


國立台灣大學資訊工程學研究所
博士論文

無線網路環境下混合式對等網路架構
及管理機制之研究

Hybrid Peer-to-Peer Architecture and
Management for Wired and Wireless Networks



研究生：王正豪
指導教授：李肇林 博士

中華民國九十一年六月

誌謝

進入台大十二年的時光，如今終於能順利取得學位，心中真是百感交集。要感謝的人實在太多，無法一一誌謝。

首先，我要感謝恩師李肇林博士多年來的悉心指導，使我在研究、生活及做人處事上獲益良多。感謝劉長遠教授，湯耀中教授，項潔教授，李嘉晃教授，陳信希教授以及傅楸善教授對論文精闢的指教與匡正。同時也要感謝 OA 網路實驗室歷屆學長學弟們多年來的相互討論切磋，使我在各方面都獲得許多寶貴的學習經驗。

感謝父母親的從小到大養育栽培，才有今天的我。同時也感謝兄姊不時的關心。

最後，特別要感謝乃文在精神及生活上給予最大的支持與照顧，使我無後顧之憂，順利完成學業。尤其每當遇到困難時，乃文就成了我堅持下去的毅力與勇氣。

謹以此論獻給我的父母，兄姊，以及乃文。

Table of Contents

ABSTRACT IN CHINESE	1
ABSTRACT	2
CHAPTER 1 INTRODUCTION.....	3
CHAPTER 2 BACKGROUND INFORMATION AND RELATED WORKS	6
2.1 TAXONOMY OF COMPUTER SYSTEMS	6
2.1.1 <i>Centralized vs. Distributed</i>	6
2.1.2 <i>Client-Server vs. Peer-to-Peer</i>	6
2.1.3 <i>Pure vs. Hybrid Peer-to-Peer</i>	7
2.2 EVOLUTION OF NETWORKING TECHNOLOGIES	7
2.2.1 <i>LAN Technologies</i>	7
2.2.2 <i>Wireless LAN</i>	8
CHAPTER 3 PROPOSED HYBRID PEER-TO-PEER ARCHITECTURE.....	9
3.1 INTRODUCTION.....	9
3.2 LOCATION SERVICE AND PROFILE MANAGEMENT	10
3.3 MANAGEMENT RECORDS	11
3.4 BASIC OPERATIONS	12
3.5 APPLICATIONS	13
3.6 SECURITY CONSIDERATIONS	13
3.7 DESIGN AND IMPLEMENTATION ISSUES.....	14
3.7.1 <i>Location Server Discovery</i>	14
3.7.2 <i>Recursive vs. Iterative Queries</i>	14
3.7.3 <i>Implicit vs. Explicit LS Queries</i>	15
3.7.4 <i>Inter-LS Communication</i>	15
CHAPTER 4 MANAGEMENT OF THE ARCHITECTURE	16
4.1 DEPLOYMENT CRITERIA	16
4.1.1 <i>Device Capabilities</i>	16
4.1.2 <i>Profile Management and Configuration</i>	16
4.1.3 <i>Data Attributes</i>	17
4.1.4 <i>Types of Applications</i>	17
4.1.5 <i>Transmission Requirements</i>	17
4.2 INTRANET MANAGEMENT	17
4.2.1 <i>Introduction</i>	18

4.2.2	<i>Motivation</i>	19
4.2.3	<i>DHCP-based Management</i>	20
4.2.4	<i>Deployment Issues</i>	24
4.2.5	<i>Implementation Issues</i>	26
4.2.6	<i>Conclusion</i>	27
4.3	SECURITY ISSUES	27
4.3.1	<i>Authentication, Authorization, and Accounting (AAA)</i>	27
4.3.2	<i>Comparison</i>	28
4.3.3	<i>Security Issues for Peer-to-Peer Architecture</i>	30
4.4	DHCP-BASED MAC-LAYER USER AUTHENTICATION AND ACCESS CONTROL	31
4.4.1	<i>Introduction</i>	31
4.4.2	<i>Managing a LAN: Using DHCP and Firewalls</i>	33
4.4.3	<i>Host Identification – MAC Address Authenticity</i>	34
4.4.4	<i>User Management – the Operation of DHCP with User Registration</i>	34
4.4.5	<i>User Management – the Kernel Lease Table (KLT)</i>	35
4.4.6	<i>Discussion</i>	36
4.4.7	<i>Conclusion</i>	37
4.5	ENHANCEMENTS FOR USER AUTHENTICATION AND ACCESS CONTROL	38
4.5.1	<i>Introduction</i>	38
4.5.2	<i>Motivation</i>	38
4.5.3	<i>The Infrastructure</i>	39
4.5.4	<i>Data Structures and Operations: ACL</i>	41
4.5.5	<i>Design and Implementation Issues</i>	41
4.5.6	<i>Discussion</i>	42
4.5.7	<i>Conclusion</i>	42
CHAPTER 5 APPLICATIONS OF LOCATION SERVICE		43
5.1	PEER-TO-PEER MAIL TRANSFER MECHANISM.....	43
5.1.1	<i>Introduction</i>	43
5.1.2	<i>Motivation</i>	44
5.1.3	<i>Infrastructure</i>	46
5.1.4	<i>Advantages</i>	48
5.1.5	<i>Implementation Issues</i>	49
5.1.6	<i>Future Work</i>	51
5.1.7	<i>Conclusion</i>	51
5.2	PEER-TO-PEER SUPPORT FOR FILE TRANSFER AND CACHING MECHANISM	52
5.2.1	<i>Introduction</i>	52
5.2.2	<i>Motivation</i>	53
5.2.3	<i>Infrastructure</i>	55

5.2.4	<i>Advantages</i>	58
5.2.5	<i>Security Concerns</i>	58
5.2.6	<i>Future Work</i>	58
5.2.7	<i>Conclusion</i>	59
5.3	PEER-TO-PEER SUPPORT FOR MOBILE IP SCHEME	59
5.3.1	<i>Introduction</i>	59
5.3.2	<i>Mobile IP</i>	60
5.3.3	<i>Our Architecture</i>	61
5.3.4	<i>Operations</i>	62
5.3.5	<i>Key Components</i>	62
5.3.6	<i>Design Issues</i>	64
5.3.7	<i>Comparisons</i>	65
5.3.8	<i>Conclusion</i>	67
5.4	EXPERIMENTAL RESULTS.....	68
5.4.1	<i>Experimental Environment for P2P Mail Transfer Support</i>	68
5.4.2	<i>Results</i>	69
5.4.3	<i>Discussions</i>	72
CHAPTER 6 DISCUSSIONS AND CONCLUSIONS		75
6.1	DISCUSSIONS	75
6.2	CONCLUDING REMARKS.....	76
6.3	FUTURE WORKS	76
REFERENCES.....		77

中文摘要

現今網際網路環境中，有各式各樣的網路應用，例如：電子郵件(e-mail)、WWW (World-Wide Web)、FTP (File Transfer Protocol)、串流多媒體影音(streaming audio/video)，以及 VoIP (voice over IP)等皆廣為使用。然而，不同的服務必須透過不同應用程式來存取，這常造成使用上的不便，且也使得服務上的整合 (service integration) 更加困難。一般而言，這些應用所採用的 client-server 架構中，由於伺服器(server)必須處理所有用戶端 (client)的需求 (request)，且亦視應用不同還須有額外的處理，例如：郵件過濾或網頁內容過濾等，而這些額外的處理常使得伺服器負荷過重。而且，在整個過程中伺服器又得針對各種應用的差異而有不同的個人化 (personalization)設定，這又更加重了伺服器的負擔。

因此，本論文提供一個輔助網際網路應用的混合式對等網路架構 (hybrid peer-to-peer architecture)，增設定位服務 (location service)，使我們可以找出任何一個行動主機 (mobile node)目前的上線狀況和所使用設備的能力 (device capabilities)，及其最新的位置，例如：使用者的最新連絡方式或資源所在的 IP 位址。無論我們想連絡的人或想要存取的資源，皆可在任何時間以各種不同設備存取得到。且透過伺服器的 load balancing，使得現有的網際網路應用效能也能大獲改善。故在此架構下，每個使用者或是個人化的設定都可以很容易地實作 (implement)出來，進而達到更好的 customization。

Abstract

In current Internet environment, various kinds of applications exist, for example, e-mail, WWW (World-Wide Web), FTP (File Transfer Protocol), streaming audio/video, and VoIP (voice over IP). However, each service has to be accessed with different application programs, and this makes service integration more difficult. Moreover, in client-server architecture widely deployed in these applications, server load is higher since extra processing is needed besides handling excessive client requests, for example, mail filtering or web content filtering. Most importantly, personalization settings for all kinds of services would further increase server load.

Therefore, in this dissertation, a hybrid peer-to-peer architecture for Internet applications is proposed. With the deployment of location service, the current online status, device capabilities and latest location of a mobile node can be dynamically determined, for example, the current way of contact for users or the current IP address for resources. We can access a mobile node at any time with any device, and existing Internet applications can be improved by offloading the server. Moreover, personal configurations for each user and resource can be easily implemented for each service, thus achieving better customization for each individual user.

Chapter 1 Introduction

With the exponential growth of the Internet, various networking applications are able to develop rapidly. For example, WWW (World-Wide Web), E-mail, FTP (File Transfer Protocol), Streaming Audio/Video, and VoIP (voice over IP) are widely used. The use of e-mail and WWW, for example, has been ubiquitous for the delivery of important messages, advertisements, latest news, and the like.

In order to access different services, separate application programs are required which users must learn to use each one of them. It's not user-friendly, and service integration is not easy.

Client-server architecture has been adopted in these applications in which server plays a critical role. The heavy load of server comes from servicing the vast amount of client requests and also some extra processing needed in these applications, for example, mail filtering and web content filtering. When personalization configuration is concerned, server load becomes even heavier.

In existing Internet architecture, performance optimization is not always considered. For example, mail transfer mechanism is done via SMTP (Simple Mail Transfer Protocol) [1] where mail servers are critical in each step of mail delivery. With its *store-and-forward* design, the purpose is to make sure each mail is delivered correctly to the destination. However, mail servers are mandatory in the path of mail delivery, from sender mail server, mail exchanger, to receiver mail server. Mails get queued on each mail server, and delivered when appropriate. This takes a lot of processing time, network bandwidth, and storage.

When receiver gets online, he/she will first check if e-mails are available for him/her via POP3 (Post Office Protocol Version 3) [2]. However, garbage mails as well as critical mails are all stored in user mailbox on mail server. Usually, mail filtering can be applied in two different places: user client or mail server. On the first hand, mail filtering on client side can facilitate personalized filtering rules for each user. However, only after users retrieve all of their e-mails, and apply mail filtering rules or check manually will unwanted mails get deleted. This is not only a waste of server storage, but also a waste of precious network bandwidth and processing time for delivering these garbage mails. Although IMAP4 (Internet Message Access Protocol Version 4) [3] has been designed to address this problem by providing more advanced functionalities like prefetching the mail headers and synchronization between client and server mailboxes, POP3 has to be deployed since IMAP4 is not widely implemented. On the other hand, mail filtering on the server side is not feasible with the large overhead for filtering rule enforcement, especially when personalization is desired in filtering rule configuration.

In the case of WWW, all web pages are stored in web server, and requests from

browsers will be responded with corresponding pages. Although proxy server has been designed to cache the files from FTP or web server in order to reduce repetitive outward file accesses, inter-cache coordination and cooperation are still not much utilized. Data already cached by other peer proxies will improve the processing time and necessary bandwidth if carefully utilized.

Therefore, peer-to-peer technology has received increasing attention besides client-server architecture. In the last two years, many peer-to-peer systems, such as instant messaging software like ICQ and MSN messenger and file sharing software like Napster [4] and GnuTella [5], have been used worldwide. More and more researches are also focusing on this field [6, 7]. Peer-to-peer applications can be roughly divided into three categories: distributed computing, file sharing, collaboration and communication. The common features in these peer-to-peer applications are to aggregate distributed resources, like computing power, storage and content, and network bandwidth, and efficiently fulfill the goal of information exchange by direct communication among peer hosts.

For peer-to-peer architecture, we can divide it into two categories: pure and hybrid. In pure peer-to-peer architecture, each peer host is equal in capabilities. All data exchange has to be done in two phases. In the first phase, peer hosts with desired data are searched via broadcasting methods like request flooding. After locating the peer host, real data exchange can take place directly between peer hosts. On the other hand, for hybrid peer-to-peer architecture, there is one or more index server(s) or “super” peers that contain meta information like the location of peer hosts and the index of data content on them. Peer hosts with desired data only have to be searched via queries to the index servers. After that, direct communication among peer hosts can proceed as in the former case. Therefore, for searching phase, it's time-saving for hybrid peer to peer architecture that was adopted in this dissertation.

With the fast development of networking media and mobile devices, wireless LAN [8] has become another popular way of network access. There are two modes of operations in wireless LAN: *infrastructure* and *ad hoc*. In *infrastructure* mode, mobile nodes can connect to the wired network via access point (AP). In mobile environment, each user can be moving anytime. Roaming into the range of different APs belonging to different subnets is possible. Under such circumstances, IP address may be changed and TCP/IP protocol stack as well as upper layer applications will be affected.

The most common solution is to use mobile IP scheme [9] where no modification to hosts is necessary. Mobile nodes need not change their IP addresses, and only mobility agents are required for achieving routing transparency. Mobility agents are responsible for the packet redirect and tunneling. Although mobile IP tries to address the IP roaming problem by allowing routing transparency, it's not feasible to adopt this scheme because of its large overhead for *triangle routing*. All packets destined for mobile nodes roaming into

foreign networks must be redirected via home agent (HA) in home network to the foreign agent (FA) in foreign network, which in turn forwards the packets to mobile node. Therefore, large overhead will result, especially in frequent handovers.

Therefore, in this dissertation, a hybrid peer-to-peer architecture for augmenting Internet applications is proposed. Location service is introduced into existing Internet architecture, and servers as well as clients can dynamically determine the online status and the latest location for mobile nodes, for example, way of contact for users and current IP address for resources. On the other hand, Internet applications can thus be improved by offloading servers and providing personalization support on each client, for example, access control list (ACL) for mail filtering or file access. Better customization for individual users can be supported. In a word, the main contribution of this dissertation is to address the mobility problem introduced into existing Internet applications in wireless environments and to provide a way of load balancing for overloaded Internet servers like mail servers, web servers, proxy servers, ... etc.

The dissertation is organized as follows: the background information and related works are briefly reviewed in Chapter 2. In Chapter 3, the proposed hybrid peer-to-peer architecture is introduced along with its mechanism and basic operations. The management of the architecture will be covered in Chapter 4, where intranet management and security issues will also be discussed.

Then, the design of various applications of our architecture will be introduced in Chapter 5 including peer-to-peer support for mail transfer, file transfer, caching and mobile IP scheme. More importantly, the experimental results supporting the main contribution of this dissertation is also provided in Chapter 5. Finally, some discussions and concluding remarks will be given in Chapter 6.

Chapter 2 Background Information and Related Works

In this chapter, I will briefly introduce some background information for this dissertation. Firstly, the concept of peer-to-peer computing will be introduced and the comparisons between client-server and peer-to-peer architectures will be made. Secondly, the evolution of network technologies will be described including the latest wireless LAN and mobile computing.

2.1 Taxonomy of Computer Systems

First of all, a taxonomy of computer systems is illustrated in Fig. 1.

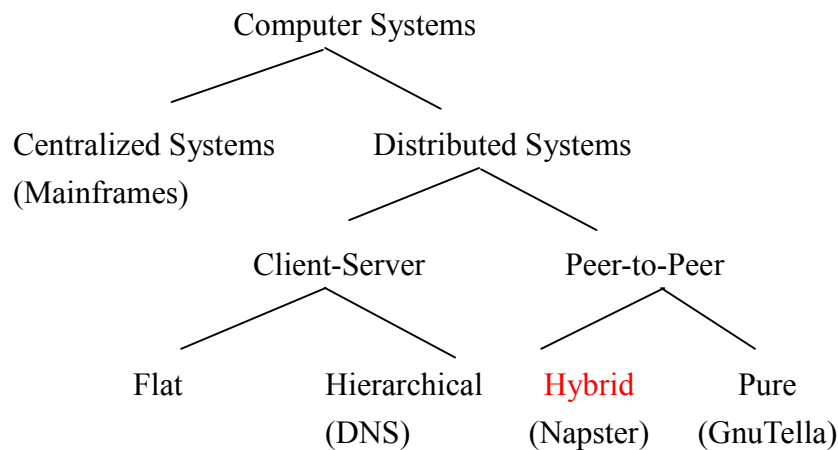


Fig. 1 shows a taxonomy of computer systems.

2.1.1 Centralized vs. Distributed

Broadly speaking, computer systems can be divided into two categories: centralized systems and distributed systems. Old mainframes and workstations belong to the first group, while most of the current systems are distributed. Since the centralized system rely on the availability and computing power of the only computer system, once it fails the service disrupts. Therefore the scalability of centralized system is bad because of *single point of failure* problem. On the other hand, distributed systems try to distribute computing efforts across different system components thus providing better scalability.

2.1.2 Client-Server vs. Peer-to-Peer

Distributed systems can be further divided into two sub groups according to the relationship between hosts. Client-server model is most widely deployed since the development of the Internet is based on client-server model. For hosts in client-server model, the roles are not equal. Servers are passive in connection establishment and they wait for incoming connections to be made. On the other hand, clients are active in connection

establishment and they make connection requests to servers. Most Internet applications are designed in client-server model.

For peer-to-peer model, all hosts are equal in role. No special functions are provided on specific hosts since every one is the same. Every host is both a client and a server. For example, the host in GnuTella protocol [10] is called *servent* to signify this point.

Client-server model can be divided into flat and hierarchical. Flat client-server is not common in practice since a single server serves all clients. However hierarchical client-server is very common in Internet operations in which servers are organized as a hierarchy, for example, in Domain Name Systems (DNS) [11].

2.1.3 Pure vs. Hybrid Peer-to-Peer

For peer-to-peer model, we can further divide it into two categories: pure and hybrid. For pure peer-to-peer systems, every host is the same in functionality. No individual index server or “super” peer is provided. For searching the other peer hosts, each host begins with flooding or broadcasting requests. Once peer hosts with the requested data are located, direct communication between peer hosts is conducted.

For hybrid peer-to-peer systems, index servers or “super” peers are deployed as the storage for meta information of peer hosts and their contents. For locating the data wanted, each host will query the index server. Once peer hosts are located, direct communication between peer hosts will be conducted as in pure peer-to-peer systems.

In this dissertation, I will focus on hybrid peer-to-peer model deployed in our proposed architecture.

2.2 Evolution of Networking Technologies

With the fast development of the Internet, networking technologies have undergone tremendous growth. Networking devices commonly used in LAN (Local Area Network) include hubs, switches, and routers. With the advent of wireless LAN, devices bridging wired and wireless networks such as access points (APs) are needed. In the following sections, wired and wireless networks will be briefly introduced.

2.2.1 LAN Technologies

In conventional Ethernet, hubs are used as a multiport repeater connecting local hosts. Traffic generated at one port will be forwarded to all other ports in a hub. However, since the nature of Ethernet is CSMA/CD (Carrier Sense Multiple Access/Collision Detection) bus, as the number of hosts in a domain grows, the chance of packet collision becomes much higher. Therefore, bridges are commonly adopted in a local area network to physically separate different segments of networks and unnecessary packet collisions can be avoided among different hosts. For example, consider a small enterprise consisting of several

departments in the same building. Traffic inside each department has better be contained within its own collision domain without interfering with other departments.

As the number of hosts grows, the extraordinary broadcast packets may cause unnecessary traffic to be flooded across bridges in the whole LAN. Therefore router goes one step further in containing broadcast packets in each domain. As new technology evolves, switches are getting more attention. Layer 2 switches are just bridges with more fancy features such as VLAN (virtual LAN) [12] and full-duplexing on separate port, and layer 3 switches incorporate network layer address handling functions except routing. In such environment, all packets must go through these switches before reaching other hosts.

2.2.2 Wireless LAN

Network planning had to accommodate building structure and wiring in the old days, and it's usually annoying and complicated. Thanks to the new transmission media, we may also deploy wireless LANs [8] as less wiring is needed in most of the offices. In such cases, wireless access points APs become the bridge between wired and wireless networks.

IEEE 802.11 wireless LAN has gained more and more popularity with the rapid growth of mobile communication devices like PDAs, portable computers, ... etc. The possible application of wireless LAN is ubiquitous, from schools and organizations, to stations and airports. Starting from the basic 802.11 standard, there are 802.11b [13], 802.11a [14], and more recently, 802.11g, the high-speed extension to 802.11b. As the data rate grows from 1Mbps to 54Mbps, the problems inherent in the 802.11 standard are becoming evident and critical as we already faced in wired networks: Quality of Service (QoS), and security issues. Since the basic security mechanism in 802.11 called WEP has been proved [15, 16, 17] to be vulnerable to attacks, IEEE 802.11 Working Group is still making great efforts to improve the wireless security. Moreover, the access points (APs) from different vendors could very possibly not interoperate with each other across the same distribution system because of the flexibility in the real implementation of 802.11 standard.

Therefore, 802.11 Working Group has activated Task Groups 802.11e [18], 802.11f [19], and 802.11i [20], for the issues of MAC layer Quality of Service (QoS), Inter-Access Point Protocol (IAPP), and MAC layer security, respectively. All these standard specifications are still in the phase of IEEE Drafts.

Chapter 3 Proposed Hybrid Peer-to-Peer Architecture

Location management is a major issue in mobile computing environments. Current researches have focused on the geographical positioning of mobile nodes, not the current way of contact for mobile users that is more critical for Internet applications to be supported in wireless networks. Although service location protocol [21] has been proposed, it's primarily for servers, not clients.

