the previous computer [24]. Since MAHP2P uses a distributed resource deployment mechanism to establish communication connections among mobile peers, and then performs multiple interactions over these connections, each of these interactions generates network traffic. When this kind of interaction happens frequently, the effect of traffic on performance is salient. The mobile agent provides a suitable solution to these problems. Certain significant characteristics of the mobile agent have been

provided for analysis.

- **Reducing the need for bandwidth** As a computing entity performing certain tasks, the mobile agent allows all implementations to take place locally. Usually the mobile agent encapsulates all required data. Wherever a mobile agent reaches, it has all its required data, and therefore does not need to communicate with other peers. Therefore, the mobile agent reduces network traffic by moving processing to raw data instead of moving raw data to processing [24].

- **Asynchronous Operation** When dispatching a mobile agent, the peer does not need to wait for the mobile agent to return. This means that the original peer does not have to maintain a connection waiting for the mobile agent to return. After finishing the task, the mobile agent checks the original peer to see whether it is available. The mobile agent can continue probing until the original peer returns to the system. This makes peers more flexible.

- **An intelligent entity** Very often a mobile agent brings new data with it when mobile agents move among peers. Each peer can benefit from accepting a visiting mobile agent [24]. Moreover, all mobile agents can learn from local peers. By analyzing the history of visited peers, the mobile agent can intelligently determine the next action or destination. A mobile agent can also be easily duplicated and dispatched in different directions [24]. It provides a mechanism for parallel computing.

We employ mobile agents for lookup mechanisms which are in charge of collecting data required by the original peer and returning results to this peer. We call this mobile agent *AgLookup*. If the querist wishes to request a certain resource, it creates a new *AgLookup* with a unique ID and sends this *AgLookup* to the super peer or index server if it is available. After collecting all respective locations of resources, *AgLookup* duplicates itself in the amount equal to the number of locations. These duplicates of *AgLookup* then move to the respective locations. In each location, *AgLookup* begins to collect code words of resources, and moves back to the original peer; that is, the querist. On the way to the querist, if *AgLookup* finds that a certain peer is suitable for hosting code words and there is more than one code words that *AgLookup* brings, *AgLookup* sends a copy of

Table 4.1: Notation

Term	Meaning
$ID(P_{i,n})$	$VSCID(i) + GUID(P_{i,n})$ Unique identifier of a peer P in global MAHP2P system
$L(P)$	An index to represent the performance level of peer P
$AgDir$	The mobile agent for the main directory, which resides in the super peer
I	Index server

one code word to this peer, and after sending a notice to the super-peer, agent continues its trip.

4.4 The Analysis of the Model of MAHP2P and the Algorithm of MAHP2P

In the previous sections, we described the characteristics of MAHP2P. In this section, we prove the feasibility of this protocol through analyzing the system model and algorithm. We summarize the notations in Table 4.1, which we use in this chapter.

4.4.1 Initialization of the System

Given a virtual system community $VSC(i)$, each $VSC(i)$ has a unique ID, $VSCID(i)$, which is produced at random from a large identifier space by the first peer in this virtual system community. In actuality this first peer is the creator of the virtual system community. Each peer is also assigned globally unique identifiers $GUID(P_{i,n})$. MAHP2P uses these two unique identifiers as a name space. The peer can be identified globally uniformly by $VSCID(i) + GUID(P_{i,n})$, denoted $ID(P_{i,n})$. Every peer has a state parameter which represents how fresh this peer is. This parameter is the state change indicator of a peer implemented by a unique incremental sequence number. For example, if a peer's member list changes, the state will be increased by one. Each peer has a performance index to represent the peer's level of the battery life, storage capacity and stability. These parameters are combined by adding weighted values. Stability refers to whether the peer disconnected due to failure or involuntary departure. We denote this performance index of peer P as $L(P)$. We define the three levels of $L(P)$: *Super*,

Median, and *Peer*. *Super* is the highest level that peers can obtain as candidates for the position of super peer; Peers with *Median* can be looked at as candidates for the position of code words host; and *peer − level* peers are the common peers, which are peers at the lowest level. Thus, in the initialization phase, the first peer P creates a new virtual system community $VSC(1)$ with $VSCID(1)$. The ID of this peer P in $VSC(1)$ is $ID(P_{i,n}) = VSCID(i) + GUID(P_{i,n})$. Each refresh peer P has an index $L(P)$ equal to *Super*. Hence, peer P is transferred to the super peer, represented by S. Therefore, in $VSCID(1)$ there is only one peer, a super peer, S. The first step for the super peer, S, is trying to connect to the index server. If an index server is available, S registers itself in index server I. S also initializes the main directory using the dynamic hash table. The structure of the main directory is shown in Fig.4.4.

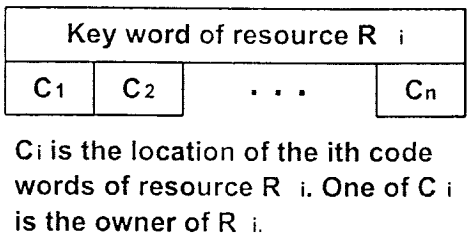

Figure 4.4: Structure of the Main Directory

4.4.2 Peer Joining

In the mobile P2P system, peers can join (and leave) at any time. The main challenge in implementing these operations is preserving the stability of the system. In the following section, we describe the overall joining algorithm. We would like the results of the joining to be the same as those which the new peer would have had had it already been a member of the current system. This means that correlative notices about joining are reduced to the smallest number possible. There are two options for a peer P that want to join the system. This peer can choose to create a new VSC, or else it can join an existing VSC. Creating a new VSC is discussed in the section entitled **Initialization of the System**. Therefore, we analyze how to join an existing VSC. We define that a peer can only join a VSCs to which neighboring peers belong. In other words, a new peer can only join the VSC that is the same as that of neighboring peers.

The detailed joining processes are described as follows. We assume that the new peer is denoted P_n and the existing virtual system communities $VSC(i)$ with super peer S_i and peers $P_{i,1}$, $P_{i,2}$, $P_{i,3}$.

1. First, peer P_n knows the accessible VSC from its neighboring peers, and selects one of the peers to join (e.g. one of its neighboring peers, $P_{i,1}$, which VSC is $VSC(i)$ and super peer is S_i).

2. P_n configures itself with VSC and corresponds super peer S_i, and sends a joining notice to $P_{i,1}$. When receiving this joining notice, $P_{i,1}$ updates the member list and increases its state indicator by one. This state indicator is advertised periodically to others.

3. Each peer has a record of all members' state indicators. One peer checks the state table periodically to see if there is a peer with a newer state indicator. If one exists, the peer (for example $P_{i,2}$) sends a subscription request to $P_{i,1}$ with the old state indicator.

4. After receiving the subscription request from $P_{i,2}$, $P_{i,1}$ sends all updated content according to the difference between the current state indicator and the old state indicator received from $P_{i,2}$. Using the updated subscription mechanism, MAHP2P provides the feature to synchronize peers. For example, through this mechanism new peer P_n can be updated with a complete member list from $P_{i,1}$. Super peer S_i, as well as index server I, also anticipate this updating process.

4.4.3 Distributed Deployment of Resource

The dynamic nature of peer systems produces great challenges for reliability and fault tolerance in implementation. A peer that is distributing resources may go off-line at any time. In the MAHP2P system, we consider both reliability in terms of data delivery and fault recovery in terms of failure serve peers. To ensure reliable distribution of resources, we employ distributed deployment of encoding segments for reliability and delivery efficiency. In addition, this distributed deployment of encoding segments can benefit from

load balance and fault tolerance. MAHP2P uses erasure code, an abundant data encoding technique, to implement this scheme. The main idea of erasure code is that m data packets are encoded into n fragments with an erasure code algorithm, and these m original packets can be reconstructed by employing any m out of n code words ($n > m$).

The detailed distribution processes are described as follows. We also assume that there is a peer $P_{i,j}$ in $VSC(i)$ with resource R_j that needs to be distributed. Fig.4.5. illustrates the deployment process.