In this chapter, a generalized user location and profile management protocol was introduced for different levels of applications. Location information and profiles for domain users and resources are kept in a domain location server that can be updated and queried when necessary by servers and clients. In this way, mobility and personalization for current Internet applications can be better supported.

3.1 Introduction

With the advent of mobile devices, like personal digital assistants (PDAs) and portable computers, wireless networks have undergone a tremendous development and received more attention from industries and academies. Nowadays, users are not always connecting to the Internet via a fixed host at a fixed location. Users may be moving around the building using a mobile device while accessing Internet servers.

Since Internet hosts are not necessarily fixed nowadays, their points of attachment may change. Conventionally, in the TCP/IP networking model, IP address is the equivalent of host location, and uniquely identifies the point of attachment to the Internet. However, as mobile nodes change their locations, their IP addresses may change as well when moving across the boundary of different subnets. Therefore, several IP-layer roaming or handover problems need to be addressed as far as mobile nodes are concerned.

Firstly, existing connections in a mobile node may be interrupted when IP roaming takes place. Since the change of IP address will affect the proper working of TCP/IP protocol stack, existing Internet connections will be interrupted. After changing their IP addresses, either manually or automatically via DHCP (Dynamic Host Configuration Protocol) [22], users have to manually re-connect to the services they were accessing.

Secondly, after IP roaming, new incoming connections cannot be made for a mobile node if the latest location is unavailable. We may need to contact a mobile user or access a resource with unknown location since their mobile nature. This may happen in peer-to-peer applications like instant messaging and file sharing where each host is able to receive incoming connections. With conventional DNS (Domain Name Systems) [11] lookups, we may only obtain its fixed IP address which may change over time for a mobile node. Although the combination of DHCP and DDNS (Dynamic DNS) [23] can be used, it's not

mandatory and not necessarily implemented. For mobile servers, Service Location Protocol [21] deals with the discovery, location, and configuration of servers, but not ordinary clients.

Therefore, mobile IP [9] has been proposed for resolving the problem of IP-layer roaming. Instead of changing the IP address of mobile nodes, a *care-of address* is obtained when a mobile node roams into a foreign network, and two mobility agents, *HA (Home Agent)* and *FA (Foreign Agent)*, are responsible for redirecting and tunneling the packets destined for the *home address* of mobile node to its *care-of address*.

The main drawback for mobile IP is that packet tunneling and triangle routing required between *HA* and *FA* takes too much overhead when mobile node is not in home network. Proposals for route optimization [24] in mobile IP environment try to address the problem of routing overhead by caching the binding of mobile nodes. But service disruption problem still exists in mobile IP scheme.

Therefore, in this chapter, we introduced a new mechanism for maintaining the current “location” of each user and resource in a domain. The “location” we mentioned here is the latest way to contact desired resource. In each domain, a location server is responsible for managing the locations of domain users and resources. For location information to be available, there is an agent on each host that dynamically updates the current user/resource location into the corresponding location server. When making connection requests to other resources, the agent is also responsible for sending queries to domain location server in order to get the current status of the user/resource he wants to access.

3.2 Location Service and Profile Management

Location management is a major issue for mobile nodes since their mobile nature. When connecting to a mobile node, we need to know exactly where it is and how to route packets to it. But many existing solutions on location management [25] focus on the geographical location positioning of mobile nodes. Service Location Protocol [21] deals with the discovery and location of servers, but not ordinary clients which we may want to directly communicate with in peer-to-peer applications. In Session Initiation Protocol (SIP) [26], multimedia sessions can be established, modified, and terminated via SIP proxy/redirector, and the location of users are also stored in a location server. However, the location server in SIP is only for multimedia session management, not for the other common Internet applications.

Specifically, what we need is a location service, for maintaining the location of Internet users and resources, not just a domain name service mapping hostnames into IP addresses or vice versa. Conventional DNS lookup could not satisfy the dynamic needs of the mobile environments. Moreover, with the advent of peer-to-peer computing, many applications such as instant messaging (IM) and file sharing can be utilized in very different

manner from conventional TCP/IP networking applications such as e-mail, FTP, and web browsing. For example, in a possible scenario depicted in Fig. 2, user u_i may have a list L_i of possible ways of contact, e.g., via IM, e-mail, and a mobile phone, with corresponding relative precedence or priority. Since the online status and current location of a user is registered to location server, we can decide the order of different ways of contact to try, their precedence, and fallback sequence.

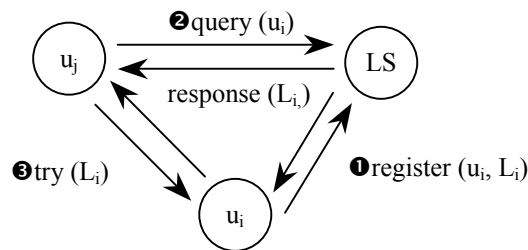


Fig. 2 shows a possible scenario where user u_j wants to contact user u_i . Location server (LS) supports a list L_i of multiple ways of contact for user u_i with corresponding precedence.

Besides location information, user profiles for each application can also be recorded in the location server for more flexibility in applications. For example, each user may want to configure one's own settings for different applications. With the support from infrastructure, personalization of Internet applications for each user becomes easier easy to achieve without much overhead. Furthermore, service availability will be improved since we can automatically fall back to different mechanisms according to the status of both parties in different applications. This facilitates the auto-configuration and fault tolerance of many Internet applications.

3.3 Management Records

In common applications the objects we may want to access can be divided into two categories, that is, user and resource, which need mobility and configuration management. Specifically, user/resource profiles and location information have to be stored. Therefore, four different types of records are possible in our location server: User Location Record (ULR), User Profile Record (UPR), Resource Location Record (RLR), and Resource Profile Record (RPR).

1. Location Records

For location records, current way of contact for user or resource is kept. In a User Location Record (ULR), the following information has to be kept in order to keep track of current user status, for example, user name, e-mail address, phone number (mobile), current IP address, and the current online status of users.

In a Resource Location Record (RLR), the information about resources is kept: resource name (URN), resource identifier (URL), alias, current host address, current status

of resources.

2. Profile Records

For user profiles, there are two kinds of profiles to deal with. The first is the personal way of contact for other people to reach the user. The second is for configuring the personal preferences for the user to access different services. In a User Profile Record (UPR), the following information has to be kept in order to keep track of current user status, for example, user name, e-mail address, phone number (mobile and PSTN phone), current IP address, and the current online status of users.

For resource profile, it's the access right or configuration for resources, for example, an access control list (ACL) for a specific group of resource.

3.4 Basic Operations

In our infrastructure, there will be an agent for each user or resource on a host. It is responsible for interacting with the location server. Generally, there are three types of basic operations between location server (LS) and the agent.

1. *update*: for agent to modify records in LS, when register/sign-on (login), de-register/sign-off (logout), register or de-register a notifier of events.
2. *query* (lookup): for agent to lookup in LS.
3. *notify*: for LS to notify some event to agents.

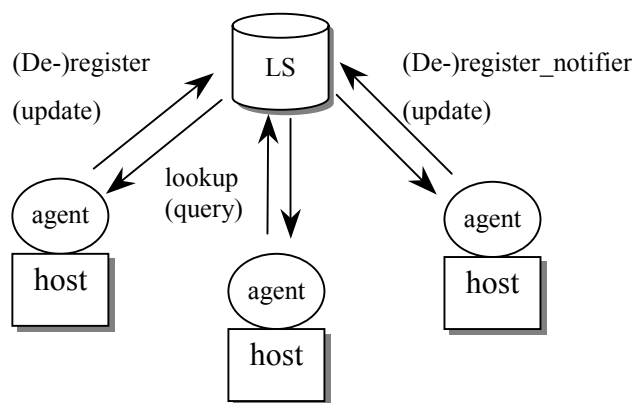


Fig. 3 The basic operations of location server.

However, for each of the four different records in location service, the detailed operations are slightly different in which different data are manipulated.

For ULR:

- (1) Update (u_i, L_i)
- (2) Query ($u_i \rightarrow L_i$)

where $L_i = \{(IP_j, port_j, p_j) \mid \text{for } j=1, 2, \dots, n\}$ is a list of n possible ways of contact ($IP_j, port_j$) for user u_i with precedence p_j . The unique identifier for user u_i can be his e-mail

address.

For RLR:

(1) Update (r_i, L_i)

(2) Query (r_i) $\rightarrow L_i$

where $L_i = \{(IP_j, port_j, p_j) \mid \text{for } j=1, 2, \dots, n\}$ is a list of n possible replications ($IP_j, port_j$) for resource r_i with precedence p_j . The unique identifier for resource r_i can be the URL (Uniform Resource Locator) or URN (Uniform Resource Name) [27].

For UPR:

(1) Update (u_i, L_i)

(2) Query (u_i) $\rightarrow L_i$

where $L_i = \{(attr_j, value_j) \mid \text{for } j=1, 2, \dots, n\}$ is a list of n attribute-value pairs for the profile of user u_i . The unique identifier for user u_i is the e-mail address.

For RPR:

(1) Update (r_i, L_i)

(2) Query (r_i) $\rightarrow L_i$

where $L_i = \{(rule_j) \mid \text{for } j=1, 2, \dots, n\}$ is a list of n rules for the access control list of resource r_i .

3.5 Applications

There are different layers of applications for User/Resource Location/Profile Management Protocol: application layer and network layer. At application layer, peer-to-peer and mobility support can be added for applications like mail transfer (SMTP), and file transfer and caching (FTP/HTTP). At network layer, location service can also be applied in improving the routing efficiency for mobile IP scheme [9]. More details for the applications will be discussed in chapter 5.

3.6 Security considerations

In this framework of location profile management protocol, security concerns are among the most important since the location and profile configuration in location server is critical for correctly contacting users and accessing resources in mobile environments. When designing such as framework, security is taken into consideration, especially, authentication and access control for user/resource location/profile management records.

For each of the basic operations mentioned above, several steps have to be done. Firstly, authentication has to be done to ensure that the one making requests is indeed as the client claims. Secondly, access control is enforced to ensure that the client really has the appropriate right to do this operation. Thirdly, only for *UPDATE* operations only, we should also validate the information content the client provides. Finally, after doing all the above validity checks, we can really service the requests and make appropriate changes. For

example, for updating a ULR, a request will be received as follows:

Update (user_i, IP_j)

- (1) User authentication for user_i is performed.
- (2) Access control list for user_i is checked to see if update operation is allowed for the requestor.
- (3) Verify if IP_j is a valid IP address.
- (4) Update the pair (user_i, IP_j) into ULRs.

Another example for updating a RLR is as follows:

Update (URL_i, server_j)

- (1) Host authentication for server_j is performed.
- (2) Access control list for URL_i is checked to see if update operation is allowed for the requestor.
- (3) Verify if server_j is a valid host.
- (4) Update the pair (URL_i, server_j) into RLRs.

In this section, only security concerns for the location service are described. Further details on more security issues for the whole architecture will be discussed in Chapter 4.

3.7 Design and Implementation Issues

In this section, the design and implementation issues of location service (LS) will be discussed.

3.7.1 Location Server Discovery

There may be the problem of how to find the location server itself for each host. Since each mobile node may be moving around and roaming into foreign networks, to locate the home LS will require several steps. Firstly, network access from the foreign network has to be obtained. This would require authentication and authorization to be passed for that foreign network. Secondly, after gaining the access to the network, the mobile node will have to update its current location and profile either directly or indirectly through the LS in that foreign network. If direct update is used, then each host must have the ability to discover its own home LS through methods like broadcasting or querying the foreign LS for the location of home LS. Otherwise, if indirect update is used, foreign LS will be queried and the discovery of home LS will be carried out by foreign LS. Then there should be an inter-LS communication protocol for the discovery of neighbor LS.

3.7.2 Recursive vs. Iterative Queries

As in the case of DNS servers, queries can be classified into two different modes: *recursive* and *iterative*. *Recursive queries* can be done when a location server receives a query from either client or server, and returns the final results after querying other location

servers on behalf of the requestor. On the other hand, *iterative queries* will be preferred when a location server only redirects or refers the requestor to the possible location server with wanted data. No query overheads will be needed for location server in iterative query mode.

3.7.3 Implicit vs. Explicit LS Queries

Since the location service query can be issued by either servers or clients, there may be two types of queries in terms of the query requestor. Firstly, *explicit query* is possible when clients need to look up for the location records themselves. After getting query results, clients will be able to connect directly to the peer host supporting the requested functionality. Note that *explicit query* is more like *hybrid* peer-to-peer operations since the peer host can directly communicate with each other after locating peer hosts. However, client support for such operation is required for both sides of connection parties.

Secondly, there are *implicit queries* that are issued by application servers, for example, mail servers, web servers, or proxy servers. Since this kind of queries is issued by application servers, communication protocols or even application servers need to be modified to support this option. There may be three modes of operations for implicit queries, *redirect* mode, *proxy* mode, and *server-to-server copy* mode. These will be further discussed later in Chapter 5.

3.7.4 Inter-LS Communication

Since the design of location service is hierarchical, communication among location servers in different domains is possible in the dynamic environment of mobile nodes. For inter-LS communication to occur, there may be three possible cases, namely, *location server discovery*, *recursive query*, and *indirect update*. Firstly, when roaming into foreign networks, any update of location records will be sent to home LS. However, in order to search for its home LS, communications between location servers are necessary. Secondly, query operations across different domains are possible since we may need to contact users or access resources in foreign domains. Finally, update operations must be maintained by its home LS. Since mobile nodes may be roaming into foreign networks, location server update is managed by the home LS. For example, the home LS for a user is the LS in the domain part of his e-mail address. The home LS for a resource is the home LS for the peer proxy which issues the update request.

Chapter 4 Management of the Architecture

In this chapter, I'm going to describe the management issues of our architecture. Firstly, the deployment criteria for the architecture are discussed. Secondly, intranet management strategies used in the architecture are illustrated. Thirdly, security issues are described.

4.1 Deployment Criteria

Since our architecture tries to accommodate the advantages of both client-server and peer-to-peer architectures, flexibility, security, and scalability are the major concerns. When deploying our architecture, several criteria need to be considered. There are four types of criteria that will be considered: device capabilities, profile management and configuration, data attributes, and types of applications. Usually, different requirements on data include: synchronization, replication, bandwidth utilization, and response time. According to the following criteria, our infrastructure can adjust accordingly and provide the best support for the communication parties.

4.1.1 Device Capabilities

The communication parties can be hosted on different kinds of devices besides desktop computers, for example, personal digital assistants (PDAs), portable computers, and even input/output devices like scanners and printers that are connected to the network. Therefore, different levels of capabilities must be negotiated before the best communication quality can be determined. The possible capabilities include the computing power, memory and storage size, input/output capabilities such as audio/video playback/recording support, resolution of rendering, and the protocol support for each application.

4.1.2 Profile Management and Configuration

Each user may have different precedence on configuration settings of each application. These could affect the operations of our infrastructure. For example, user u_a may have different ways of contact, say ICQ, e-mail, and mobile phone. User u_a can configure the precedence of these different ways of contact to be of the same order as shown above. When other users want to reach him, the universal communication client will first need to locate the user before deciding which way to try. If user u_a cannot be reached via ICQ, then the client will try the method with second precedence, e-mail. If all of above fails, then the mobile phone number will be called via VoIP through the voice gateway as configured in user profile for u_a .

4.1.3 Data Attributes

For different types of data, our infrastructure will provide necessary support. For example, for urgent data that need to be transmitted immediately, we will focus on system response time. For bulk data, network bandwidth requirement has to be met before transmission can be conducted. For personal, sensitive, and critical data, synchronization and replication will be the major concerns.

4.1.4 Types of Applications

For each type of application, there will be different communication requirements. For example, mail transfer applications require the successful delivery of e-mails, while multimedia file sharing applications require network bandwidth to be reserved since the amount of data could be large. Instant messaging requires the immediate delivery of messages and thus response time is the concern. For data in collaboration application, synchronization and replication will be the considered as the major concerns.

4.1.5 Transmission Requirements

For each transaction, different requirements are possible due to various factors. For example, we may want our transaction to be done as fast as possible. On the other hand, we may want to minimize the storage requirement due to the small size of storage on embedded systems. Other factors include reliable or secure transmission, and minimal power consumption, computing power, memory requirement.

4.2 Intranet Management

DHCP (Dynamic Host Configuration Protocol) [22] is widely deployed in resource allocation and intranet management. However, DHCP mechanism is not mandatory, and DHCP server can neither force DHCP clients to release their leases, nor enforce cooperation from externally configured hosts that are DHCP-unaware. Although new DHCP options such as DHCP reconfigure extension [28] have been proposed, the basic problems inherent in DHCP mechanism cannot be solved without first strengthening its operations.

In this section, a DHCP-based infrastructure for intranet management [29] was proposed by combining the resource allocation functions of DHCP server with the packet filtering features of MAC (Medium Access Control) bridges [30] such as Ethernet switches and wireless access points. DHCP clients and DHCP-unaware hosts that do not abide by DHCP mechanism or our management policy will be denied network accesses by MAC bridges. Resource allocation and access control can be integrated and local configuration conflicts can be reduced to the minimum.

4.2.1 Introduction

Network security has continued to be a major issue in all kinds of applications as Internet becomes a necessity. Various types of intrusions and attacks such as DDoS (Distributed Denial of Service) are threatening the enterprises and individuals as well. Unlike attacks from the outside, local conflicts in network configurations have direct impact on the daily operations of the intranet.

The primary concern of intranet management includes allocation of resources such as IP addresses, network configuration of hosts and servers, among others. Manual configuration of hosts is prone to errors and any modification would require human interventions that are time-consuming. Therefore, DHCP (Dynamic Host Configuration Protocol) [22, 31], an extension to BOOTP protocol [32], has become more widely adopted as a mechanism for automatic and dynamic resource allocation and configuration in intranet management. Although it is commonly deployed, some drawbacks inherent in DHCP mechanism may cause more trouble than the benefits it can bring.

First of all, DHCP server cannot force DHCP clients to release their leases. DHCP server only acts as a resource dispatcher, and normally DHCP clients will not release their leases at shutdown. Although the new *DHCP reconfigure extension* option [28] can be used for DHCP server to force a “cooperative” DHCP client to renew its lease, malicious hosts may still be able to allocate new addresses without releasing them at all which would easily exhaust available IP addresses.

Secondly, since DHCP is not mandatory, externally configured hosts may deliberately or accidentally use the same network addresses as DHCP clients. For such hosts, their IP addresses are manually configured and other local network parameters can be obtained via *DHCPINFORM* requests [22]. However, *DHCPINFORM* messages are not commonly implemented. If manually configured IP addresses conflict with DHCP clients without notifying DHCP server, we cannot regulate their misuse and network disaster may occur. Furthermore, the new *DHCP reconfigure extension* option [28] can only be used for cooperative DHCP clients, not DHCP-unaware hosts.

In order to make the most of DHCP, we have to strengthen its power of regulation. New options such as *DHCPINFORM* and *DHCP reconfigure extension* have to be enforced and integrated into the infrastructure to make DHCP clients more manageable. In addition, there must be a mechanism to force DHCP-unaware hosts to cooperate with DHCP management policy. Once non-cooperating hosts are detected, we will alert them by *DHCP FORCERENEW* or RHCP (Remote Host Configuration Protocol) [33] messages. That means intranet hosts need to be extended by DHCP/RHCP processing modules to receive instructions from management server, in this case, a DHCP server. If they still don't abide by the instructions, we will restrict their network access rights at bridges. With appropriate enforcement of network access control in MAC bridges, we can compensate the

disadvantages of DHCP mechanism and local conflicts can be reduced to the minimum. On the other hand, MAC bridges can also be enhanced with address allocation flexibility. Mechanisms for access control and notification of invalid connection attempts are possible in this infrastructure.

4.2.2 Motivation

In our previous work [34], a mechanism for extending DHCP capabilities with MAC-layer user authentication was proposed, as shown in Fig. 4

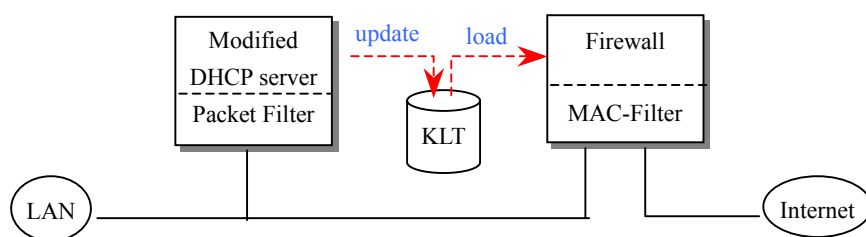


Fig. 4 shows the infrastructure of DHCP-Firewall combination in our previous work [38] where KLT is the Kernel Lease Table that maintains DHCP lease information at kernel level.

As shown in Fig. 4, DHCP server was coupled with firewall in order to regulate local hosts from network address misconfiguration. However, firewalls are not always deployed in all kinds of network configurations although it's better to have one. In ordinary LAN environment, bridges and routers are more widely used.

In conventional Ethernet, hubs are used as a multiport repeater connecting local hosts. Traffic generated at one port will be forwarded to all other ports in a hub. However, since the nature of Ethernet is CSMA/CD (Carrier Sense Multiple Access/Collision Detection) bus, as the number of hosts in a domain grows, the chance of packet collision becomes much higher. Therefore, bridges are commonly adopted in a local area network to avoid unnecessary packet collisions among different hosts. For example, consider a small enterprise consisting of several departments in the same building. Traffic inside each department has better be contained within its own collision domain.