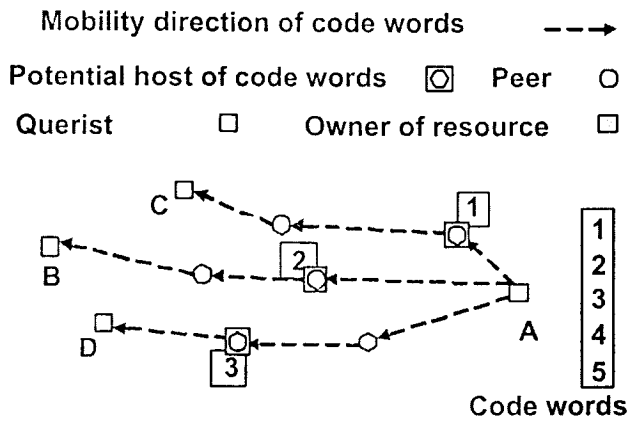

Figure 4.5: Peer Resource Deployment

1. First, peer $P_{i,j}$ encodes resource R_j into n fragments or code words, r_k, $k = 1...n$. $P_{i,j}$ then sends the group of key-address pairs to index server or super peer S_i. The fraction of dynamic hash table is shown in Fig.4.6.

2. When resource R_j is requested by peer $P_{i,j}$, and after the querist receives the locations of resources from the index server I or super peer S_i, the querist will inform the current sole host, namely owner $P_{i,j}$. After receiving this notice, $P_{i,j}$ will begin to deliver these encoded code words to the querist.

3. On the way to the querist, if the packet of code words finds that the performance of the intermediate peer it passes is suitable to host or that level $L(P)$ is equal to

Median, it leaves a duplicate of the code word to this intermediate peer.

4. When intermediate peer, $P_{i,g}$, receives a duplicate of a code word, it sends a notice to the index server or super peer S_i. The index server or super peer then updates its dynamic hash table. The updated dynamic hash table is now shown in Fig.4.7.

5. According to the erasure code technique, as long as the destination peer, the querist, receives only certain of specific length subsets of code words, it can recover the original resource. Hence, through distributed deployment of encoding segments, MAHP2P offers reliable delivery and fault tolerance.

6. When another querist requests the same resource, and after the querist receives the locations of the resource from the index server or super peer, the querist will inform two hosts: $P_{i,j}$ and $P_{i,g}$. $P_{i,g}$ then sends its code words to the destination peer, and $P_{i,j}$ sends the remainder. Therefore, code words are distributed at two peers now. Each intermediate peer can host only one code word from one resource. This ensures that there are no more than two code words from the resource in the same intermediate peer. This rule is not suitable to the owner of the resource.

7. The owner, $P_{i,j}$, continues deploying its code words when delivering the remainder. Therefore, the more often a resource is requested, the more likely the resource is distributed in one VSC, and the more efficient the load balance will be. This matches the scheme of popular resources deployment.

8. In case, one of these intermediate peers leaves or fails, the delivery task of the respective code words is replaced by the owner of the resource. If the owner leaves volunteer, the index server, the super peer, and the respective intermediate peers delete all of the resources of this peer.

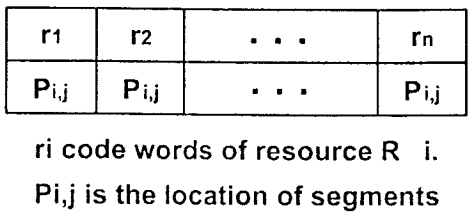

Figure 4.6: Fraction of DHT in Super Peer

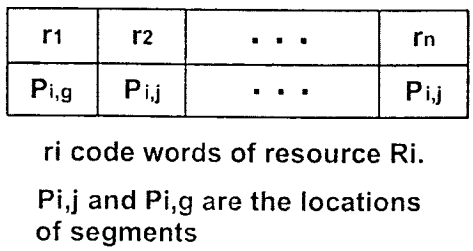

Figure 4.7: Updated Fraction of DHT in Super Peer

4.4.4 Peer Lookup

In the MAHP2P system, the mobile agent mechanism is employed to implement peer lookup.

1. When one peer requests a certain resource, it first creates a mobile agent *AgLookup* with a unique ID. This querist then determines to check whether the index server is available or not. If the index server is available, *AgLookup* moves itself to the index server. If not, it moves itself to the super peer. If both of them are not present, this query is terminated.

2. After receiving *AgLookup*, the index server or super peer checks the local dynamic hash table. If no item satisfies the query condition, the index server or super peer returns mobile agent *AgLookup* to the original querist. The dynamic hash table stores the code words of resources and their corresponding locations. Therefore, if the index server or super peer finds a specific resource, it duplicates the mobile agent *AgLookup* in an amount equal to the number of different locations. These duplicate of *AgLookup* have the same unique ID of the original querist, but have

different series numbers. Next, the index server or super peer sends all mobile agents to corresponding locations.

3. When a duplicate of mobile agent *AgLookup* reaches the host of code words, it collects the corresponding code words. *AgLookup* then checks to see if the original querist is present. If the querist is online, *AgLookup* returns to the host. If not, *AgLookup* continues checking until the querist becomes available. Each *AgLookup* is assigned an expiry time when it is created by the original querist. If *AgLookup* expires, it deletes itself.

4. The querist collects mobile agent *AgLookup* and its duplicates. After an interval time equal to the expiry time of *AgLookup*, the querist checks to see if the number of code words it collected is sufficient for it to recover the original resource based on the erasure code algorithm. If an insufficient number of code words have been received, the query process fails. Fig.4.8. illustrates the lookup process.

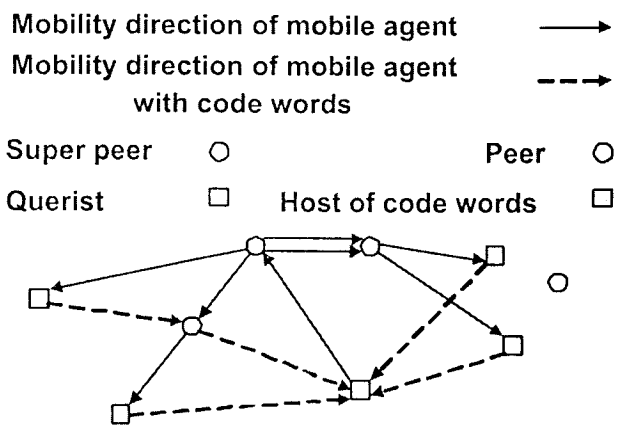

Figure 4.8: Peer Lookup

4.4.5 Peer Leaving

When peer $P_{i,j}$ volunteers to leave the system, it removes itself in a way that sends a notice to one of the members. After receiving this notice, member $P_{i,g}$ updates its state

and advertises it to other members. Similar to **Peer Joining**, other members can obtain this updated state from $P_{i,g}$. If peer $P_{i,j}$ distributes certain resources, the index server or super peer updates its dynamic hash table when it receives this updated state from peer $P_{i,j}$. All of the host peers of the code words of $P_{i,j}$ can also be informed of the updated state and delete these code words respectively. If a peer disappears without notice, and when this peer returns, it contacts one of its nearest members in order to obtain the latest updates

4.4.6 Peer Failure

When a peer notices that certain peers fail, it does everything it can to fix its own state. For example, the host of code words stops moving the mobile agent back to original querist. However, different peers are in charge of handling different types of defunct peers. Four types of peers must be considered separately when these peers fail: the super peer, peers with some partition of the dynamic hash table, peers with code words for resources, and other common peers.

Super Peer Failure

This failure is found and handled by peers with the partition of the dynamic hash table. If the super peer fails, all peers with the partition of the dynamic hash table become candidates for the position of super peer. The youngest peer will be chosen as the new super peer. Other candidates then move the partition of the dynamic hash table to the new super peer in order to recover the hash table. After that, the new super peer changes its state and advertises this change to all members. The next job is to find another peer to replace its former position. Finally, if the index server is present, the index server and the new super peer begin to synchronize their hash tables with one another.