As the number of hosts grows, the extraordinary broadcast packets may cause unnecessary traffic in a LAN. Therefore router goes one step further in containing broadcast packets in each domain. As new technology evolves, switches are getting more attention. Layer 2 switches are just bridges with more fancy features such as VLAN (virtual LAN) [12] and full-duplexing on separate port, and layer 3 switches incorporate network layer address handling functions except routing. In such environment, we can actually combine DHCP server with layer 2/3 switches since all packets must go through these switches.

Network planning had to accommodate building structure and wiring in the old days, and it's usually annoying and complicated. Thanks to the new transmission media, we may also want to deploy wireless LANs [8] as less wiring is needed in most of the offices. In

such cases, wireless access points become the bridge between wired and wireless networks.

4.2.3 DHCP-based Management

1. Infrastructure

As a matter of fact, we can enforce access control in whatever types of MAC bridges. Our main idea is to combine the resource management function of DHCP server and the access control function of bridges. Manually configured hosts are encouraged to utilize *DHCPINFORM* or *RHCP* messages to inform DHCP server of their network address configurations. Alternatively, a simple registration step may be used for each new user or a user with a new NIC (network interface card) prior to his first Internet connection as in our previous results [34]. As shown in Fig. 5, a general infrastructure for DHCP-based management is illustrated.

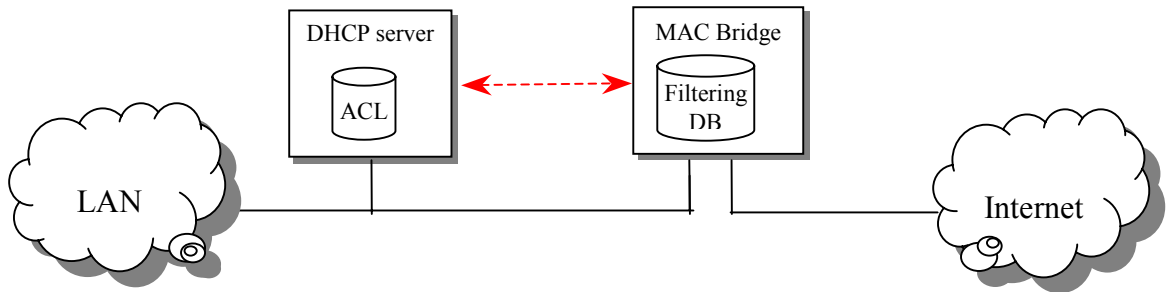


Fig. 5 shows the basic infrastructure of DHCP-Bridge Combination.

The idea is simple: we keep track of an access control list (ACL) of hardware address and network address pairs for authorized hosts, namely (*MAC*, *IP*) pairs, and then enforce the ACL by the Filtering Database in MAC bridges [30]. Our policy is to protect those hosts that are pre-configured (externally configured hosts like servers), registered, or DHCP-aware. For all other hosts, we will not protect their packets from being filtered. All packets with unauthorized (*MAC*, *IP*) pairs will be dropped by bridges.

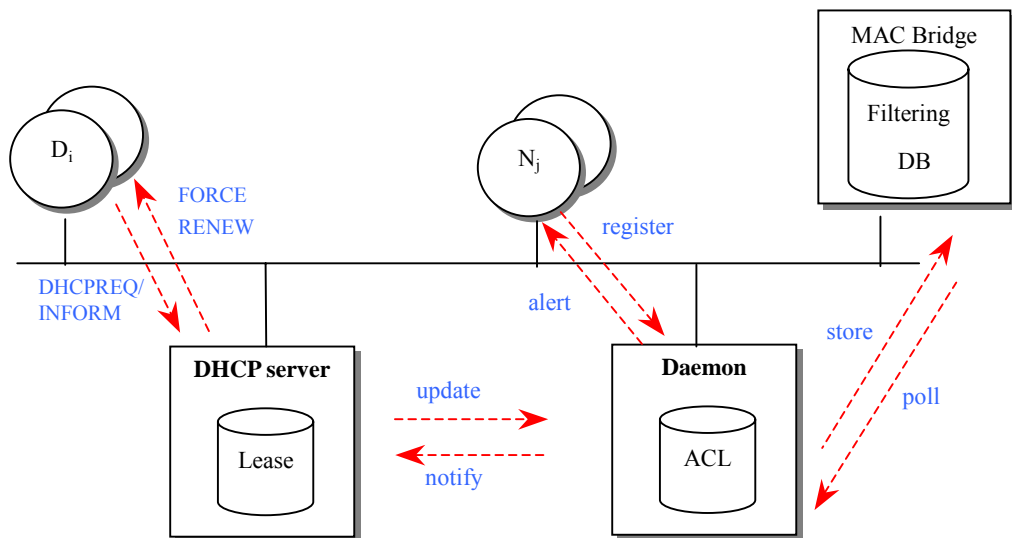


Fig. 6 shows the interactions among DHCP server, Monitoring Daemon and MAC bridge, where D_i denotes DHCP clients and externally configured hosts that are DHCP-aware, and N_j denotes DHCP-unaware hosts.

In Fig. 6, a common network configuration where hosts are connecting through MAC bridges to the Internet is illustrated. In this infrastructure, two components are needed: DHCP server for resource allocation and a monitoring daemon for keeping an access control list (ACL). ACL is corresponding to the Filtering Database in MAC bridge which actually performs packet filtering and forwarding. On the one hand, monitoring daemon is responsible for receiving ACL update requests from DHCP server and enforcing ACL modifications into Filtering DB on MAC bridge. On the other hand, it is responsible for polling statistics of packets flowing through MAC bridges, and sending notifications of illegal connection attempts to DHCP server. Therefore, it is the “bridge” or “proxy” between the DHCP server and MAC bridges.

For DHCP-unaware hosts, registration is needed as an authentication for hosts. In our infrastructure, registration server can be put on the same host as monitoring daemon. Therefore, monitoring daemon is also responsible for receiving registration requests and sending Force Register messages in response to illegal connection attempts from hosts without registration.

2. Basic Operations

The basic operations among the key components of our infrastructure for DHCP-based management work as follows:

(1) Hosts Authentication and ACL Collection

For DHCP clients, it's mandatory to make lease allocation or renewal requests (*DHCPDISCOVER* / *DHCPREQUEST*) to DHCP server. It is therefore natural for DHCP server to verify and authenticate their MAC addresses in the process of handling their requests. Note that our DHCP server will check not only the ‘*Client Identifier*’ option but also the ‘*chaddr*’ field [22] in DHCP requests and match them with the authentic MAC address in the Ethernet header of packets. Therefore, only one legal IP address at a time can be allocated for each MAC address, hence for each Ethernet adaptor. This keeps malicious hosts from allocating new addresses without releasing them as described earlier in the introduction, even if malicious hosts are DHCP-aware.

For externally configured hosts, such as intranet servers, system administrator may choose to configure their leases manually in DHCP server, or in a more dynamic way, configure them to notify DHCP server of their externally configured IP address via *DHCPINFORM* messages if supported. Although *DHCPINFORM* is specified in RFC 2131 [22] as a required feature, not many externally configured hosts support this option.

DHCP leases maintained on DHCP server will be translated into ACLs and updated accordingly on daemon, which will then be enforced into Filtering Database on MAC

bridge.

For DHCP-unaware hosts, authentication can be done by our registration server as in [34], and their (MAC, IP) pairs will also be marked as legal in the process of registration.

(2) ACL Enforcement and Notification

Since a transaction log is kept for recording any illegal connection attempts on MAC bridge, packet statistics can be periodically polled by daemon and a list of illegal (MAC, IP) pairs can result.

For DHCP clients on the list, daemon will notify DHCP server which will then send *FORCERENEW* messages to notify DHCP-aware hosts of their illegal (MAC, IP) pairs. This will trigger renewal of DHCP leases.

For DHCP-unaware hosts on the list, they will be alerted directly by daemon via RHCPC messages and re-registration will be triggered.

3. Client-Server Interactions

As illustrated in Fig. 7, there are four possible cases of client-server interactions in our infrastructure. First of all, when DHCP client C_1 obtains its lease through normal DHCP procedures as shown in Fig. 7(a), DHCP server S will inform monitoring daemon D of a valid pair (MAC_{C_1}, IP_{C_1}) . The monitoring daemon will then pass the updated part of ACL to bridge B. Packets from C_1 can then pass through the bridge.

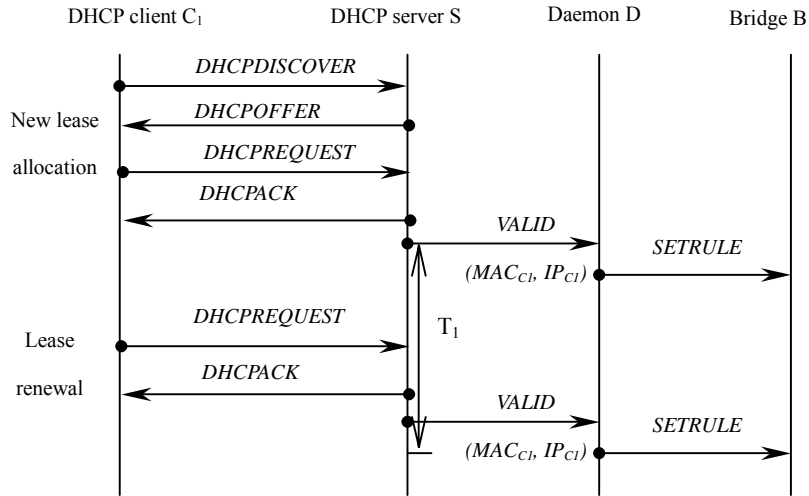


Fig. 7(a) shows the first case of client-server interactions where DHCP client C_1 allocates and renews its lease automatically in normal cases.

Secondly, after time duration T_1 DHCP server S finds out that the lease of DHCP client C_1 will soon expire. If C_1 renews its lease automatically, things will go in its normal way. However, as illustrated in Fig. 7(b), if C_1 doesn't renew its lease, DHCP server will send a *FORCERENEW* message [28] to force C_1 into RENEW state. Then C_1 will try to send *DHCPREQUEST* message to renew its existing lease as in normal cases. If for some period of time τ_1 (a configurable parameter) C_1 still doesn't renew its lease, DHCP server will

inform the monitoring daemon of an invalid pair (MAC_{C_1}, IP_{C_1}) and packets from C_1 will be prohibited from passing through bridge B. If C_1 renews its lease at a later time, DHCP server S either allocates a new lease or renews the old one, and informs the monitoring daemon of such changes.

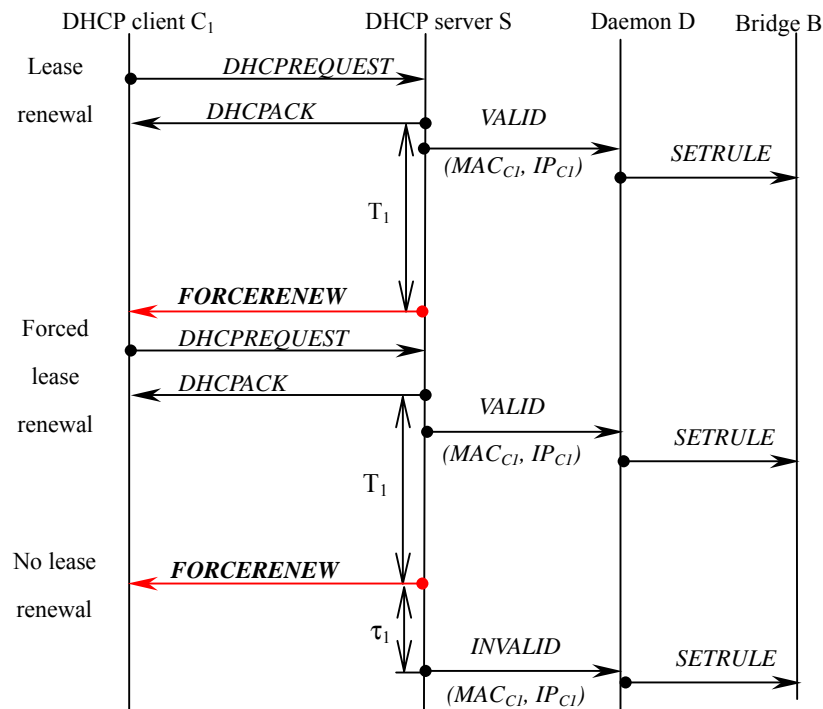


Fig. 7(b) shows the second case of client-server interactions where DHCP client C_1 renews its lease automatically in normal cases. If C_1 doesn't renew after lease expires, DHCP server S will send $FORCERENEW$ message to it. If for some period of time τ_1 , C_1 still doesn't renew its lease, (MAC_{C_1}, IP_{C_1}) will be marked as invalid pair.

Thirdly, when a non-DHCP host D_1 registers to monitoring daemon via some registration procedure or notifies to DHCP server S via $DHCPINFORM$ messages, monitoring daemon will inform the valid pair (MAC_{D_1}, IP_{D_1}) to bridge B. D_1 will then be able to connect through the bridge. The process is shown in Fig. 7(c).

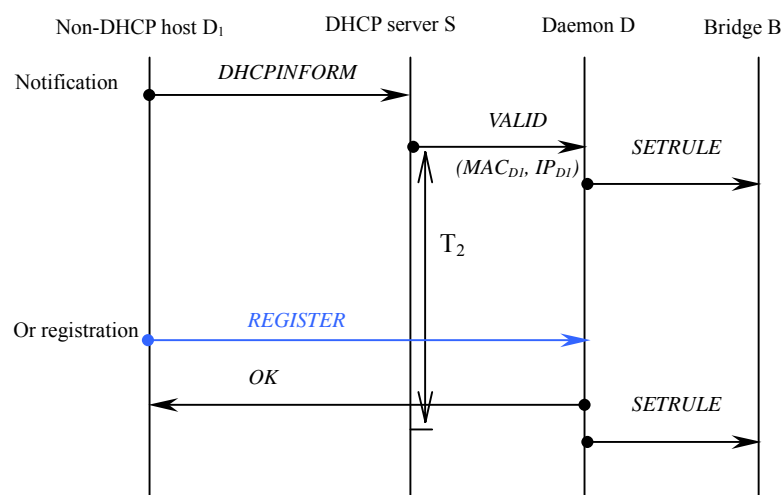


Fig. 7(c) shows the third case of client-server interactions where non-DHCP host D_1 notifies with *DHCPINFORM* message to DHCP server or registers via registration client to Daemon D.

Lastly, when a manually configured host N_1 makes its connection attempts as shown in Fig. 7(d).

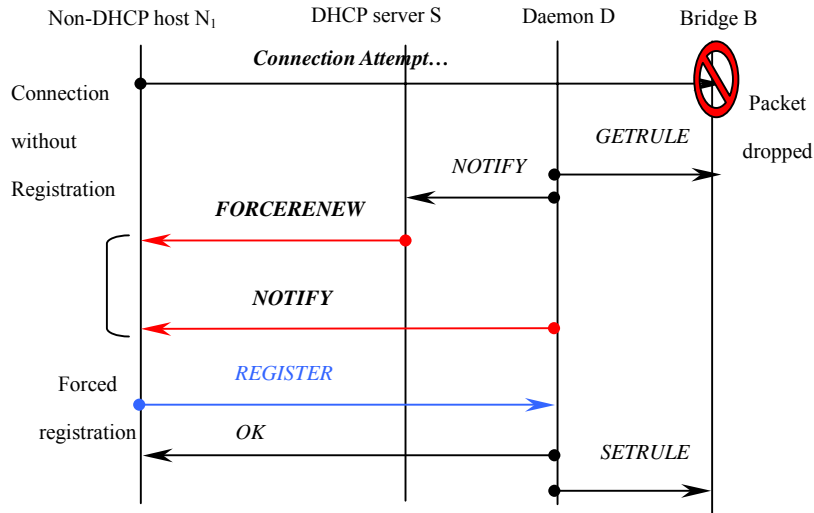


Fig. 7(d) shows the fourth case of client-server interactions where non-DHCP host N_1 attempts to connect without registration. N_1 will be denied of Internet access until registration is completed.

Since N_1 is not registered to monitoring daemon D, bridge B will by default drop its packets and mark (MAC_{N_1}, IP_{N_1}) as invalid. Daemon D will periodically poll from the system logs of bridge B and get the list of such illegal hosts. Then daemon D will either send *RHCPRENEW* messages to these illegal hosts one by one or notify DHCP server S, which in turn sends *FORCERENEW* messages. When N_1 receives such messages, it can either respond with registration requests to daemon D or it can send *DHCPINFORM* message to DHCP server S. If neither was done, after a period of time τ_1 (a configurable parameter), DHCP server will inform daemon D of an invalid pair (MAC_{N_1}, IP_{N_1}) and N_1 will be prohibited from passing through bridge B as in the second case above.

4.2.4 Deployment Issues

In a switched environment, our DHCP-based management infrastructure can be illustrated as in Fig. 8.

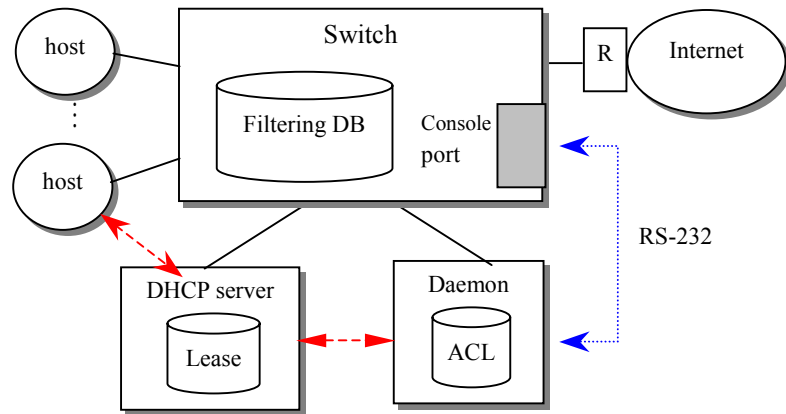


Fig. 8 shows the DHCP-based management infrastructure in a switched environment.

Monitoring daemon is configured to connect through two interfaces: an Ethernet link to contact with DHCP server and other hosts, and a RS-232 link to collect information from and enforce rules to the switch. Note that DHCP server could be standalone or integrated with monitoring daemon. If DHCP server is combined with monitoring daemon, some traffic can be reduced but the load would be higher. Slight overhead under such switched environment is inevitable unless the daemon/DHCP server modules could be hardwired into the switch.

For ordinary layer 2 switches, Filtering Database can be accessed in many ways, for example, through the web interface, Telnet, SNMP (Simple Network Management Protocol) [35], or via a console port dedicated for management purposes, as in the case of 3Com SuperStack II Switch 3300XM [36].

In the case of wireless bridges, access points are often hardware-based, which is difficult to configure dynamically according to our needs. Therefore, in our solution, a software AP is incorporated into the infrastructure on which we can build Filtering Database for regulating the traffic across it as shown in Fig. 9.

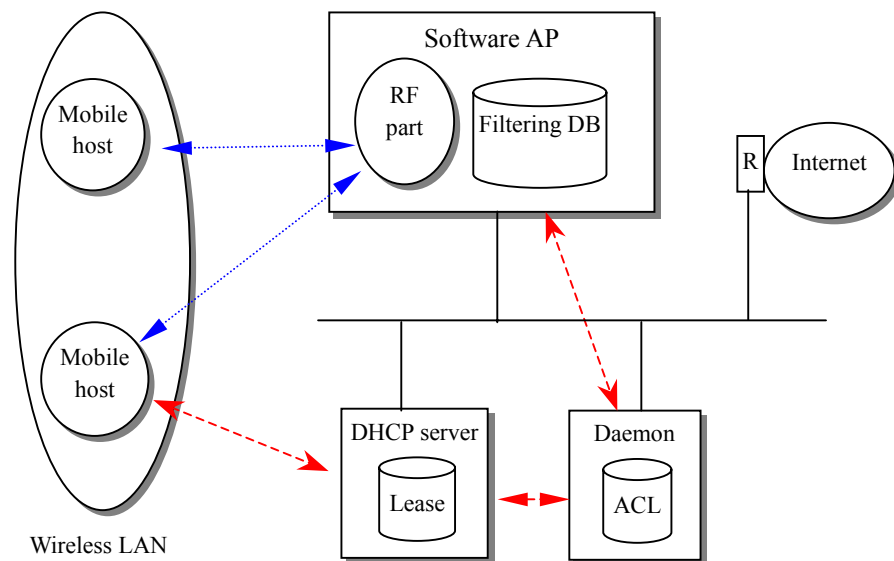


Fig. 9 shows the DHCP-based management infrastructure in a wireless environment.

However, there are some differences between these two infrastructures. Firstly, monitoring daemon needs not but would be better integrated into the software AP as a module. In the case of wired environment, a daemon module cannot be integrated into a hardware-based switch unless the switch is re-designed to do so. That's the reason why we incorporate a software-based AP instead of hardware-based one. Actually, we could also use normal hardware-based AP since under normal configurations it will eventually connect through switches somewhere in the switched environment. The advantage of software AP is its flexibility and access control at the very first point of attachment for mobile hosts.

Secondly, DHCP server will usually be on the Ethernet-side of the AP rather than the RF-side. That means DHCP requests from mobile hosts will pass through the software AP to DHCP server that incurs overhead for both wireless LAN and the Ethernet. If DHCP server is also integrated into the software AP, more traffic will be reduced on both wired and wireless networks.

4.2.5 Implementation Issues

1. Layer 2 vs. Layer 3 Switches

For layer 2 switches, only MAC addresses are inspected and added into packet filtering rules of Filtering Database. Such level of control is not tight enough in some cases as shown in the following IP-spoofing example.