Peers with the Partition of Hash Table Failure

This failure is found and handled by the super peer. If these peers fail, the super peer finds another peer to replace the failed peer.

Peers with Code Words of Resource Failure

This failure is found and handled by the owner of the resource since every owner of resource has a duty to maintain all the code words of the resource that it distributes. If a owner finds that a peer with code word has failed, the owner continues to distribute these code words. In addition, the owner informs the super peer to update the items of these code words. It makes the super peer replace the location with the owner's address.

Other Common Peer Failure

This failure is found and handled by the super peer. If the super peer finds that a common peer fails, it deletes the corresponding items in its hash table. It then advertises this undated state to all of the members. All members receiving this updated state delete this defunct peer from the member list or delete the relevant code words of the resource.

4.4.7 Synchronization Between the Super Peer and the Index Server

Synchronization between the super peer and the index server employs the advertising approach of the updated state. When the index server returns after an absence, the index server compares its state of the super peer with the current state of the super peer. If the state is smaller than the current state of the super peer, the index server sends a subscription to super peer with the old state and requires for an update. If the super peer finds that the index server has a newer version of the hash table, the super peer implements the counter-operation.

4.4.8 Super Peer Migrating

Super peers always act by mobile devices, with perfect performance. These performances are evaluated by battery power, storage capacity, and the stability of the user. However, every device ages during the using process. For example, battery power will weaken, storage capacity will decrease, and users of this device will become more unsteady. So when the performance of the super peer is reduced, we need to consider letting another peer with high performance play the role of super peer before the older one breaks down. In order to reduce network traffic, MAHP2P selects one peer from these peers with the partitions of the hash table. These candidates are used to recover the super peer

when it fails (see section **Failure Recovery of the Super Peer**, and for the detailed implementation of this program, see section **Super Peer Failure**).

4.4.9 Communications Among VSCs

If two peers from different VSCs wish to communicate, they need the authorization of both super peers. For instance, we have two VSCs: $VSC(1)$ and $VSC(2)$, respectively with the super peer, S_1 and S_2, as well as peers $P_{1,1}$ and $P_{2,1}$. If two peers $P_{1,1}$ and $P_{2,1}$ want to establish a communication channel, they first need to register at the respective super peer. Two peers also need to know the ID of the each other. The peers' ID is presented as $ID(P_{i,n}) = VSCID(i) + GUID(P_{i,n})$, and the real address is hidden behind the ID. Each super peer has a mapping that relates to the real address of the peer's ID of its members. Another requirement is that two VSCs must be reachable to each other. Through local super peer S_1, peer $P_{1,1}$ sends a request to the super peer of $VSC(2)$, S_2. S_2 checks the register list, then creates a token with the real address of $P_{2,1}$ and an expiry time, denoted $T_{2,1}$. S_2 returns token T back to $P_{1,1}$ through super peer S_1. After receiving the token, $P_{1,1}$ can begin to deliver data to $P_{2,1}$.

Data uses the token as a pass to move information to the destination. When travelling within the $VSC(2)$, data must use the token in order to pass every intermediate peer and be accepted by destination peer $P_{2,1}$ successfully. The key question is how to handle intermediate peers that come from the third VSC. In order to avoid malicious attack, delivery will be stopped unless $P_{1,1}$ also has a token from this VSC.

4.5 Related Work

In this section, we discuss current approaches to solving the challenges facing the P2P system.

4.5.1 The Centralized P2P System

As we know, the new wave of P2P architecture occurred during the late 90s. This system focused on the free file sharing system, which can be considered the first generation of the P2P system. This first generation functioned more as a combination of client-server and peer-to-peer mode [25] than a P2P system. The typical first generation application

of the program is Napster, which consists of peers and a dedicated server that maintains an index of the connected peers and their resources. Fig.4.9 illustrates the architecture of Napster. In the Napster system, all peers need to maintain their connection to the index server. When peer A wants to query for resources, a query message is delivered to the index server (1). The server then co-operates to find resources from its index database, and returns the address of resources to A (2). After receiving the address of resources, A sends a request for resources to the owner of resource B and obtains the resource from B(4).

In such centralized architecture, peers have to pay for a large amount of maintenance. Moreover, Napster has only one single access. The system can completely break down if this index server fails. In addition, the centralized system architecture has poor scalability. The centralized server suffers heavy load from all peers.

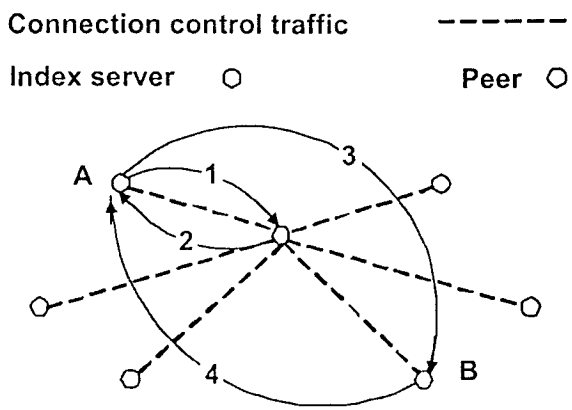

Figure 4.9: The Architecture of Napster

4.5.2 The Decentralized P2P System

The second generation P2P system can be described as a pure P2P system since every peer in this P2P architecture has equal functions and no dedicated servers are needed [25]. The peer discovery and query delivery functions are distributed to all peers [25]. There are many successful examples of this system, including Gnutella [1], Chord [27] [26], and Tapestry [30] [16]. We employ Gnutella as an example of a decentralized P2P system in order to analyze its characteristics.

In the Gnutella system, each peer first searches for neighboring peers by broadcasting a routing message. When a peer receives this message, it re-broadcasts the message to its neighboring peers. This broadcast continues repeatedly until certain constraints are satisfied. Gnutella uses Time-To-Live (TTL) to restrict this propagation. Through this pattern, Gnutella can obtain the locations of peers. Fig.4.10 illustrates the architecture of Gnutella. If peer A sends a query to all of its neighboring peers, and if the queried resource is not found, neighboring peer B re-broadcasts this query to its neighboring peers C, D, E, F, G. In this case, E has the resource that peer A wants. Therefore, peer E delivers the resource to peer A.

Gnutella is a decentralized pure P2P system that overcomes certain problems of the

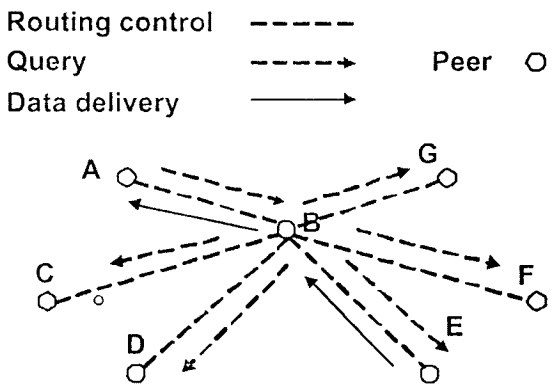

Figure 4.10: The Architecture of Gnutella

Napster system. The main advantages of Gnutella are its de-centrality, and thus scalability and fault tolerance. But Gnutella is based on the propagating broadcast. It will produce a large amount of network traffic to maintain peer discovery. And it is also obviously heavy to use in a mobile environment in which mobile network peers move dynamically.

Chapter 5

Conclusion

The goal of this thesis is to describe a mobile application architecture that is designed for museum visitors, especially group visitors, and instructor-student groups in particular. This application is built on a wireless network environment and is mounted on a mobile device with wireless connectivity. Museum visitors use the mobile application to guide their visit. They can also utilize these mobile applications to form a virtual community in which they can communicate with each other. The application architecture allow visitors to take advantage of many benefits, such as obtaining the batch of information from the network or using the flexible functionality provided by the mobile application.