In the first place, when hosts A and B with (MAC_A, IP_A) and (MAC_B, IP_B) respectively are trusted by our server, layer 2 switch will mark MAC_A and MAC_B as authorized. However, when trusted host A tries to send packets using the same IP address as trusted host B, layer 2 switch will not notice invalid packets from (MAC_A, IP_B) since the Filtering Database lacks layer 3 information when trying to keep track of invalid host connections. This will cause big problems since unauthorized hosts can gain access rights in this way.

On the other hand, with layer 3 switches, the problem can be solved since the Filtering Database could contain both layer 2 and layer 3 information, i.e. all valid (MAC, IP) pairs. In the above example, layer 3 switch will mark (MAC_A, IP_A) and (MAC_B, IP_B) as authorized pairs. When host A starts sending spoofed packets with (MAC_A, IP_B) , layer 3 switch will notice these spoofed packets and no access will be allowed from host A.

2. Integrated vs. Separated Modules

In our infrastructure, monitoring daemon and DHCP server are separated for illustration purpose only. In real implementation, we could have combined these two modules and experienced less overhead for inter-process communications. However, as individual functional modules, DHCP-related functions are better put together in a DHCP server module while communications between DHCP server and bridges in another separate monitoring module. That would be a cleaner design.

3. DHCP vs. RHCP options

In RFC 3203 [28], it's not clearly specified when and how to trigger DHCP *FORCERENEW*. In our infrastructure, it's triggered by illegal connection attempts of DHCP-unaware hosts. With the installation of appropriate DHCP/RHCP modules on them, notification can be done via DHCP *FORCERENEW* or RHCP messages.

4.2.6 Conclusion

As new network technologies and applications are being developed, intranet management plays a critical role in both wired and wireless networking environments. Even with the widespread deployment of DHCP mechanism, there would still be more problems if it couldn't be enforced among DHCP clients as well as manually configured hosts.

In this section, we proposed a management infrastructure that strengthens DHCP with MAC bridges such as Ethernet switches and wireless access points. We also showed some possible uses of new DHCP options like *DHCPINFORM* messages and DHCP reconfigure extension. The advantage of this combination of DHCP server and MAC bridge is two fold. Firstly, functionality of MAC bridge can be enhanced by address allocation flexibility. Secondly, DHCP mechanism can be strengthened by MAC bridge. If this management scheme is carried out over the whole intranet, both DHCP clients and DHCP-unaware hosts can be regulated under the same infrastructure. Local configuration conflicts can thus be reduced to the minimum, and a better networking environment can be expected.

4.3 Security Issues

Since security issues are the most important, in this section, we will focus on the security issues for both wired and wireless networks.

4.3.1 Authentication, Authorization, and Accounting (AAA)

For security issues in existing wired networks, AAA is among the most important. Authentication is the process of confirming the real identity of the one making requests. It's the first line of protection since misuse of the identity may lead to problems in following steps.

After authentication is done, Authorization continues to ensure the rights for requestor to access a specific resource. It is also known as Access Control since access rights to resource are classified and no access will be granted without suitable rights.

Finally, although the requestor is real and has the right to access the resource, we still have to keep track of any activity done to any resource with enough information to know and recover if necessary the details of any modification. Besides, resource consumption data is collected for the purpose of capacity and trend analysis, auditing, and billing. This is known as Accounting since the detailed information of users can be kept.

In our infrastructure, DHCP-based authentication and authorization are done in the

process of resource allocation (i.e. DHCP lease allocation or renewal). This has the benefits of access control at the first point as in the case of authentication in dialup services like RADIUS. Although IEEE 802.1x [37, 38] has been proposed as a port-based network access control mechanism for LAN, it's shown that security flaws are possible for the combination of 802.1x and 802.11 wireless LAN. In the next section, a comparison of IEEE 802.1x and DHCP-based mechanism [39] will be presented.

On the other hand, in our infrastructure, accounting is done when accessing each service. Detailed access logs for each operation will be recorded including date, time, user, and operation.

4.3.2 Comparison

The basic security mechanism provided in 802.11 is called WEP (Wired Equivalent Privacy), and has been proved by many researchers [15, 16, 17] to contain significant flaws in its design. Generally speaking, under normal traffic in wireless LAN, the key used to encrypt transmission data in an AP will duplicate itself in only a few hours. That would be easy for attackers to recover the original data packet without much effort. Some fixes for WEP encryption standard has been proposed, for example, fast-packet keying and WEP2, and working groups in IEEE are also trying improve the security of 802.11.

Therefore, IEEE 802.1x [37, 38] aims to provide port-based network access control when used in conjunction with 802.11 and other IEEE 802 media. Taking advantage of existing Extensible Authentication Protocol (EAP) [40], 802.1x-enabled access point can forward authentication requests from clients to the authentication server. Only after finishing authentication steps can a client gain network access.

However, two security flaws are found to be possible [41] in 802.1x and its combination with 802.11, namely, *session hijacking* and *man-in-the-middle attack*. The primary flaw in 802.1x design is the absence of mutual authentication mechanism for supplicant and authenticator. In this design, authenticator is assumed to be trusted which is not necessarily true. Malicious hosts can presume the role of access points and easily receive all packets from both parties of connections without being noticed since both sides believe that the other party it is communicating is the authentic partner.

1. IEEE 802.1x

IEEE 802.1X [37, 38] is now a standard for port-based network access control. It utilizes EAP [40] to provide authenticated network access for IEEE 802 media, including Ethernet, Token Ring, and 802.11 wireless LAN [8]. The EAP messages encapsulated in 802.1X frames are called EAPOL, or EAP over LAN.

There are three entities involved in 802.1X authentication: Supplicant, Authenticator, and Authentication Server. As shown in Fig. 10, Supplicant is the client being authenticated, while Authenticator is the entity requiring authentication, and the authentication takes place

in Authentication Server.

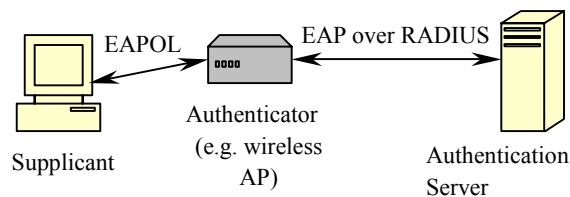


Fig. 10 shows the general topology of the three entities involved in IEEE 802.1X authentication.

For example, the principle of operation for IEEE 802.1X in a wireless LAN is depicted in Fig. 11.

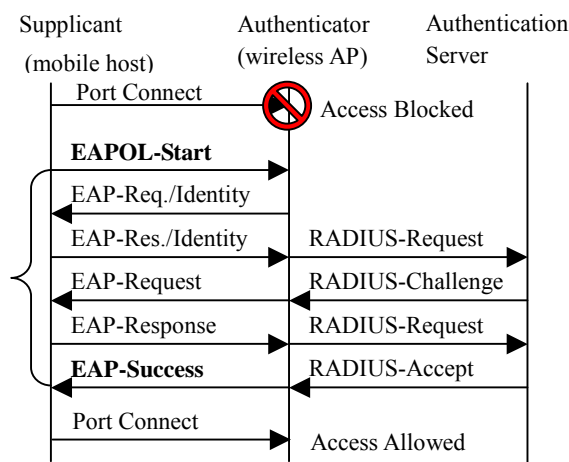


Fig. 11 shows the principle of operation for IEEE 802.1X authentication in a wireless LAN.

When a mobile host tries to connect through the nearest AP in a wireless LAN, the AP will open a port and forced it into un-authenticated state in which only 802.1x packets will be able to pass. Then the client starts the authentication request by sending an EAPOL-Start message, and the AP will request its identity and forward the responses to the Authentication Server. As an optional support in 802.1X, RADIUS [42, 43] is used between Authenticator and Authentication Server. After finishing the authentication work, Authentication Server will pass the result back to Authenticator, which will set the port state into authenticated state. Then the client will be able to connect through the AP.

2. DHCP-based vs. IEEE 802.1x

As compared to our DHCP-based approach in this paper, there are several differences between IEEE 802.1X and our infrastructure:

1. IEEE 802.1X explicitly requires authentication requests to be sent from clients, and an authentication server is necessary. In our approach, DHCP clients do not need explicit authentication since authentication and network access control are all done in the process of resource allocation. For DHCP-unaware hosts, a simple registration process is still needed, but the handling of registration requests is integrated in DHCP server, the

central management server in our infrastructure, thus eliminating extra overhead.

2. In normal network configurations, DHCP server may have been operating, but not authentication server. Little overhead will be incurred for the deployment of DHCP-based management. Besides our DHCP-based mechanism, we could have also adopted IEEE 802.1X and authentication server as an extra layer of control. Higher level of security could be achieved but much overhead would be needed for DHCP clients.
3. DHCP server costs less and it's simpler to integrate access control functionality. But one drawback is that accounting abilities may not be provided. AAA (Authentication, Authorization, and Accounting) functionalities are usually integrated in one server.
4. IEEE 802.1X is a port-based network access control scheme. In our approach, we extend the idea further to MAC layer user authentication and access control. Applying finer level of access control we can truly differentiate the real identity of intranet hosts, thus guarantee the authenticity. Finer access control leads to better local host management and conflict prevention.

4.3.3 Security Issues for Peer-to-Peer Architecture

For peer-to-peer architecture, special security concerns are necessary due to the direct communication characteristic among peer hosts.

1. Data Authenticity

In ordinary pure peer-to-peer systems, the emphasis is on the anonymity of data publishing and retrieval. However, quality of data will be unpredictable if the source of data is totally unknown. And the authenticity of data cannot be guaranteed. Therefore, our focus is on manageability instead of anonymity. The exact date, time, publisher information will be recorded.

2. Data Theft

Direct communication among peer hosts will result in more incoming connections for each peer. There will be more unauthorized access attempts. Therefore, access control will be enforced not only at resource allocation stage, but also at the data communication stage that takes place directly among peer hosts.

3. P2P Virus

In addition to ordinary contamination path of viruses, a new breed of virus is emerging. For example, MSN IM (Instant Messaging) virus, and GnuTella Mandragore virus are two examples. The operation starts when a malicious instant message is sent from one of your friends claiming the availability of an important web page. When the user connects to the claimed web page, it will utilize a vulnerability of Microsoft IE (Internet Explorer), and invade the host. Then the same malicious message will be propagated to the list of your

good friends and do the same thing as described above. Therefore, eventually, thousands of thousands of malicious messages will be flooding the whole IM channel, and the real important messages will not be able to deliver at all. Users will be forced to stop using IM.

One way to avoid this scenario is to authenticate and confirm every message that you send before delivering them. Also be sure to install personal firewalls to invalidate unauthorized incoming and outgoing connections.

4. Effects of Firewall/VPN/NAT on P2P Systems

Since the direct communication nature among peer hosts, hosts are facing more direct incoming threats. Although firewall could stop all incoming connections from the outside including p2p connections, there are still ways to get around this. For example, solutions have been come up like *rendezvous server* outside the firewall, or polling mechanisms.

The same problem takes place in the case of VPN-protected or NAT-translated private networks. Although protection from outside attacks is important, it's equally important to get peer connections to cross the firewall, VPN, or NAT. It's critical to strike the balance between the tradeoffs for security and connectivity.

4.4 DHCP-based MAC-Layer User Authentication and Access Control

Resolving local resource conflicts is often more critical than preventing outside attacks in a LAN. Since DHCP option is not required for all hosts, misconfigured or non-cooperating ones could jeopardize the whole LAN. Although firewall can filter out unwanted traffics, it cannot resolve conflicts within a LAN. As for SNMP, agent processes must be installed on hosts to be coordinated by network managers, which would incur overhead in an environment without prior management at all.

In this section, a MAC-layer user authentication mechanism is presented for blocking outside intrusions and reducing local configuration conflicts simultaneously without much overhead. By combining DHCP and firewall, local users are managed by their username-MAC-IP triples, and unauthenticated users will be limited in network resources by DHCP server and be denied Internet access rights by firewall. No extra agents are needed, and only a simple registration process needs to be done. Also, a web-based management interface for DHCP server, firewall, and user registration is incorporated for the ease of administration.

4.4.1 Introduction

Network management is becoming critically important with the explosive growth of Internet. Managing such limited resources as IP addresses is always a big problem since resources are better utilized if regulated properly. Several solutions have been suggested, for

example, for “IP-scarcity” problem, such as IPv6 (IP version 6) [44], IP sharing (or IP masquerading [45]), Private Address Space [46], ...etc. However, the cost of manually configuring hosts on all kinds of platforms would be extremely high since manual configuration is prone to errors. Misconfigured hosts may not be able to connect to the Internet and might even interfere with other hosts and probably the whole LAN. Therefore, DHCP (Dynamic Host Configuration Protocol) [22, 31] was suggested to manage as many kinds of network configuration information as possible, automatically and dynamically.

However, since DHCP is not mandatory for all hosts, non-participating fixed-IP hosts may still interfere with DHCP clients. System administrators have to make sure that the range of IP addresses of fixed-IP hosts is separate from those of DHCP clients. The LAN can even be divided into different subnets if necessary. Even so, there’s still a possible security leak within the LAN: the user authentication problem.

Firstly, someone may occasionally or deliberately “steal” a valid IP address from legal owners, as shown in Fig. 12. He may send out anonymous e-mails or do anything harmful to the Internet with this stolen IP. If we track down the transaction log (if there is any), this stolen IP would be recorded, but not the one who stole and used it. Since it is a valid IP, we cannot restrict its access rights if we cannot distinguish between the legitimate owner and the thief.

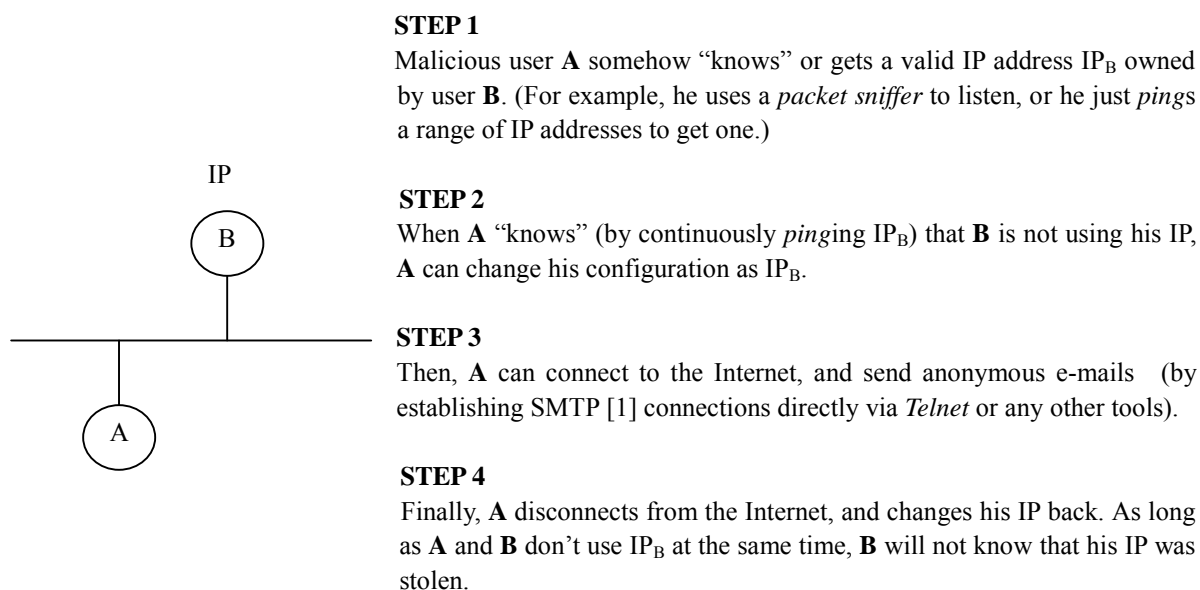


Fig. 12 shows how a malicious user can “steal” and use a valid IP address owned by legal users

Secondly, malicious DHCP clients may masquerade as legitimate ones in MAC (Medium Access Control) address level, retrieve configuration information, and claim all resources intended for legal clients. Therefore, we cannot manage network configurations safely by using DHCP alone. In other words, we need to incorporate a user authentication scheme that distinguishes between legitimate and illegal users in a LAN, and only legal users who have passed the user authentication will be granted the DHCP leases and Internet

access rights. Unauthenticated users will be restricted and even denied any network resource from DHCP servers.

4.4.2 Managing a LAN: Using DHCP and Firewalls

As illustrated in Fig. 13, a simple configuration of DHCP server [47] and firewall [45, 48, 49, 50] can be easily setup for resource allocation and isolation between the LAN and the Internet. Private Address Space schemes [46] can even be adopted in DHCP server for security concerns since private addresses cannot be reached from outside the LAN as explained in RFC 1918 [46].

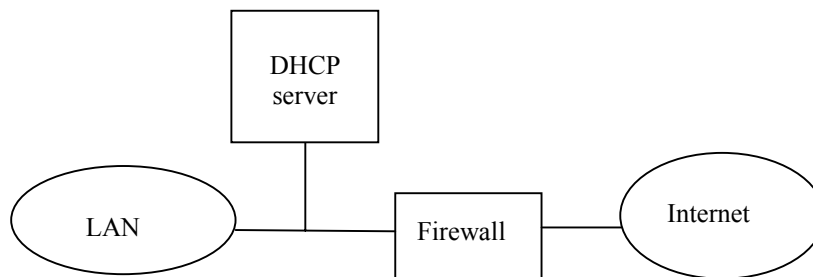


Fig. 13 shows a simple configuration of DHCP server and firewall.

However, the problem of non-participating fixed-IP hosts and unauthenticated users as mentioned above still remains unsolved since Private Address Space cannot avoid local malicious users. Those with valid stolen IP addresses can penetrate firewalls with no effort. Therefore we present an architecture which prevents local demeanor as well as outside attacks all at once as shown in Fig. 14.

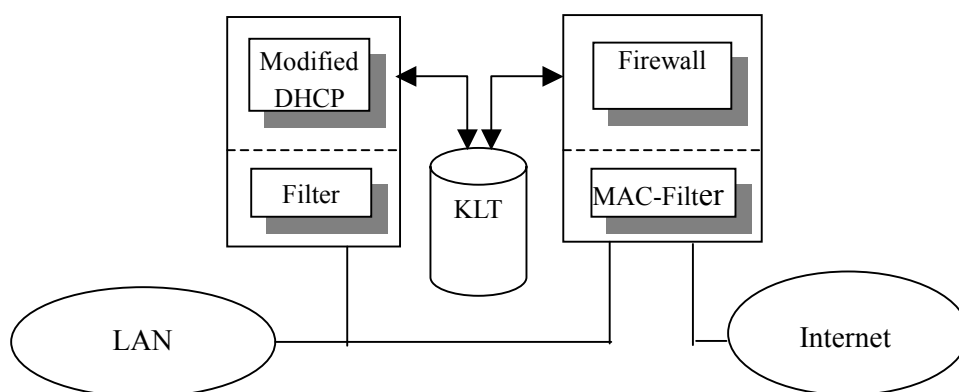


Fig. 14 shows the architecture and relationship of key components, where KLT is Kernel Lease Table.

As in normal firewalls, we could set up (IP-layer) packet filters [45, 48, 49, 50] which drop unwanted traffics from unregistered clients. Besides, a MAC-layer filter is needed as shown in Fig. 14 to discard illegal packets at MAC-layer. After checking the validity of

source MAC address in packets in the first set of filter (at the MAC-layer of firewall), those packets from legal source MAC addresses are then passed on to the second set of (IP-layer) filters which do appropriate verification as specified in the filtering policies of firewall.

In order to facilitate such MAC-layer checking in firewall, we have to uniquely identify network hosts as well as individual user since each one may have different access rights on different hosts. Therefore, not only personal information of all legal users in a LAN but also IP and MAC addresses of all legitimate hosts are collected in the process of user registration.

4.4.3 Host Identification – MAC Address Authenticity

In DHCP message format [22, 31], ‘*chaddr*’ field contains the client MAC address while the ‘*Client Identifier*’ option could be a hardware address, a DNS name, or any other type of unique identifier. Ordinary (IP-layer) DHCP servers keep track of DHCP leases by either value or both as long as it is unique, but these two values need not be the same. Even a MAC address is filled in the ‘*Client Identifier*’ option, we still can’t make sure that it’s the real address if DHCP client deliberately cheats.

To ensure identification of individual network hosts, not only ‘*Client Identifier*’ option and ‘*chaddr*’ field, but also the authentic MAC addresses in the Ethernet header [51, 52, 53] of DHCP packets are recorded in our modified DHCP server. A mechanism for carrying up Ethernet addresses to upper layers in the TCP/IP protocol [52, 53] engine was designed to facilitate recording of real MAC address in Ethernet headers. All DHCP packets with invalid ‘*chaddr*’ or ‘*Client Identifier*’ field will be discarded at MAC layer. Since most DHCP clients are tightly coupled with the operating system, for example Microsoft Windows 9x, they cannot be modified easily. So our DHCP extensions focused on the server side, and DHCP clients need no modification.

4.4.4 User Management – the Operation of DHCP with User Registration

As Fig. 15 shows, the modified DHCP server starts up as normal ones, and DHCP clients may make *DHCPDISCOVER* and *DHCPREQUEST* requests [22, 31] (at time τ_1) for allocating a new lease. But in our design, only a short lease T_1 will be granted at first if the client has not registered yet.