5.1 Achievements of This Work

The work described in this thesis presents and discusses the following two contributions:

- A reliable transmission mechanism based on the ad hoc wireless network. This unicast-based self-adaptive reliable transmission protocol is a combination of ARQ and FEC that provides low overhead, high delivery rate, and low re-transmission. Low overhead is achieved by using accumulative acknowledgement. The rate of data delivery is increased by using the self-adaptive window technique. The use of the probabilistic erasure code technique provides an approach to reduce re-transmission while having only a small amount of redundant data. Important metrics, delivery rate, and delivery delay were evaluated and compared with TCP and UDP and show that this new transmission mechanism has a higher delivery rate and medium delivery delay. These qualities make it a good candidate for transmission protocol for use in mobile ad hoc wireless networks.

- A mobile-agent-based self-adaptive hybrid peer-to-peer system is a hierarchy with three layers. Peers can benefit from the high performance of the index server when it is available. However, these peers can work on super-peer mode self-adaptively. This transition is seamless. In addition, the system employs a distributed erasure code storing mechanism for fault tolerance and load balance. These significant characteristics make this system more suitable for mobile ad hoc wireless network environment.

5.2 Future Work

The primary users of this application are museum visitors. Therefore, visitors satisfaction is the only standard for evaluating the success of an application. In addition, the main layer of the system that visitors interact with is the interface, so in future work we will focus on designing user interfaces. Since the wireless mobile device is portable and small and consequently has a very small screen, we need to take considerations other than desktop application into account. Generally speaking, people should be allowed to use this application conveniently. Moreover, our future work should extend to analyzing the multi-mode user interface such as the voice. Use of voice recognition will help older visitors interact with mobile devices easily and conveniently. The second focus is the data format used in our application. We will use XML standard to format the data that transfers to the mobile nodes. Hence, we can exchange data between different platforms.

Bibliography

[1] The gnutella protocol specification v0.41.

[2] Rfc793-transmission control protocol. *http://www.faqs.org/rfcs/rfc793.html*, 09 1981.

[3] Napster protocol. *http://opennap.sourceforge.net/napster.txt*, April 2000.

[4] The use of forward error correction (fec) in reliable multicast. *http://rfc3453.x42.com*, 12 2002.

[5] Forward error correction. *http://www.cs.tut.fi/tlt/stuff/ads/node12.html*, 05 2005.

[6] Go-back-n error recovery. *http://www.erg.abdn.ac.uk/users/gorry/course/dl-pages/gbn.html*, 05 2005.

[7] Polling. *http://www.erg.abdn.ac.uk/users/gorry/course/arq-pages/poll.html*, 05 2005.

[8] Reliable delivery. *http://www.erg.abdn.ac.uk/users/gorry/course/arq-pages/reliability.html*, 05 2005.

[9] Reliable transmission. *http://www.cs.arizona.edu/llp/book/node30.html*, 05 2005.

[10] Selective repeat error recovery. *http://www.erg.abdn.ac.uk/users/gorry/course/arq-pages/sr.html*, 05 2005.

[11] Short for round-trip time. *http://www.webopedia.com/TERM/R/RTT.html*, 05 2005.

[12] Stop and wait arq. *http://www.erg.abdn.ac.uk/users/gorry/course/arq-pages/saw.html*, 05 2005.

[13] Timeouts. *http://www.totaltele.com/acronyms/Default.asp?Letter=T*, 05 2005.

[14] Traffic congestion. *http://www.cs.bris.ac.uk/Teaching/Resources/COMS11301/jargon.html*, 05 2005.

[15] Ucb/lbnl/vint network simulator. *http://www-mash.cs.berkeley.edu/ns/*, 05 2005.

[16] Jeremy Stribling Sean C. Rhea Anthony D. Joseph Ben Y. Zhao, Ling Huang and John D. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal*, 22, January 2004.

[17] Reza B'Far. *Mobile Computing Principles.* Cambridge Press, 2005.

[18] Azzedine Boukerche and Sajal K. Das. Congestion control performance of r-dsdv protocol in multi-hop wireless ad hoc networks. *Wireless Networks*, 9:261–270, 05 2003.

[19] Shirky C. What is p2p and what isn't. *O'Reilly Network*, November 2000.