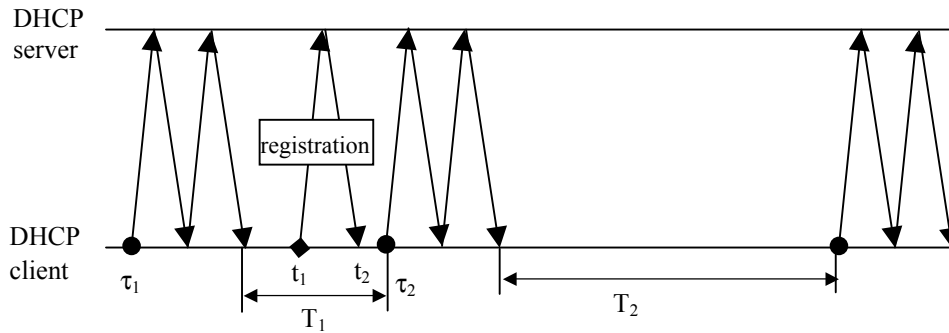


Fig. 15 shows the timeline diagram of DHCP operations and user registration. The durations of DHCP leases before and after a successful user registration are also illustrated. At the DHCP client side, four time points and two periods of time are depicted as follows:

- Time τ_1 : to allocate a new lease
- Time τ_2 : to extend an existing lease
- Time t_1 and t_2 : user registration starts and completes
- T_1 : a short lease at first
- T_2 : a longer lease after user registration completes.

After allocating a short lease T_1 , users have to register with a special registration client with which one can login the modified DHCP server (at time t_1) and type in username and password which will be encrypted and verified at DHCP servers. Once this simple registration process is complete (at time t_2), usernames, IP and MAC addresses in the Ethernet headers of DHCP packets are simultaneously recorded in the kernel lease table that constitutes user authentication information for later filtering controls.

When the short lease T_1 is about to expire (before time τ_2), DHCP client will send *DHCPREQUEST* message for extending existing lease. At this time, the modified DHCP server will check if the client has been registered. If so, a longer lease T_2 will be offered, and the client will get full Internet access rights until the lease expires. If it has not registered, short leases may still be offered, but the client will be limited from going out of firewall, and even prohibited from allocating any network resource unless it reboots, makes DHCP requests again, and successfully completes the registration process. For security reasons, users are required to re-register when new users come, and when a user connects via new adapters (hence new MAC address) or different hosts (thus different IP). For normal users who seldom change their working host or network adapters, no further registration will be needed once they have registered.

4.4.5 User Management – the Kernel Lease Table (KLT)

As illustrated in Table 1, KLT (kernel lease table) contains critical user authentication information for MAC-level control, which should not be exposed to casual accesses. Only DHCP servers and firewalls are allowed to modify and manipulate it when DHCP servers check for user authentication, and when firewalls need to update their MAC-layer filtering

rules.

Table 1 shows the format and an example of the kernel lease table.

Username	IP Address	MAC Address	Expiration Time	Expired	Registered
Vicky	192.168.2.2	00:01:C8:0A:00:0B	3600	No	Yes
Chris	192.168.2.3	00:01:C8:0C:FE:0E	7200	Yes	Yes
John	192.168.2.4	00:01:C0:0B:EB:30	300	No	No

Therefore, a proprietary protocol similar to FTP (File Transfer Protocol) [54, 55], with username and password encryption, is used to access KLT. For a more user-friendly management environment, a Web-based management interface is also incorporated, as shown in Fig. 16. In addition to KLT, we can also configure parameters on DHCP servers and firewalls with this same Web interface with which only superuser authorization is allowed.

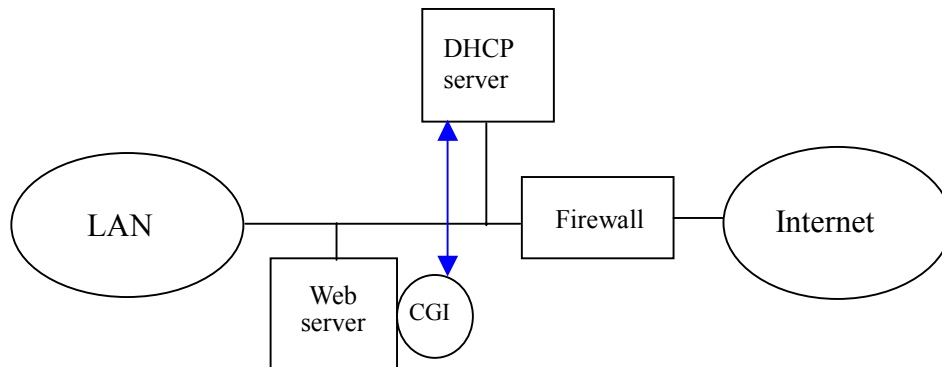


Fig. 16 shows the Web-based management of DHCP servers and firewalls.

4.4.6 Discussion

There are several design alternatives for the arrangement of DHCP servers and firewall. For example, as shown in Fig. 17, DHCP server and firewall can be on the same physical machine, as have been implemented and well-tested in our experiments.

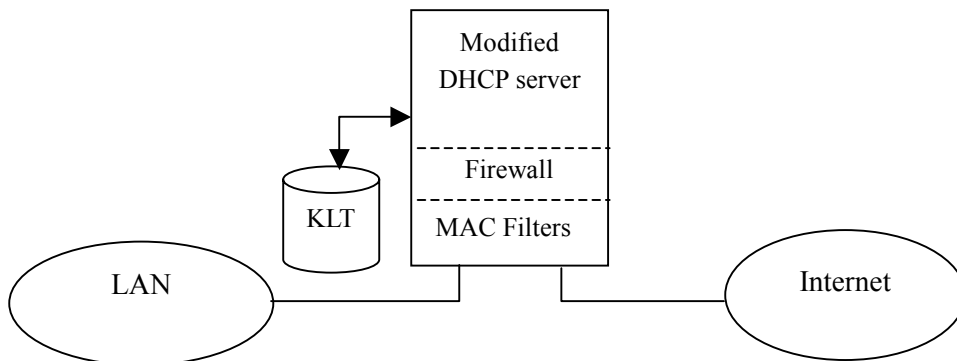


Fig. 17 shows an implementation with DHCP server and firewall combined which has been well tested.

There are several advantages of combining DHCP server and firewall on the same

machine. Firstly, since KLT is shared by DHCP server and firewall, it will be easily available to both if they are on the same host. Besides, firewalls should get the latest user authentication information and lease updates as soon as possible, or unwanted packet loss and packet slip-through may result. Most importantly, the packet exchanges for KLT between DHCP server and firewall will be removed thus reducing the network traffic load and avoiding eavesdropping on critical user authentication information at the same time.

However, one possible disadvantage for this approach is: packet filtering rate of firewall will drop for the processing of DHCP messages and user registration packets. It is a trade-off between speed and security. In fact, as the result of our one-year experiments with this implementation shows, in which a firewall is built on a Pentium-90 PC with 16 MB of RAM in a small LAN consisting of no more than 10 hosts, the firewall response time is not greatly influenced and no user inconvenience is reported during a long period of time.

As compared to current network management protocols, such as SNMP (Simple Network Management Protocol) [35], there are several advantages in our design. Firstly, agent processes are not needed on each host to be managed which means no overhead incurred on the client side except for a one-time registration process. Only a registration client is needed on each platform (which has been done in our work). In other words, our scheme is platform-independent. Secondly, we can monitor and control the access rights of LAN users simply by managing the combined DHCP server and firewall. No other manager applications will be needed and the well-known Web interface makes administration easier. We could have used the same Web interface for user registration as well as for server management, but the security would be greatly reduced when host and user authentication information need to be passed through the web server and CGI program.

There are other ways to extend DHCP for better user authentication. For example, we could have modified DHCP clients to encapsulate user authentication information in new options or fields in DHCP messages. However, as we mentioned earlier, DHCP clients may not be easy to modify, especially if they are tightly coupled in the operating system. Besides, not only DHCP server and all existing DHCP clients, but also DHCP protocol itself would have to be modified to handle user authentication information in DHCP messages. That would be a lot of changes and cost. So we decided to design a separate special registration client, and only the slightest changes need to be made.

4.4.7 Conclusion

More often than not, managing local users is more critical and trickier than preventing outside invasions in a LAN. In this section, we explained the need to verify the real identity of LAN users and network hosts – in MAC level. A method of managing user authentication information was presented with interactions between modified DHCP server and firewall. We can dynamically configure network hosts while limiting the access rights of

unauthenticated users, thus making the most of the power of DHCP and firewall. We can keep track of the allocations of network resources and Internet service accesses. In the future, we may also be able to detect and notify all misconfigurations automatically. A web-based integrated management interface for monitoring and modifying network configurations and Internet access rights of all users of the whole LAN may even be adopted. Of course, this requires superuser authentication and more careful planning of security control.

4.5 Enhancements for User Authentication and Access Control

Intranet management is becoming as critical as intrusion detection since misconfiguration of local network parameters may jeopardize the whole intranet. In our previous paper [34], a MAC (Medium Access Control) layer user authentication scheme combining DHCP (Dynamic Host configuration Protocol) [22] server and packet filtering firewalls was proposed [56] for better control of local resources and Internet accesses.

In this section, DHCP mechanism will be further strengthened by the newly proposed DHCP options *DHCPINFORM* [22] and DHCP *Reconfigure Extension* [28]. An access control list (ACL) was maintained at the firewall according to the states of DHCP leases and packets from intranet hosts. Hosts not obeying our DHCP-based management scheme will be restricted Internet access by firewall.

4.5.1 Introduction

As the diversity and complexity of intranet environment grow, local resource and configuration management has become as critical as intrusion detection. In order to reduce the tedious and error-prone manual configuration process, DHCP (Dynamic Host Configuration Protocol) [22] has gained increasing popularity because of its dynamic and automatic allocation capability. However, some inherent problems in DHCP operations may introduce more trouble. First of all, DHCP clients play active roles in allocating, renewing, and releasing the leases, while DHCP server cannot force DHCP clients to do so. If malicious hosts keep allocating new leases without releasing old ones, precious network resources could soon be exhausted and the whole intranet will be jeopardized. Secondly, DHCP mechanism is not mandatory. Manually configured hosts would conflict with DHCP clients if not properly configured. Therefore we need a way to enforce DHCP-based management policy for both DHCP clients and externally configured hosts.

4.5.2 Motivation

In our previous paper [34], we focused on the MAC (Medium Access Control) layer user authentication scheme and the regulation of DHCP clients at firewall. DHCP server distributes resources and keeps track of MAC addresses in addition to normal lease

information such as IP addresses. These MAC addresses are then passed into MAC-layer filters in firewall for controlling network traffic from the intranet. On the other hand, user authentication is carried out in a registration process. The user is first allocated a shorter lease just for registration purpose. After user information is collected and confirmed, one can really acquire a longer lease for Internet access. Registration is required for each new user or network interface card (NIC) in the intranet.

Recently, new DHCP options such as *DHCPINFORM* and *DHCP Reconfigure Extension* [28] have been proposed to address the above-mentioned drawbacks of DHCP operations. However, for these options to be useful, support from server and the infrastructure are required. In this paper, these new options will be integrated into our DHCP-firewall infrastructure and both DHCP mechanism and our management scheme can be further strengthened.

4.5.3 The Infrastructure

As shown in Fig. 18, the original infrastructure of DHCP-firewall combination is illustrated.

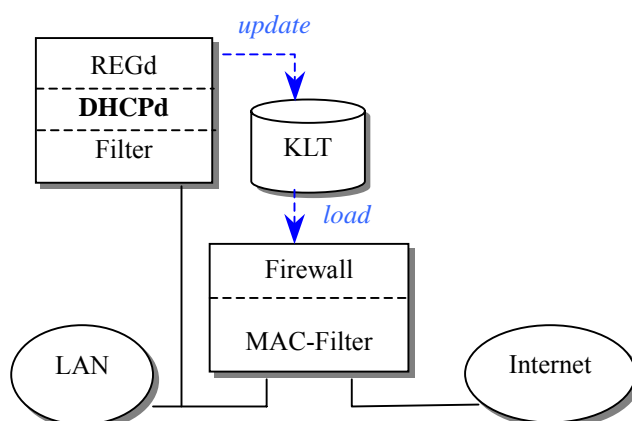


Fig. 18 shows the original infrastructure and key components in DHCP-firewall combination, where KLT is Kernel Lease Table.

The operations of the original DHCP-firewall combination are as follows:

- (1) For each local DHCP client, DHCP server first allocates a private IP address [46], for example, 192.168.1.2, which is only used for registration requests. The lease duration will be short at first.
- (2) During this short period of time, DHCP client is required to register to our registration server (REGd).
- (3) The (MAC, IP) pairs of all registered clients will be updated into the Kernel Lease Table (KLT) which will be enforced by MAC-layer filter in firewall.
- (4) If a specific host doesn't renew its expired lease, the (MAC, IP) pair will be marked as invalid in KLT unless re-registration is done.
- (5) New users or users with new network adapters will have to re-register in order to

update their MAC-layer authentication information.

Note that, in Fig. 18, KLT can only be updated by DHCP server when new leases or authentication information are available. The entries in KLT are only readable from MAC filters in firewall. In other words, the flow of lease information is one-way.

As new options such as *DHCPINFORM* and *FORCERENEW* are integrated into our infrastructure as illustrated in Fig. 19, the operations can be modified as follows where steps with asterisks in front are different from the original one.

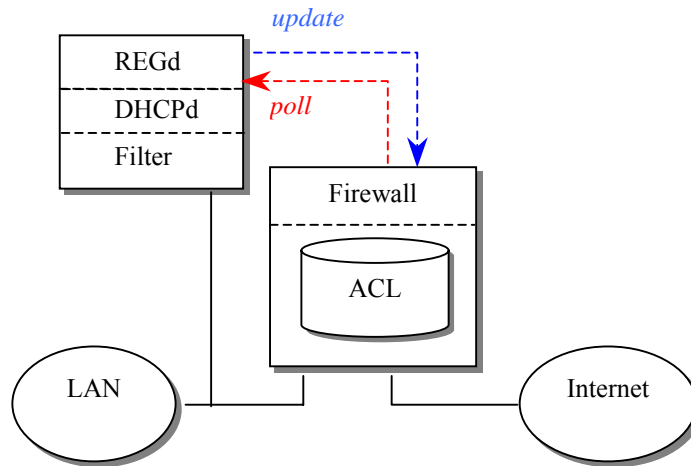


Fig. 19 shows the new infrastructure and components, where ACL is the access control list which may be updated by both DHCP server and Firewall.

- (1) For each local DHCP client, DHCP server first allocates a private IP address which is only used for registration requests. The lease duration will be short at first.
 - **(1b)* For externally configured hosts, *DHCPINFORM* messages could be used to notify DHCP server of their presence. However, MAC layer information could also be automatically collected by firewall. Registration also applies to such hosts if further user authentication is needed.
- (2) During this short period of time, DHCP client is able to register to our registration server, but this is not mandatory as in the case of externally configured hosts. However, if user authentication is needed, registration is strongly encouraged.
- (3) The *(MAC, IP)* pairs of all registered clients will be updated into the Kernel Lease Table (KLT) which will be enforced by MAC-layer Firewall.
 - **(3b)* Before the leases expire, DHCP server will send *FORCERENEW* messages to the corresponding DHCP clients one by one.
- (4) If a specific host doesn't renew its expired lease or when *FORCERENEW* messages are received, the *(MAC, IP)* pair will be marked as invalid in KLT unless re-registration is done.
- (5) New users or users with new network adapters will have to re-register in order to update their MAC-layer authentication information.

Resource allocation is maintained by DHCP server while MAC-layer access control is enforced by firewall. With the integration of these two functions, infrastructure for DHCP-based management can be strengthened. DHCP clients as well as manually configured fixed-IP hosts can be managed in our new infrastructure.

4.5.4 Data Structures and Operations: ACL

As shown in Fig. 2, ACL (access control list) is the central data structure in DHCP-Firewall interactions. It serves as the main filtering database for firewall to regulate network traffic at MAC layer. Although IP-layer packet filtering tools, such as *ipchains* [57] or *iptables* [58] for Linux kernel 2.4, are available, MAC-layer packet-filters are still in lack.

As opposed to the original infrastructure, ACL updates could happen in two cases: updates by DHCPd/REGd when lease or authentication information changes, and by firewall when access violation occurs which is illustrated in the following example.

For example, when an externally configured host *c* tries to connect to the Internet without registering or informing the DHCP server about its fixed IP address. Since all valid (*MAC*, *IP*) pairs of registered hosts and DHCP clients are already kept in ACL, firewall can immediately identify the packets originating from the unregistered hosts *c* and drop them out. In such condition of access violation, firewall will mark its (*MAC*, *IP*) pair as invalid in ACL. DHCP server can periodically poll from ACL and issue *FORCERENEW* messages in order to notify violated hosts to abide by out management policy. Therefore, ACL is read by firewall when filtering packets and by DHCP server when polling the new list of violated hosts.

4.5.5 Design and Implementation Issues

First of all, the location of ACL can be versatile if software-based packet-filtering firewall is used. We can put it in DHCP server or in firewall. It's too difficult (if not impossible) for hardware-based firewall to read data from ACL on a separate machine. In our implementation, DHCP server is on the same machine as a Linux-based firewall.

Since Linux kernel version is evolving so fast, we have to apply our MAC-layer filtering patch to all versions of Linux kernels. In order to reduce the difficulty in maintaining the patches, we use a modular design in our infrastructure. The main components in our infrastructure include: MAC-layer ACL management module, system setting scripts for firewall, and user-mode programs: DHCPd and REGd.

MAC-layer ACL management module is the kernel-dependent part that may change with the design of different versions of Linux kernels. In our previous implementations, we created a new system call [59, 60] for the communication between user-level programs and kernel-level MAC ACL. However, as many new system calls are "officially" being added

into new versions of Linux kernels, we have to adjust the system call number for compatibility reasons.

Therefore, in our latest implementation, we create a character device and implement appropriate *ioctl()* functions [61, 62] in kernel modules. Once *ioctl()* handling routines are added, no modification has to be made except for including our *ioctl*-related header files in user-level programs. *Ioctl* is more flexible than adding new system calls. And it's cleaner in design.

4.5.6 Discussion

In fact, user registration is not required for the correct operation of our new infrastructure, and externally configured hosts as well as DHCP clients can be managed in the same infrastructure.

In our management scheme, MAC-layer access control, user authentication, and resource management can be achieved with only a minor changes to the DHCP server and firewall. Although DHCP server and firewall have to be modified in order to enforce DHCP management policy and to handle DHCP server-firewall interactions, this could be easily done by installing the corresponding DHCP and firewall modules in the original Linux server since our design is modular.

Adding MAC-layer support in DHCP server and firewall may have some impact on their servicing and filtering performances. However, by dropping illegal packets the unnecessary network traffic can be reduced to its minimum. The performance degradation compared to the reduced network traffic is relatively small. Local conflicts in intranet can thus be reduced to its minimum.

4.5.7 Conclusion

In this section, we proposed a DHCP-based management scheme for MAC-layer user authentication and access control. DHCP clients as well as manually configured hosts can be gracefully controlled in the same infrastructure.

In addition, we proposed a possible application of new options DHCP *Reconfigure Extension* and *DHCPINFORM* messages. Not only can we strengthen DHCP operations, but we can also integrate the resource allocation capabilities of DHCP with the access control functionality of packet filtering firewall. With the coupling of these functions, DHCP mechanism can be strengthened and local configuration conflicts can be greatly reduced.

Chapter 5 Applications of Location Service

In this chapter, several common Internet applications will be reviewed and the design of peer-to-peer support for each application will be discussed. Finally, experimental results supporting the main contribution of this dissertation will also be illustrated and discussed.

5.1 Peer-to-Peer Mail Transfer Mechanism

In current Internet mail transfer mechanism, mail servers usually do a lot of extra mail processing like mail filtering. Junk mails as well as important messages are all stored on mail server, retrieved and then deleted by end users. It's a waste of processing time, storage space, and precious network resources.

In this section, peer-to-peer technology was included in ordinary mail transfer mechanism to reduce unnecessary overhead. Through the user location service, mail server can dynamically query the online status and current location of users. By redirecting mails to online mail clients, work load for mail servers will be greatly reduced, and personalization of mail processing configuration can be better supported.

5.1.1 Introduction

With the tremendous growth of the Internet, various networking applications such as WWW (World-Wide Web), E-mail, and FTP (File Transfer Protocol), have been widely used. Specifically, e-mail applications have become indispensable and critical to many people's daily lives. All kinds of information like important messages, notifications, and advertisements, to name just a few examples, arrive at your mailbox by e-mail without classification. However, in current mail transfer mechanism, mail server plays a crucial role since every step in mail delivery requires the intervention of mail servers. The protocol used in mail delivery, Simple Mail Transfer Protocol (SMTP) [1], is store-and-forward in nature where guarantee of mail delivery is the main concern. Therefore, mails in the process of delivery will have to be queued in every intermediate mail transfer agent (MTA) before arriving at the destination mail server. This takes too much overhead in network bandwidth and processing time.

When a user gets online, a mail user agent (MUA), such as Outlook Express or Netscape Mail, can be used to check if there are new e-mails or not. Mails are retrieved from a mail server via POP3 (Post Office Protocol version 3) [2]. However, both junk mails and important messages arrive at the same mail server regardless of their priority. Users can only retrieve each mail in order, filter them automatically or manually, and then delete unwanted e-mails. Another protocol called IMAP4 (Internet Message Access Protocol version 4) [3] provides more advanced mail management functions, for example, one can

get mail headers first before deciding which e-mails to get. Mail synchronization problems can also be easily resolved. However, not all mail servers implement IMAP4 functions. Most users still have to use POP3 clients for mail retrieving.

On the other hand, peer-to-peer technology has been widely deployed in various applications, for example, file sharing applications like Napster and ezPeer, instant messaging software like ICQ and MSN Messenger, and open source protocols like Jabber [63] and GnuTella [5]. Moreover, decentralized systems evolve towards centralization if scalability is concerned, and centralized applications evolve towards decentralization [64]. We have to find out what combination of centralization and decentralization works best for current Internet applications. Therefore, an infrastructure for integrating current Internet mail transfer mechanism and peer-to-peer instant messaging was proposed to provide better services.

5.1.2 Motivation

We will quickly review the current architecture for mail transfer and its shortcomings, and then propose our infrastructure as a feasible solution.

In current Internet mail transfer mechanism, separate protocols are used: SMTP [1] for mail delivery and POP3 [2] or IMAP4 [3] for mail retrieval and management. As shown in Fig. 20, a typical scenario for current mail transfer mechanism is illustrated.

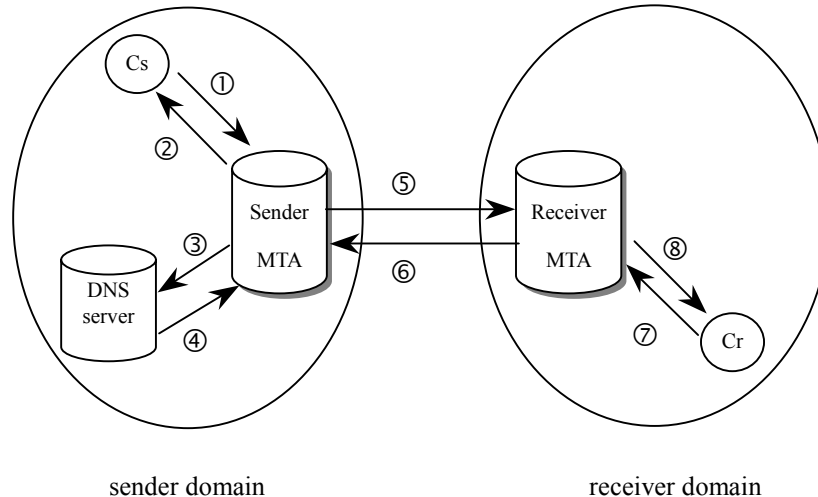


Fig. 20 shows a typical scenario for mail transfer and retrieval where

- Cs: mail sender
- Cr: mail recipient,
- Sender domain: home domain for sender,
- Receiver domain : home domain for receiver,
- Sender MTA: MTA for sender domain,
- Receiver MTA: MTA for receiver domain,
- User-side SMTPd: SMTP daemon on the user side.

As shown in Fig. 20, a user composes his e-mail by a MUA like Outlook Express or

Netscape Mail and then sends it to the MTA in his domain (or the ISP he connects to), or sender MTA, via SMTP (step 1, 2). Sender MTA then checks the recipient e-mail address for its validity and determines how to relay this mail to the right person. Usually the MX Record of the DNS (Domain Name System) [11] server will be queried for the mail exchanger of the domain specified in the recipient e-mail address (step 3, 4). After a relayed mail is received (step 5, 6), receiver MTA will store it into the recipient mailbox. When the recipient gets online, he can use a MUA to check and retrieve his own emails from the mail server via POP3 or IMAP4 (step 7, 8). No matter if the recipient is online or not, e-mails will be delivered to his domain mail server first. Then after the user decides to check his e-mails, he will connect to the mail server and fetch them back. Mail servers cannot notify users of incoming mails unless their MUA is configured to periodically check for new e-mails.

This process works fine, but there are several drawbacks that affect the performance of mail servers. Firstly, the load on a mail server is heavy in terms of storage for mailboxes and processing time for SMTP/POP3/IMAP4. Each mail server has to deal with every e-mail destined for domain users no matter if they are currently online or not. This could be a waste of server storage and processing time since e-mails are unnecessarily stored in server and then retrieved by on-line users. As the number of users and e-mails grow, the storage requirement of mails in receiver MTA will become larger and larger.

Secondly, personalization cannot be done very efficiently in mail server. For example, it's common to configure an anti-spam list on receiver MTA for the whole domain. But for each individual domain user, different configurations may be needed. We need a finer-grain control of such configuration for each individual user, for example, a separate anti-spam list for each user, which is more reasonable since each user may want to filter mails from different senders and hosts. Although it's possible to configure external programs for mail processing, for example: *procmil* [65] or *Milter API* [66] in *sendmail* [67, 68] version 8.10 or above. But it's still time-consuming for mail server to deal with personal configurations for all domain users.

In current implementations junk mails are removed as soon as possible after a user checks and retrieves his e-mails from server. That would be a waste of time and space for storing these unwanted junk mails in server followed by deleting them anyway.

In order to offload mail server and to provide complete customization in mail processing, a peer-to-peer infrastructure for mail transfer was proposed. Specifically, we want to bypass the mail server if the recipient is currently online. Users can customize their personal configurations for all kinds of mail processing.

5.1.3 Infrastructure

As shown in Fig. 21, an infrastructure for peer-to-peer mail transfer is illustrated.

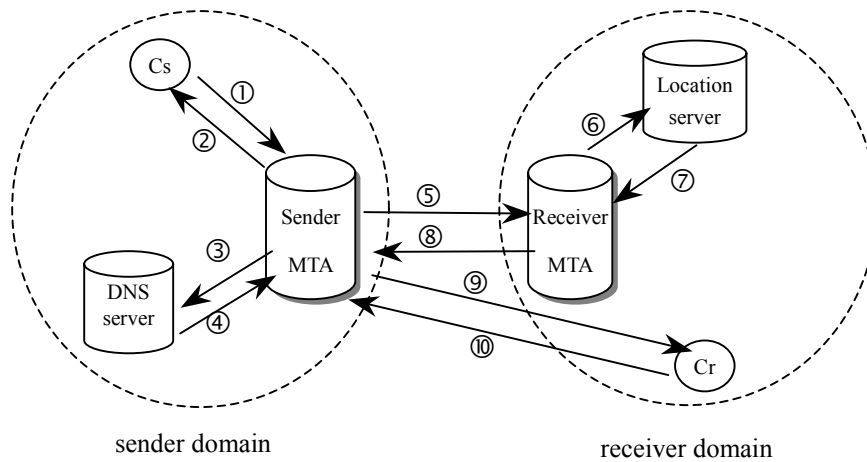


Fig. 21 shows our infrastructure for peer-to-peer mail transfer.

As shown in Fig. 21, key components in the infrastructure include: location servers, DNS servers, mail servers (sender and receiver MTAs), and mail clients (sender and receiver MUAs). The functional description of each component is provided as follows.

1. Location Server

Location server is responsible for storing the current location and way of contact for each domain user. User Location Record (ULR) includes user name, e-mail address, current online status, current IP address, device capabilities (audio/video support), and user profiles like access control list (ACL) for user resources. Since the amount of data may be quite large, a distributed scheme may be used, for example, one location server for each domain (like DNS server) may be a feasible way. ULR of each user is stored in the location server of his own domain as specified in the e-mail address.

Most of the related works in location service are about geographical positioning of mobile nodes in a wireless network, the location of servers, and location-based services. They mainly focused on the physical positioning of mobile nodes or server, not the current way of contact for clients and users.

As shown in Fig. 22, there are two possible operations for a location server: update and query. Clients update the ULRs to location server when users login, logout, change their location, or modify their configurations. On the other hand, mail servers query the location server for ULR of a particular user in order to directly deliver e-mails to him. In other words, location server has to be coupled with the management of user sign-on and sign-off. Users must do registration/de-registration when sign-on/sign-off.

However, when mobile user is roaming into a foreign network, he must register to his home location server for location update. This can be done directly or through the help of

foreign location server (*Indirect Update*). For a mobile node to detect its movement into a foreign network, the mechanism of Agent Advertisement/Solicitation in Mobile IP [9] can be deployed.

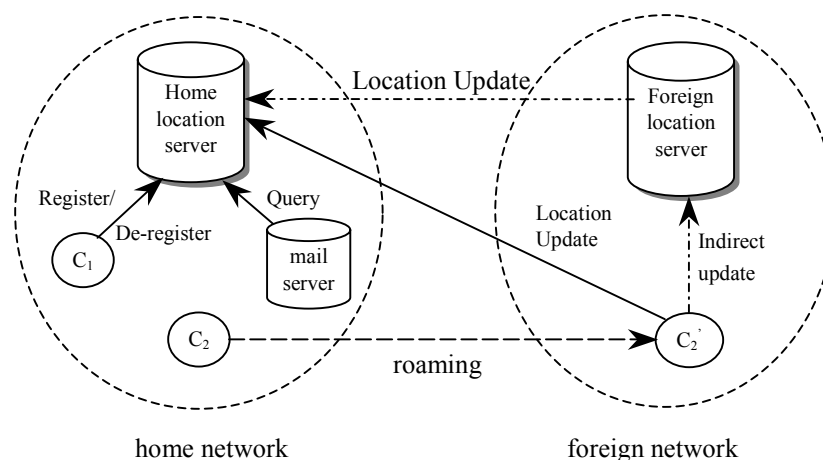


Fig. 22 shows the operations of location servers.

2. Mail Transport Agent (MTA)

As shown in Fig. 21, after sender forwards his mail to his mail server (sender MTA) (step 1, 2), sender MTA will query the MX Record of DNS server for mail exchanger (step 3, 4) in order to know which mail server to redirect to. When connected from sender MTA (step 5), receiver MTA will not receive the e-mail directly. Instead, location server is queried for the online status of recipient (step 6). If he is not on-line, mail gets delivered in its normal way, and saved in recipient mailbox.

On the other hand, if recipient is currently online (step 7), receiver MTA will reply to sender MTA a SMTP *REDIRECT* message (SMTP return code 551 [1]) (step 8), and sender MTA will send mails directly to recipient host. Then user-side SMTPd will check the validity of destination e-mail address and start receiving his e-mail (step 9, 10). After reception of e-mails, MUAs will be popped up for recipient to read e-mails.

One possible error may occur under the circumstances when the recipient host cannot be reached by sender MTA. Possible reasons could be abnormal broken connection or power failure that may not be immediately reflected in location server. In such cases, sender MTA will queue this mail delivery request in its local waiting queue as usual. After a configurable timeout, sender MTA will retry transmission to the normal mail exchanger (receiver MTA) queried from MX record of DNS server, not directly to user-side SMTPd since user location may have been updated again.

There is one design issue for location server. The user on-line status stored in a location server could be inconsistent with his exact location due to possible reasons as power failure or broken connection. We didn't focus on maintaining the timeliness of location information in a location server. Alternatively, applications making use of location

information are the ones that try their best to query location server just before connections are to be established to the user host. The reason for such design is user host mobility can be checked only when needed since it may often change.

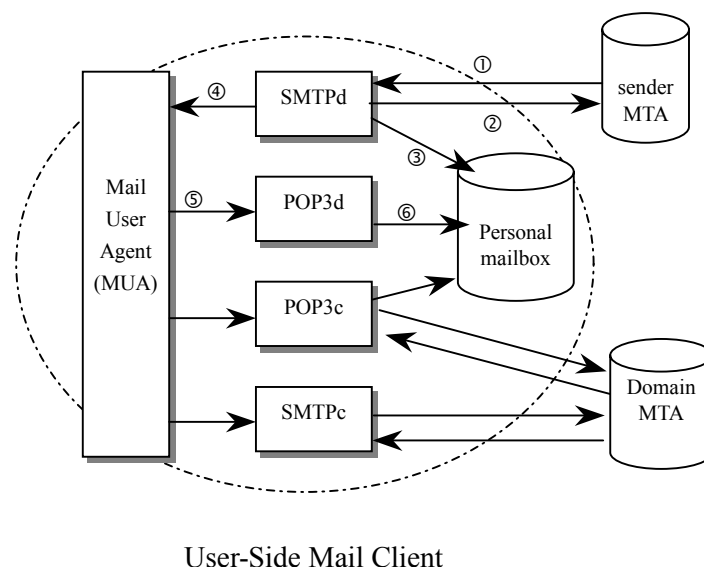


Fig. 23 shows the operations of a user-side mail client.

One design alternative is to modify mail servers to query ULR from a location server, instead of querying DNS server for MX record. But every application must be modified. Therefore, another alternative is to modify DNS server for handling user location queries. ULRs can be stored in a DNS server, and DNS protocol has to be modified since the input of ULR queries is user e-mail address, not hostname. Thus every application using DNS service will be able to utilize the user location service.

3. Mail User Agent (MUA)

As shown above in Fig. 23, SMTPd and POP3d (POP3 daemon) are needed in addition to the original SMTPc (SMTP client) and POP3c (POP3 client) in order to receive e-mails from sender MTA and to automatically pop up MUA for notifying recipient. Specifically, when user logs in, SMTPd and POP3d must be first initiated waiting for incoming connections from sender MTA. After mails are received (step 1, 2) and saved in a personal mailbox (step 3), MUA will be triggered (step 4) and popped up for receiving mails from its default POP3 server (step 5), the local POP3d, which in turn reads the saved mail in personal mailbox (step 6).

Note that mails delivered to an off-line user will be stored in mailboxes on receiver MTA. When recipient gets online, MUA should be configured to automatically receive mails from domain mail server via POP3.

5.1.4 Advantages

In this infrastructure, several advantages are possible. Firstly, load balancing between

mail server and clients can be achieved. Mail servers can be offloaded since they don't have to receive e-mails when recipient is online.

Secondly, finer-grain personal configuration for mail processing like mail filtering can be done separately on each client in a distributed way, which further contributes to more offloading from mail servers. Global mail filtering can still be done on mail server.

Thirdly, peer-to-peer support can be integrated into current mail transfer mechanism. Users will be automatically notified of their new mails immediately when they get on-line.

Lastly, mail clients with different levels of capabilities, such as the presence of POP3d/SMTPd functionality, can be integrated in this infrastructure since capability information is also available through our location server. ULR query also serves as a way of capability exchange.

5.1.5 Implementation Issues

In this section, we introduce our implementation method for peer-to-peer mail transfer.

1. Location Server

Since the design of LDAP (Lightweight Directory Access Protocol v3) [69] server is optimized for reading, and the tree structure in LDAP is easy to maintain the different level of user information, we choose LDAP server as the location server. In our experiment, we use the "OpenLDAP" system developed by LDAP community which is an open source [70]. Besides, we also use the DB library developed by the University of Berkeley.

To store the user location information, we design one new object "ULR" (user location record) in LDAP server, and also define some suitable attributes as we need:

- (1) *Username*: the name to login.
- (2) *Userpassword*: the password to login and receive email.
- (3) *Email*: the email account.
- (4) *Online*: (True/False) to show whether the user is online.
- (5) *Ipaddress*: the IP address of the current user computer.

In practical, we insert the following definitions of attributes and objectclass into the LDAP schema:

```
(1) attributetype ( 9.8.7.6.5.4.3.2.1 NAME ( 'username' ) SUP name )
(2) attributetype (9.8.7.6.5.4.3.2.2 NAME 'userpassword'
EQUALITY octetStringMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.40{128} )
(3) attributetype (9.8.7.6.5.4.3.2.3 NAME ( 'email' )
EQUALITY caseIgnoreIA5Match
SUBSTR caseIgnoreIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} )
(4) attributetype (9.8.7.6.5.4.3.2.4 NAME 'online'
EQUALITY numericStringMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.36{16} )
(5) attributetype (9.8.7.6.5.4.3.2.5 NAME 'Ipaddress'
DESC 'IP address as a dotted decimal, eg. 192.168.1.1, omitting leading zeros'
EQUALITY caseIgnoreIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{128} )
(6) objectclass (9.8.7.6.5.4.3.2.6 NAME 'UserLocationRecord'
```



```
mailfilter: 22140.112.4.11
}
```

3. Mail Server

In our experiment, we choose the *sendmail* [67] package based on Linux environment. We modify *sendmail* to check the user status from location server when new mail is coming. If the user is online, it sends the redirect message back to sender mail server for sending the mail directly to the user. If the user is offline, it receives the mail for the user as usual.

4. Personal SMTP/POP3 Daemons and Login Procedure

In the user side, we focus on the Windows platform and develop the login/logout software and SMTP/POP3 daemon by the Java language. When one user logs in, he has to input his username and password, then the login procedure connects to location server for authentication, and reads the user's ULR record if it passes the authentication. After these operations finish, we run SMTP daemon with parameters (username, mail filters), and run POP3 daemon with parameters (username, password). Finally, the POP3 client immediately connects to domain mail server for receiving mails.

5.1.6 Future Work

There are some cases where the infrastructure still needs some modifications. Firstly, we have to deal with the cases for users behind firewall or NAT (Network Address Translator) [71]. In the case of firewalls that only allow certain types of traffic (for example, HTTP) to pass through, some encapsulation methods could be used, for example, Firewall Enhancement Protocol (FEP) [72]. For users inside a private network or NAT, some address and port translation has to be done, for example, Network Address Port Translation (NAPT) [71]. Secondly, we can also implement peer-to-peer support for other applications like FTP and HTTP, and a universal messaging service will be possible. Finally, security issues for Instant Messaging (IM) are also likely to occur in such environment, for example, IM Virus for MSN. All these need further considerations.

5.1.7 Conclusion

In the fast-changing world of efficiency, instant messaging and communications are critical for all people. The rapid growth of wireless devices and technology facilitates broader range of applications in wireless communications. Location service plays a major role in such an environment where user mobility management must be maintained for various services.

In this section, an infrastructure for peer-to-peer mail transfer mechanism was proposed for offloading mail servers and providing better personal customization on mail processing. This infrastructure can also be applied in all kinds of Internet applications where peer-to-peer support is needed.

5.2 Peer-to-Peer Support for File Transfer and Caching Mechanism

In existing Internet file transfer mechanism, proxy servers play a major role in load balancing and reducing duplicate file access requests for services like FTP and WWW. However, proxy servers are usually unaware of the availability of cached contents on other peer proxy servers. This is a waste of time since duplicate requests are needed. Unnecessary traffic can be reduced if cooperation and coordination among peer proxies can be utilized.

In this section, peer-to-peer support was incorporated in ordinary file transfer and caching mechanism to reduce unnecessary processing time and storage. Through the location service, hosts requesting file services can dynamically determine if a copy is available and its current location. Work load for file servers will be greatly reduced, and personalization of file transfer configuration can be fully supported.

5.2.1 Introduction

With the tremendous growth of the Internet, numerous networking applications such as WWW (World-Wide Web), E-mail, and FTP (File Transfer Protocol) [54] have been widely used. Specifically, file access applications like WWW and FTP have become ubiquitous and central to many people's daily lives. However, in current file transfer mechanism, FTP and Web servers play a critical role since all file access requests require the intervention of these servers. This could result in overloaded server and no service could be provided. Therefore proxy servers have been widely deployed to eliminate unnecessary transfers for file objects already retrieved by other clients.

When a user browses a web page through a proxy server, the URL (Uniform Resource Locator) of requested web page will be checked if a copy is available on proxy server. If so, no further outbound connections are needed since the page is already fetched. If not, the proxy server will act like an agent for the client and make HTTP (HyperText Transfer Protocol) [73] requests to the real web server as indicated in the URL on behalf of the client.

However, communication and coordination among peer proxy servers are still not much used. Proxy servers are usually configured in a hierarchical way where parent and sibling proxies are manually organized. When a proxy doesn't contain the requested file object (a *cache miss*), it may make Internet Cache Protocol (ICP) [74] requests to see if any of its neighbor proxies has the object. "Neighbor hits" where neighbor proxy has the object may be fetched from either parent or sibling proxy, but "neighbor misses" must be forwarded only to parent proxy. Since parent and sibling relationships must be manually configured in existing implementations like *squid* [75], reutilization of existing cached

contents on peer proxy servers can be very difficult. Duplicate file replications among different proxies are still possible and caching efficiency may be further improved.

With the rapid development of various mobile devices, wireless LANs (WLANs) [8] have become more popular as an alternative network access method. In infrastructure mode, mobile nodes can connect to the wired network via access points (APs) as if they have been directly attached. However, since APs are limited in range, mobile nodes may roam into the ranges of different APs. IP roaming problem occurs if different APs are located on different subnets. Mobile IP scheme [9] is one of the most common ways to solve the IP roaming problem.

On the other hand, peer-to-peer technology has been widely deployed in various applications, for example, file sharing software like Napster [4] and ezPeer, instant messaging software like ICQ and MSN, and open source protocol like Jabber [63] and GnuTella [5, 10]. Moreover, the distinction between centralized and decentralized applications has become blurred to leverage the advantages of both. Therefore, an infrastructure for integrating current Internet client-server file transfer mechanism and peer-to-peer file sharing applications was proposed to provide better integrated services. In a mobile environment, each mobile node may act as a peer proxy in which the cached content could be utilized by other nodes. Therefore, our focus is on better utilizing existing proxy caching mechanism and web cache communication and coordination protocols in peer-to-peer applications.

5.2.2 Motivation

In this section, the current architecture for file transfer and caching and its shortcomings will be briefly reviewed, and our infrastructure will be proposed as a feasible solution.

In current Internet file transfer mechanism, several protocols are used: HTTP [73] for transferring web pages, FTP [54] for transferring files, and ICP [74] for inter-proxy communication. As shown in Fig. 24, a typical scenario for current file transfer mechanism is illustrated.

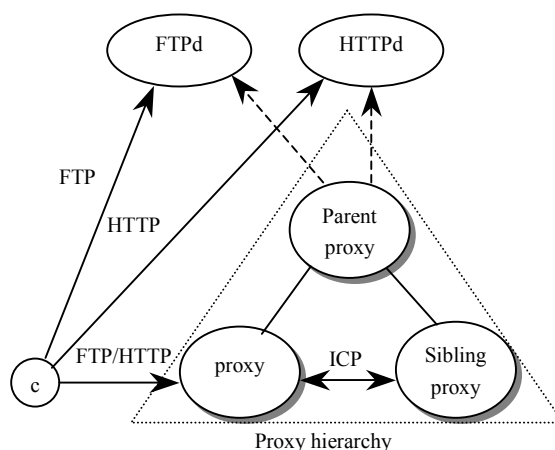


Fig. 24 shows a typical scenario for file transfer and caching, where a proxy hierarchy is deployed.