[20] A. Caglayan and C. Harrison. Agent sourcebook. *Wiley Computer Publishing*, 1997.

[21] Flick Software Company. Handheld business solutions. *www.flicksoftware.com*, 2005.

[22] Marco Conti. *Body, Personal and Local Ad Hoc Wireless Networks.* CRC Press, The Handbook of Ad Hoc Wirless Network, 2003.

[23] Mattia Monga Davide Balzarotti, Carlo Ghezzi. Supporting configuration management for virtual workgroups in a peer-to-peer setting. *SEKE'02*, July 2002.

[24] Cameron Ross Dunne. Using mobile agents for network resource discovery in peer-to-peer networks. *ACM SIGECOM Exchanges*, 2, June 2001.

[25] Jussi Ala-Kurikka Jukka Riekki Jaakko Sauvola Erkki Harjula, Mika Ylianttila. Plug-and-play application platform: Towards mobile peer-to-peer. *MUM 2004*, October 2004.

[26] David Karger Robert Morris Ion Stoica Frank Dabek, M. Frans Kaashoek. Wide-area cooperative storage with cfs. *SOSP'01*, October 2001.

[27] David Karger M. Frans Kaashoek Hari Balakrishnan Ion Stoica, Robert Morris. Chord: A scalable peer-to-peer lookup service for internet application. *SIGCOMM'01*, August 2001.

[28] David Karger M. Frans Kaashoek Hari Balakrishnan Ion Stoica, Robert Morris. Chord: A scalable peertopeer lookup service for internet applications. *SIGCOMM01*, August 2001.

[29] Jingyi He Soung-Chang Liew Jiancong Chen, S.-H. Gary Chan. Mixed-mode wlan: The integration of ad hoc mode with wireless lan infrastructure. *GLOBECOM03*, 2003.

[30] Satish Rao Kirsten Hildrum, John D. Kubiatowicz and Ben Y. Zhao. Distributed object location in a dynamic network. *SPAA'02*, August 2002.

[31] David Lundberg. Ad hoc protocol evaluation and experiences of real world ad hoc networking.

[32] Joe Macker. Reliable multicast transport and integrated erasure-based forward error correction. *IEEE*, MILCOM 97, 10 1997.

[33] David R. McDonald. Elements of applied probability. page 142, 1994.

[34] T. Mundt N. Maibaum. Jxta: A technology facilitating mobile peer-to-peer networks. *Proceedings of International Mobility and Wireless Access Workshop (MobiWac 2002)*, October 2002.

[35] V. Pless. Introduction to error-correcting codes. *Wiley Publish*, 2nd Edition, 1989.

[36] Michael Finkenzeller Rdiger Schollmeier, Ingo Gruber. Routing in mobile ad-hoc and peer-to-peer networks a comparison. *Publisher Springer-Verlag*, pages 172 – 186, 2002.

[37] Luigi Rizzo. Effective erasure codes for reliable computer communication protocol. *ACM SIGCOMM Computer Communication Review*, 27(2):24–36, 4 1997.

[38] Kevin C. Almeroth Sami Rollins. Pixie: A jukebox architecture to support efficient peer content exchange. *Multimedia'02*, December 2002.

[39] Amitabh Mishra Satyabrata Chakrabarti. *Quality of Service in Mobile Ad Hoc Networks*. CRC Press. The Handbook of Ad Hoc Wirless Network, 2003.

[40] RodrigoFonseca Sukun Kim and David Culler. Reliable transfer on wireless sensor networks. *The first IEEE International Conference on Sensor and Ad hoc Communication and Networks*, 10 2004.

[41] Hogg Robert V. Probability and statistical inference. page 294, 2001.

[42] Hogg Robert V. Probability and statistical inference. page 153, 2001.

[43] J.H. van Lint. Introduction to coding theory. *Springer-Verlag*, 2nd Edition, 1992.

[44] Andy Ogden Yoshiki Ohshima, John Maloney. The parks pda: A handheld device for theme park guests in squeak. *OOPSLA'03*, October 2003.