As shown in Fig. 24, users can browse a web page or access a file with a specific URL (uniform resource locator) via browser like Microsoft Internet Explorer or Netscape. File access requests are made via either HTTP or FTP directly or through a proxy server. Since common files on the same web site may be requested by different users, proxy server is usually deployed as a cache of similar requests for domain users.

In order to get better performance, more than one proxy servers may be deployed in a hierarchical way. There are many possible deployment schemes for proxy servers with regards to the relative place of a cache between client and server. Proxy caching as described above is the most common one. The other possible schemes include personal proxy server where cache is on each individual client, transparent proxy caching where proxy setting is transparent to clients, reverse cache where the focus is on server not clients, and active caching where applets are used for caching dynamic documents [76]. In fact, these schemes may be deployed simultaneously with better overall performance.

Proxy server configuration in a browser can be done automatically by protocols like WPAD (Web Proxy Automatic Discovery) [77], through a PAC file (Proxy Auto-Config File Format) [78], or manually configured.

The web caching mechanism works fine, but there are several problems that affect the performance of file retrieval. Firstly, the load on a proxy server is heavy in terms of file storage and time for HTTP/FTP processing. Each proxy server has to deal with every file access request from domain clients. Usually cache hits in proxy server will result in better performance for retrieving file objects. However, in the case of busy proxy server or even server failure, the performance would be worse than without proxy.

Secondly, file objects may have been cached by other peer proxies or personal proxy servers which are unknown to our proxy server. Although inter-cache communication protocols such as ICP [74], WCCP [79], HTCP [80], CARP [81], and Cache Digest [82] have been proposed, they are not widely implemented. Moreover, inter-proxy communication relationships are usually manually configured and dynamic addition and removal of peer proxy can be difficult.

Thirdly, proxy configuration in a browser is usually not versatile enough. In the case of busy server or server failure, no fallback mechanism for bypassing overloaded server is provided. This could result in worse performance than direct connection without proxy.

Fourthly, personalization cannot be done very efficiently in proxy server. For example, it's difficult to configure a content filter for each individual domain user. That would be time-consuming and impractical. It's common to configure on firewall or proxy server a content filter for the whole domain. But for each individual domain user, a finer-grain control of configuration is needed, for instance, a content filter for each user, which is more

reasonable since each user may want to filter content from different sources.

In order to offload proxy server and to provide complete customization in file retrieval, a peer-to-peer infrastructure for file transfer and caching was proposed. Specifically, we want to bypass a busy or overloaded proxy server if there are other replications for requested objects. Cached content on peer proxy servers can be utilized for improving cache utilization. Besides, users can have their own configurations for file processing like content filtering.

5.2.3 Infrastructure

As shown in Fig. 25, an infrastructure for peer-to-peer file transfer is illustrated.

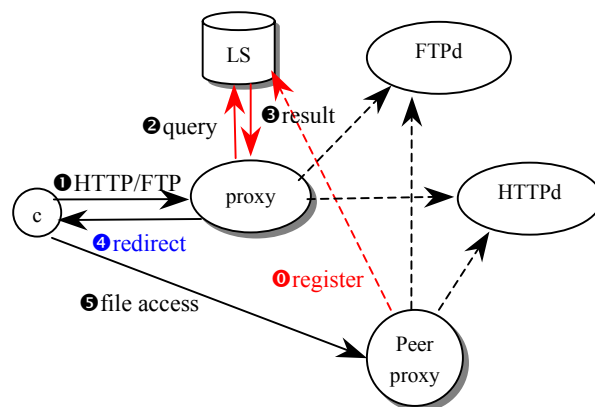


Fig. 25 shows peer-to-peer support (*redirect mode*) for file transfer and caching.

When clients issue HTTP/FTP requests to proxy server, it will first query the location server for possible replications of the given URL. Since peer proxy has registered to location server, its presence and content will be known to location server. After receiving reply from location server, proxy server will issue redirect messages to client which will then try to access from the peer proxy.

In this infrastructure, searching for proxy servers can be transparent to users since location service lookups can be done automatically by proxy servers. Besides, the availability of peer proxy can be used as a way of load balancing between servers. Since the latest information for each cache in a domain can be obtained, the most suitable proxy server can be reached and load balancing can be achieved. Fault tolerance mechanism can also be provided in the case of proxy failure. As shown in Fig. 25, key components in the infrastructure include: location servers (LS), proxy servers, and FTP/HTTP servers. The functional description of each component is provided as follows.

1. Location Server

Location server is responsible for storing the current location and content index for each peer proxy. These include hostname, current IP address, URLs for cached content, and resource profiles (for example, access control list). Since the peer proxy server may be

changing its location or contents frequently, the amount of data update may be quite large. Therefore, Resource Location Records (RLRs) can be stored in a distributed way, for example, one location server for each domain (like DNS server). RLRs for cached URLs on each proxy server are stored in location server of its home domain.

Most of the relevant works in location service are related to geographical positioning of mobile nodes in a wireless network, the location of servers, and location-based services. They mainly focused on the physical positioning of mobile nodes or servers, not the current way of accessing a particular resource, for example, the IP address of currently available peer proxy with the requested data.

As shown in Fig. 26, there are two possible operations for a location server: update and query. Proxy servers update their current location (IP address), URLs and resource profiles for cached content to location server when they are first added, changed, or removed from the domain. On the other hand, peer proxy servers query the location server for available replication of a particular URL in order to retrieve resource from it. In other words, location server has to be coupled with the management of resource addition/removal. Proxy servers must do registration/de-registration when being added or removed.

However, when mobile node is roaming into a foreign network, it must register to its home location server for location update. This can be done directly or through the help of location server in foreign network (Indirect Update). For a mobile node to detect it has left its home network, the advertisement based mechanism used in Mobile IP [9] or hint based move detection method [83] can be deployed.

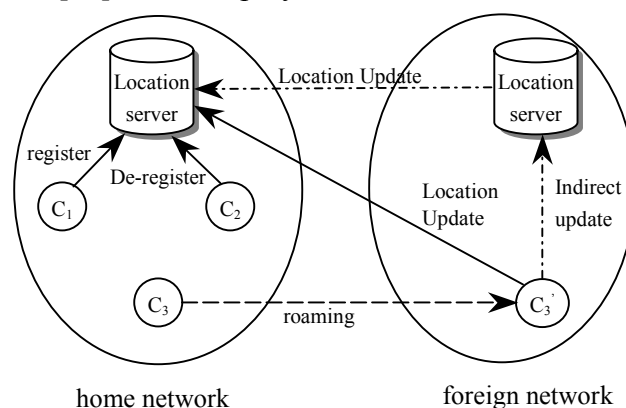


Fig. 26 shows the operations of location servers.

2. Proxy Server

In our infrastructure, each proxy server has to register to its domain location server when the cached contents are added, changed, or removed. The current IP address and the cached contents are indexed by the location server. When a peer proxy needs to search for the availability of a specific URL, a location service query will be issued and the result will be checked to see if redirect is needed.

There are several deployment alternatives for peer-to-peer file transfer and caching.

Besides the redirect mode depicted in Fig. 25, two other schemes are possible, proxy mode and server-to-server copy mode, which are illustrated as follows separately in Fig. 27 and Fig. 28.

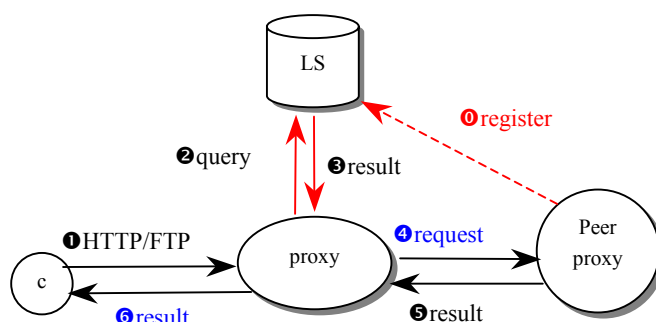


Fig. 27 shows an alternative deployment scheme (*proxy mode*) for peer-to-peer file transfer and caching.

In proxy mode, file access requests from clients are repeated on local proxy where file objects fetched from peer proxy are cached. This “greedy” caching mechanism would require more storage requirement but less penalties for a cache miss will be experienced since as much content as possible will be cached. But it’s not suitable for proxy server load balancing since the load of proxy server is heavy.

On the other hand, in server-to-server copy mode, file access requests for clients are not redirected to peer proxies. Instead, notifications to both client and peer proxy are issued by local proxy and the real file transmission takes place without the intervention of local proxy. This is illustrated in Fig. 28.

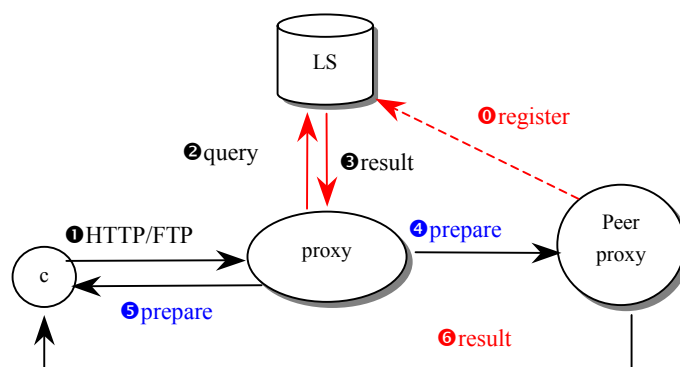


Fig. 28 shows *server-to-server copy mode* for peer-to-peer file transfer and caching.

This caching mechanism has the advantage of load balancing for *redirect* mode, without much intervention of local proxy.

Note that existing inter-cache communication protocols can still be used in different conditions. For example, ICP [74] can be used for inter-proxy communication protocol, but modifications to ICP are required for supporting mechanisms such as *server-to-server copy*. On the other hand, cache digests [82] can be used in which full index doesn’t have to be built. Only the cache digests for each peer proxy are needed.

Among these alternatives, redirect mode is better for load balancing, while proxy

mode has the advantage of “greedy” caching in local proxy. Server-to-server copy mode has the advantage of redirect mode without much intervention for local proxy server if inter-proxy communication protocol support is available.

5.2.4 Advantages

In our architecture, there are several advantages over current file transfer mechanism. Firstly, file (WWW, FTP, proxy) servers can be offloaded since replication can be found via location service lookups. Load balancing can thus be achieved. Secondly, peer-to-peer support for file transfer can be achieved, and integration of existing file transfer protocols with peer-to-peer applications can be done. Thirdly, personal configuration for file server, for example, content filtering, such as ACL: allow/deny <source URL>, can be fully supported.

5.2.5 Security Concerns

When one mobile node is roaming into a foreign network, authentication and authorization is required before it's granted network access. For example, IEEE 802.1x [37] can be used as the network access control mechanism as shown in Fig. 29.

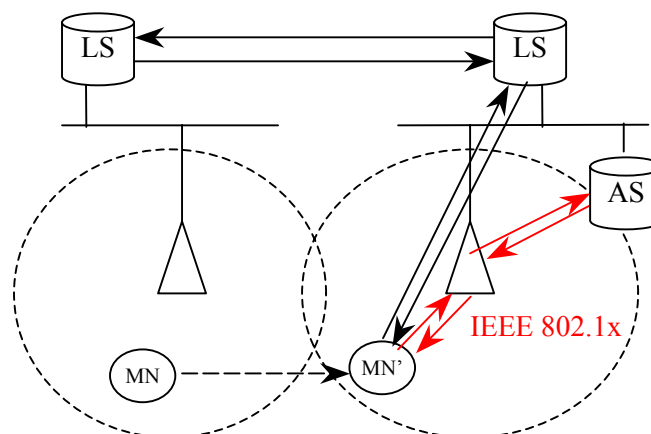


Fig. 29 shows the authentication and authorization for mobile nodes, where AS is the Authentication Server, and LS is the Location Server.

On the other hand, for each operation of update and query, authentication and authorization are required to ensure the correctness of each record in location server.

5.2.6 Future Work

Most importantly, file authenticity is the most difficult problem. We have to make sure that the file objects registered by peer proxies are indeed the objects as they claim. The authenticity and non-repudiation principle is most important. Besides, conditions for users behind firewall and users inside private network have to be dealt with.

5.2.7 Conclusion

In this section, a peer-to-peer support for file transfer and caching mechanism was proposed. Through the sharing of cached contents of peer proxies in the same domain, we could further improve the cache utilization and reduce unnecessary duplicate file access requests. In addition, load balancing for overloaded proxy servers can be achieved by means of proxy redirecting and server-to-server copy operations incorporated in our scheme.

5.3 Peer-to-Peer Support for Mobile IP scheme

Mobile IP scheme is the most popular solution to IP roaming problem. Despite its routing transparency, the large overhead in packet redirecting and triangle routing impedes its wide deployment. Although improvements on route optimization have been proposed for mobile IP mechanism, it takes a long time to deploy.

In this section, a seamless IP roaming framework was proposed for addressing the routing inefficiency problem. A filtering driver is implemented in each mobile node for move detection, address allocation, and IP address mapping, and DHCP server is deployed for address pool management and IP change notification. In our scheme, DHCP-based management can also be integrated. Overhead for triangle routing and network service disruption can be avoided while maintaining the flexibility of routing transparency for mobile nodes.

5.3.1 Introduction

With the advent of mobile devices like notebook computers and PDAs (Personal Digital Assistants), IEEE 802.11 [8] Wireless Local Area Network (WLAN) environment is becoming increasingly popular. People can connect to the wired network with a mobile device through wireless access points (APs) when operating in infrastructure mode. Since APs may be attached to different subnets, mobile devices roaming among these APs will be forced to change their IP addresses that may cause serious problems. Firstly, existing network connections are forced to terminate since a new IP address at foreign network must be allocated before manually resuming connection. Secondly, network service disruption is inevitable and users need to reconnect to network services at a later time. Although layer 2 (link layer) roaming is supported in most IEEE 802.11 [8] implementations, solutions to layer 3 (network layer) roaming problems are still not satisfactory enough.

For the first problem, one possible solution is to support DHCP (Dynamic Host Configuration Protocol) [22] across different subnets. For DHCP packets to pass through routers, DHCP relay agents may be deployed and DHCP server has to be configured to allocate IP addresses from different IP segments for hosts on different subnets. However, since the IP address of a mobile node does change, the upper layer of TCP/IP protocol stack, specifically, transport and higher layer, will be affected and existing network applications

will be forced to terminate. In other words, network service disruption cannot be solved with this method.

5.3.2 Mobile IP

The most popular solution to IP roaming problem is IP mobility support [9], or Mobile IP (MIP) scheme, as shown in Fig. 30. With this mechanism, the IP addresses of mobile nodes (MNs) are physically unchanged at all. Instead, a care-of address is obtained as MN roams into a foreign network. Then a pair of mobility agents (home agent and foreign agent) are responsible for redirecting packets destined for the original IP address (or home address). Home Agent (or HA in short) will monitor packets on the original subnet and once packets for MN are received, they will be tunneled to foreign agent (or FA in short) on foreign network, which in turn will forward the packets to the care-of address of MN. Notice that when MN needs to send packets, it follows the usual routing mechanism without the intervention of its home agent.

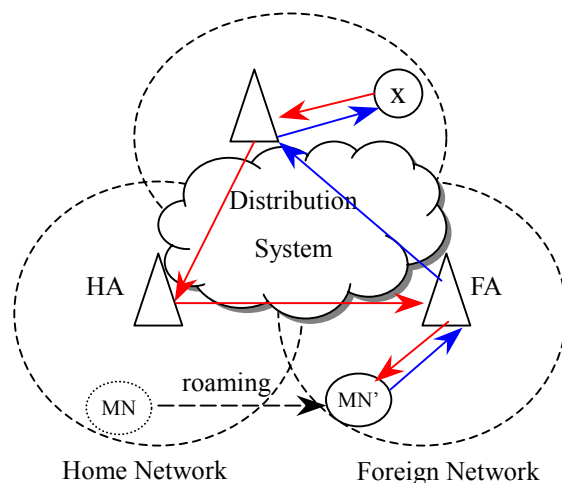


Fig. 30 shows the operations of mobile IP mechanism and triangle routing.

One advantage of mobile IP is that packet redirecting is transparent to mobile nodes. Mobile nodes are always identified by their home address regardless of their point of attachment. Since the IP addresses remain unchanged, TCP/IP protocol stack will not be affected. However, the main drawback of mobile IP is its routing inefficiency. As pointed out in the paper entitled “Optimized Smooth Handoffs in Mobile IP” [84], three issues in mobile IP are present: triangle routing, out-of-date location information, and frequent handovers. The overhead of triangle routing for mobile nodes roaming in foreign network is quite high since all packets for MN have to go through its HA. Although Internet drafts for route optimization has been proposed [24], mobility agents have to be modified accordingly that will take a long time to deploy. In addition, if there are frequent handovers, the overhead of location updates for mobile nodes will be higher. Specifically, we need a handover mechanism with minimum delay (fast handover [85, 86]), minimum packet loss (smooth handover [84, 87]), or both (seamless handover). Besides, mobility support using

SIP (Session Initiation Protocol) [26] has also been proposed [88].

In this section, a different approach from mobile IP was proposed to address the routing inefficiency problem. The goal is to provide a seamless IP roaming environment for wireless LAN with minimum handover delay and optimum routing efficiency.

5.3.3 Our Architecture

In our design, we choose a more direct method in maintaining MN location changes. Mobility agents are not necessary since location information is distributed on each mobile node. When roaming to a foreign network, MN allocates a temporary address in new subnet from DHCP server which is similar to care-of address in mobile IP. However, since the original permanent address are recorded in upper layers of TCP/IP protocol stack of existing connection parties, changing the IP address will affect the protocol stack which will be complex and unwanted. Therefore, in our infrastructure, IP addresses are physically changed only at lower layer in the filtering driver, and the changes will not be propagated to the upper layers of protocol stack since the filtering driver does all the conversions between permanent address and temporary address. Upper layers “believe” that the connection party is still the same host with the same permanent address. As compared to mobile IP, the IP address mappings are kept in the filtering driver rather than in mobility agents. Whenever roaming into a different subnet, a mobile node only needs to notify the other parties of existing connections to reflect such IP changes. In this way, existing connections will be able to remain connected and network services will not be interrupted.

As shown in Fig. 31, an infrastructure for seamless IP roaming is illustrated and the interactions among mobile nodes and DHCP server are described as follows.

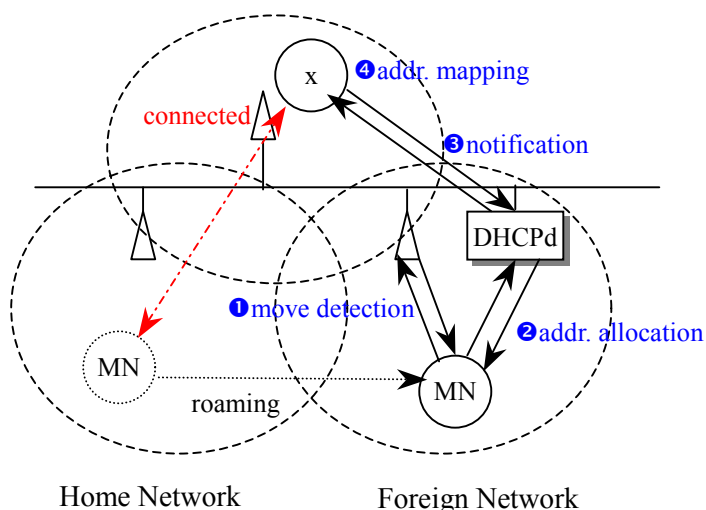


Fig. 31 shows an example of our infrastructure where host x is connected to MN.

5.3.4 Operations

As shown in Fig. 31, several steps are needed for seamless IP roaming to work as follows:

1. *Move Detection*: When a mobile node MN roams into a foreign network, move detection methods are required for MN to start handover operation. For example, MN will detect a subnet change via *Agent Advertisement/Solicitation* messages in advertisement based methods like Lazy Cell Switching and Eager Cell Switching [9] or via subnetwork layer handover “hints” in hint based methods like Hinted Cell Switching [83].
2. *Address Allocation*: MN then makes DHCP requests to allocate a *temporary address* from DHCP server in new subnet which is similar to *care-of address* in mobile IP. However, this new address is effective only in the filtering driver where conversions to and from *permanent address* take place.
3. *IP Change Notification*: After obtaining a new address IP_{temp} , MN has to notify all its connection parties of such changes which can be done by DHCP server (for example, via ICMP Gateway Redirect or proprietary protocol) or by MN itself. Since each DHCP client has to pass the authentication and authorization from DHCP server before gaining network accesses, it’s natural to issue *IP change notification* from DHCP server instead of MN.
4. *IP Address Mapping*: After *IP change notifications* are received in the filtering driver of all existing connection parties of MN, the new IP address mapping (IP_{perm} , IP_{temp}) for MN will be kept. Any subsequent incoming packets with source IP address IP_{temp} will be changed into IP_{perm} , and outgoing packets with destination address IP_{perm} will be changed into IP_{temp} . Although the MAC address of MN does not change, the packet must be routed to a different subnet, thus the destination MAC address should be that of the router. Besides, checksum re-computation in each packet is also required.

5.3.5 Key Components

The functions of each component in our infrastructure will be described as follows.

1. Filtering Driver

Since the changes of IP addresses are only known to lower layer, an intermediate filtering driver is required for each host in our design as shown in Fig. 3. At the driver layer, it’s responsible for move detection, address allocation, and keeping IP address mappings and doing related address conversions in packets which are depicted as three separate modules in Fig. 32.

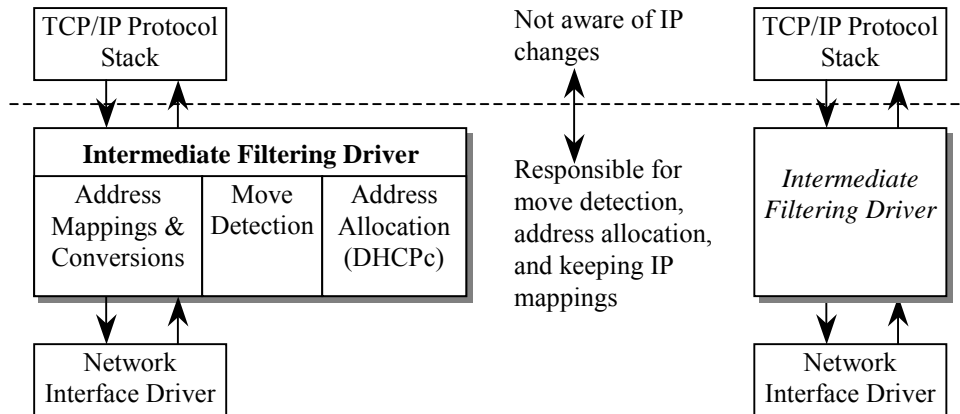


Fig. 32 shows the layers of TCP/IP protocol stacks and modules in filtering driver.

(1) Move Detection Module

There are two main categories for move detection methods: advertisement based and hint based [83]. We can use the same *Agent Advertisement/Solicitation* messages for move detection as in Mobile IP. It's based on ICMP Router Discovery messages [89]. Alternatively, we could have used *hint based* methods that incorporates inter-layer communication handover "hints" (beacons). Therefore, in this module, *Agent Advertisement/Solicitation* messages or handover hints need to be processed for the detection of subnet change.

(2) Address Allocation Module

When a mobile node roams into a foreign network, it will begin to allocate its temporary address as in Mobile IP. The easiest way is to use DHCP. So there will be a DHCP client in the filtering driver. However the IP change will only pertain in driver layer. The real IP layer is unaware of this change.

(3) IP Address Mapping and Conversion Module

Since each mobile node can roam into different foreign networks besides its home network, we have to keep track of the mappings of its permanent address and temporary address which can be obtained from IP change notifications as described later. All outgoing packets with destination address as permanent address will be changed to temporary address, while all incoming packets with source address as temporary address will be changed to permanent address. All these conversions are done only in the driver layer.

2. DHCP Server

DHCP server is responsible for the address pool management for mobile nodes as well as fixed hosts. Besides, since the IP addresses are changed at driver layer, a mechanism will be needed for IP change notification. This can be implemented in DHCP server since it's mandatory for each DHCP client to allocate leases before accessing the Internet. Besides, in the case of hardware-based APs, *Agent Advertisement* functionality should be supported by

DHCP server if advertisement based move detection methods are used.

5.3.6 Design Issues

Move detection modules are needed for the detection of subnet change. This includes the processing of Agent Advertisement/issuing Solicitation messages in advertisement based methods or subnetwork layer handover “hints” (beaconing) that’s passed on to upper layers as in hint based methods [83]. The latter requires inter-layer communication which is lacking in most of the current layer 3 roaming solutions.

There are several design alternatives in our infrastructure. Firstly, IP change notification can be issued by each individual mobile node or by DHCP server. Since DHCP based management like [90] can be deployed, DHCP server will be responsible for the authentication and authorization of network accesses. Therefore, it’s natural to make IP change notifications there. For MN to issue notifications, it should change its IP address after notification has been issued.

Secondly, IP address mappings can be kept by each mobile node or DHCP server. However, keeping IP address mappings in DHCP server is like keeping them in mobility agents. DHCP has the same responsibility for issuing Agent Advertisement messages as in mobility agents.

Thirdly, move detection can be advertisement based or hint based. For advertisement based methods, *Agent Advertisement/Solicitation* messages for mobility agents or DHCP server can be processed. For hint based methods, MAC layer handover hints must be propagated up to the filtering driver.

Finally, location service could have been added into our infrastructure for the completeness of our solution as shown in Fig. 33.

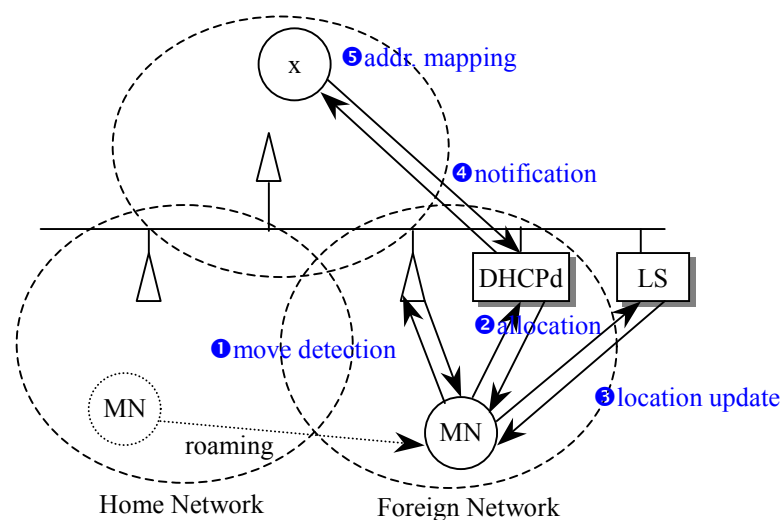


Fig. 33 shows an example of our infrastructure with location service support.

In our infrastructure, only existing connection parties for MN are notified. It's impossible to reach MN from other hosts after MN roams into a foreign network. Therefore, location server can be used to keep the latest location of mobile nodes. Location updates for each roaming is required. For an arbitrary host y to connect to MN, it will first contact its domain Location Server for the latest location of MN and then establish connections. If MN roams into another network after connected with host y , it will get notified in our IP change notification mechanism.

5.3.7 Comparisons

Some comparisons between Mobile IP and our mechanism are as follows:

1. Mobility Agents

HAs and FAs are required for mobile IP to work. And this brings lots of overhead for packet redirecting and tunneling. The address mappings of home and care-of addresses for mobile nodes are kept in these mobility agents.

In our design, no extra agents are needed for the redirecting of packets. Instead, location information is kept in each mobile node. Only move detection and address allocation operations are required, and the mapping of a mobile node's temporary address and permanent address will be kept in filtering driver of each mobile node. However, DHCP server is necessary in our infrastructure since address allocation and IP change notification are supervised by DHCP server.

2. IP Address: to Change or not to Change

For mobile IP, the IP address of a mobile node doesn't change at all. Instead, a care-of address is obtained from FA or DHCP server when roaming to foreign network.

In our design, IP addresses will change, but the changes only pertain in lower layer of protocol stack. Upper layers will not notice it. The role of temporary address and permanent address are similar to care-of address and home address in mobile IP. But the IP address mappings are kept in mobile node instead of mobility agents.

3. Deployment

It's very easy to deploy our scheme. Only a DHCP server for each domain and a filtering driver for each mobile node are needed. Since the filtering driver takes the form of a driver plug-in, it's easy to install for both Microsoft Windows or Linux platform.

For mobile IP scheme to be deployed, mobility agents are required in each domain. Moreover, a mechanism for care-of address allocation is also needed using DHCP.

4. Extra Overhead

In our design, the overhead of IP change notification for mobile node to its existing connection parties is needed. However, we avoid large overhead for packet redirecting.

When a host roams into a foreign network, other hosts cannot reach it if DNS records are not updated to reflect server location changes. Service Location Protocol (SLP) [21] or a generalized location service protocol can be used to address such issues.

5. Detailed Comparison

As a more detailed comparison, we will illustrate an example of roaming in mobile IP and our scheme. A mobile node MN with home address 140.112.29.2 is connected by a host x with IP address 140.112.31.104 while MN roams into foreign network 140.112.30.*. The detailed steps when roaming occurs will be illustrated for both mobile IP and our scheme.

Firstly, for mobile IP scheme, HA and FA are necessary and they are responsible for redirecting packets to MN as shown in Fig. 34. As MN roams into foreign network 140.112.30.*, MN will receive Agent Advertisement message from FA and knows that it's now roaming into a foreign network. Then MN will try to obtain its care-of address which is assumed to be 140.112.30.3 as allocated from DHCP server. After obtaining its care-of address, MN needs to register its mapping $\{(140.112.29.2), (140.112.30.3)\}$ to HA. Then the roaming operations are complete.

Once host x continues to send packets to MN, it will still send them to its home address 140.112.29.2. Since HA will intercept all packets for MN, these will get redirected and tunneled to FA, which in turn will forward the packets to 140.112.30.3, the care-of address for MN. The path of packet transmission is: $\{x, R1, HA, R1, R2, FA, MN'\}$.

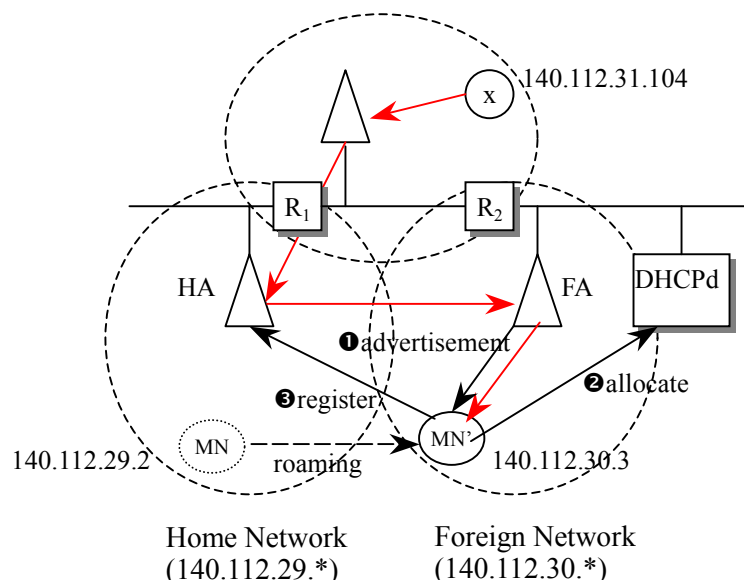


Fig. 34 shows an example of roaming in mobile IP scheme.

Secondly, as shown in Fig. 35, our scheme is illustrated. Mobility agents are not needed. Instead, when MN roams into foreign network 140.112.30.*, the filtering driver in MN will first do move detection. Since mobility agents are not present, hint based methods can be used and handover hints from lower layer of MN will be received. Then a temporary

address 140.112.30.3 is allocated from DHCP server. After that, IP change notification will be delivered from DHCP server to the connection parties of MN, in this case host x only, in which the address mapping $\{(140.112.29.2), (140.112.30.3)\}$ will be kept. Roaming operations are then complete.

Once host x continues to send packets to MN, it will still send to its permanent address 140.112.29.2. But the packet will be modified in the filtering driver of host x where destination address 140.112.29.2 will be changed into 140.112.30.3 since the address mapping is already kept when IP change notification was received. So the path of packet transmission will be: {x, R2, MN'}.

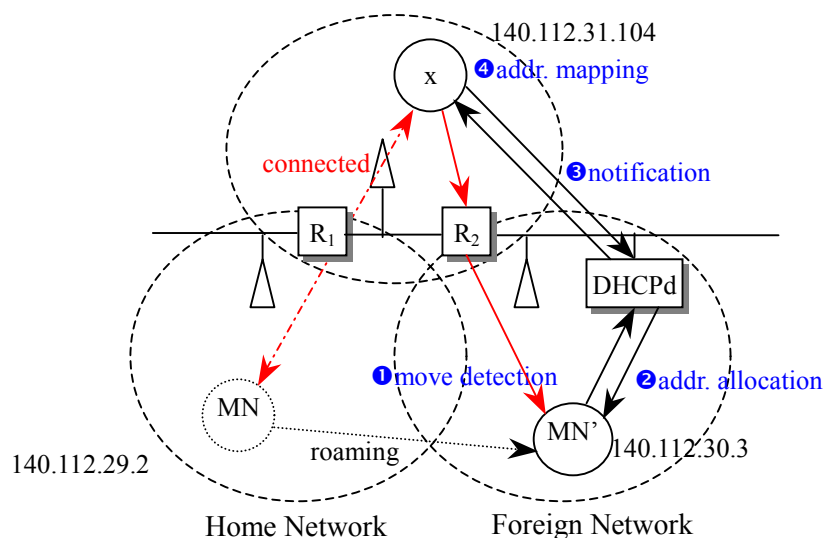


Fig. 35 shows an example of IP roaming in our scheme.

In summary, modifications required in mobile IP are: the addition of mobility agents, packet tunneling overhead, and registration/deregistration. In our design, the changes needed are: filtering driver for move detection, address allocation, and IP address mapping at each mobile node, and DHCP server that can make IP change notification.

5.3.8 Conclusion

In this section, a framework for seamless IP roaming was proposed as an alternative to Mobile IP. When mobile node roams into a foreign network, it is integrated into the new domain since the host behaves the same as the hosts in that domain except for the underlying address mapping tables and necessary address conversions. No extra packet redirecting or tunneling is required for layer 3 roaming. The need to register/deregister is also alleviated. Therefore the handover overhead will be reduced to its minimum.

5.4 Experimental Results

In this section, experimental results will be illustrated and discussed. First of all, we will focus on the configuration of experimental environments for peer-to-peer mail transfer support as shown in Sec. 5.1.

5.4.1 Experimental Environment

In this experiment, we want to demonstrate the load balancing effect for mail servers by incorporating the location server query mechanism. When a mail recipient is online, the current location will be registered at its home location server. When the mail server receives an e-mail, the online status of the recipient indicated in the RCPT command will be checked. If he/she is online, a SMTP redirect message (reply code 551) will be replied and the mail will be delivered directly to the recipient.

The environment for mail delivery experiment is set up as follows. A Linux server (Slackware distribution 7.0) with kernel version 2.2.19 was configured as the local mail server. Since the source code for the most widely deployed mail server *sendmail* is more complex and difficult to modify, a simplified version of SMTP daemon developed by myself was adopted as the standard mail server (which will be called the *normal* version). The modifications needed for incorporating location server query and SMTP redirect mechanism are made to this standard mail server as the enhanced version (which will be called the *LS* version). In our implementation, location server was implemented on a LDAP server (OpenLDAP version 2.1.2) and caching was enabled for LDAP queries.

As for the test data, there are six files with different sizes delivered automatically to each mail server, namely the *normal* and *LS* version, respectively. Comparisons of server loads between these two versions of mail servers are made in terms of the server processing time for each incoming SMTP connection.

Besides sending mails directly to the local mail server, we also try to show the situations where mails from other domains are relayed to our local mail server. This is closer to the real situations. Since the delivery of mails from remote to local mail server depends on the loads of remote and local mail servers and also the network bandwidth utilization between remote and local networks, the order of mail arrivals may not be the same as the order they are sent. Therefore, some constant delays between successive mail deliveries are added to guarantee the ordering of mail arrivals. This will not affect the processing time for each mail.

5.4.2 Results

In this section, figures and tables for the experimental results are illustrated and discussed. As shown in Table 2, the test data and their sizes are illustrated.

Table 2 shows the test data and their sizes.

File Name	Test.eml	Shutdown.eml	Bill.eml	Mars.eml	Careful.eml	Commercial.eml
File Size (bytes)	73	830	4512	50784	511851	6772105

Since the comparisons between *normal* and *LS* versions of mail servers have to be made under *local* and *remote* mail delivery situations, we can divide them into four cases.

Case 1. Local Delivery, Normal Version:

As shown in Table 3 and Fig. 36, when local mails get delivered to the mail server, only local server load will affect the processing time of mails. Therefore, only when the file size gets very large (several megabytes) will the server load increase tremendously.

Table 3 shows the processing time for case 1.

File Name	Test.eml	Shutdown.eml	Bill.eml	Mars.eml	Careful.eml	Commercial.eml
Processing Time (s)	0.009978	0.011494	0.010228	0.012368	0.097825	1.249285
Variance	3.80E-07	0.000189	7.61E-07	3.07E-08	0.000438	0.005042

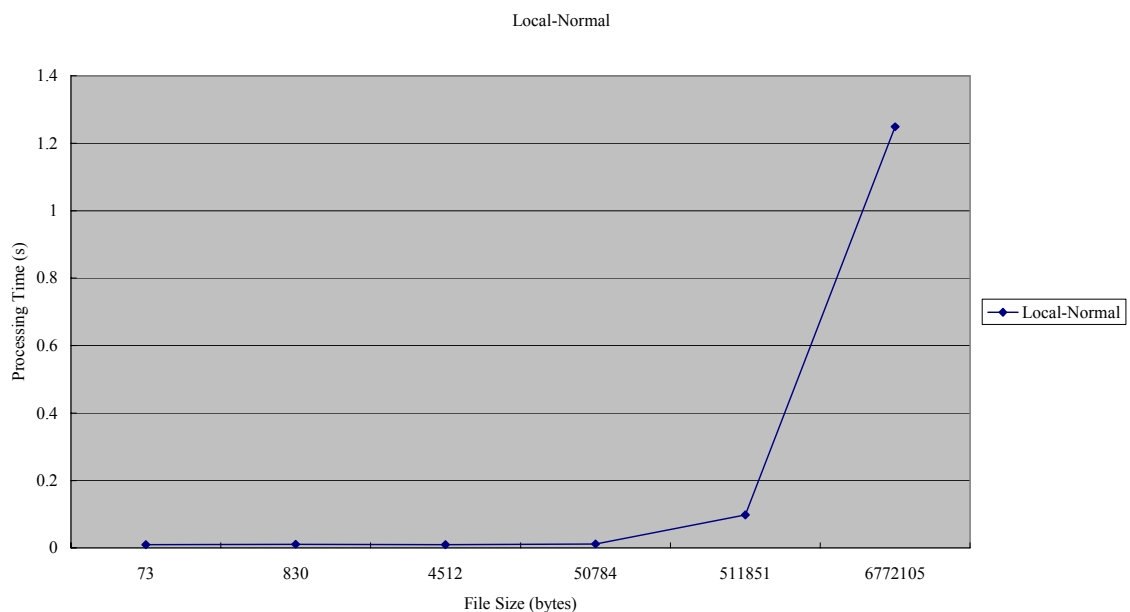


Fig. 36 shows the processing time for *local* mail delivery to the *normal* mail server.

Case 2. Local Delivery, LS Version:

As shown in Table 4 and Fig. 37, the processing time of mail delivery with the modifications of location server query and SMTP redirect is constantly stable with respect to different file sizes. Note that the slight slower processing time for file “test.eml” is due to the caching overhead for LDAP server queries.

Table 4 shows the processing time for case 2.

File Name	Test.eml	Shutdown.eml	Bill.eml	Mars.eml	Careful.eml	Commercial.eml
Processing Time (s)	0.011811	0.011678	0.011640	0.011564	0.011610	0.011699
Variance	9.81E-07	9.78E-07	7.43E-07	7.79E-07	1.03E-06	2.14E-06

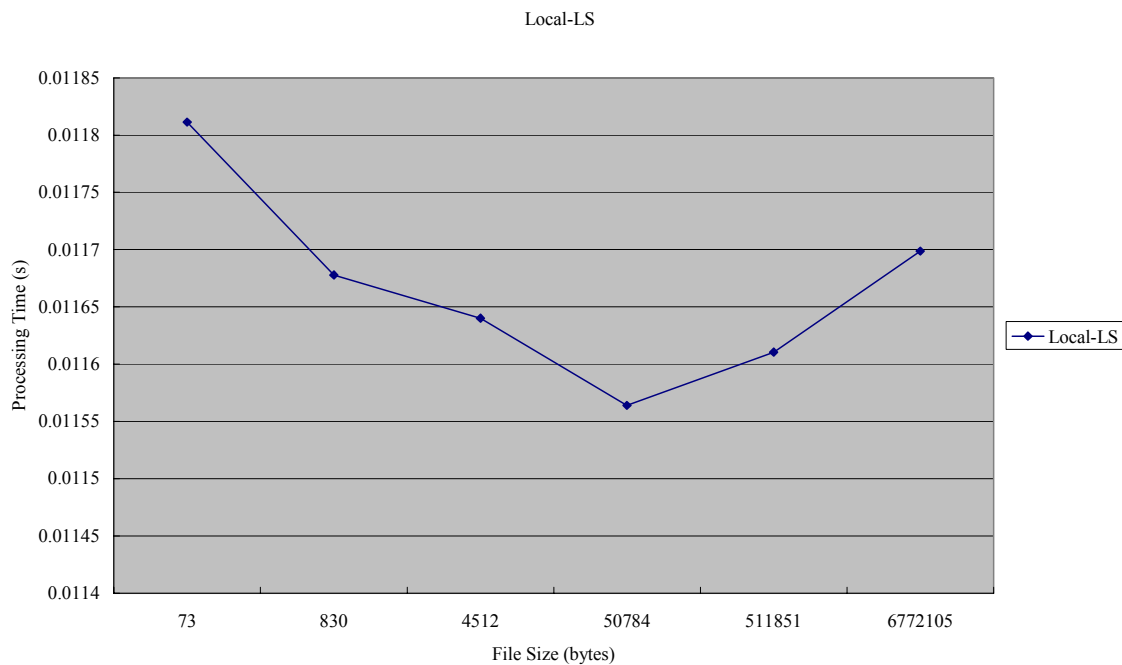


Fig. 37 shows the processing time for *local* mail delivery to the *LS* version of mail server.

Case 3. Remote Delivery, Normal Version:

As shown in Table 5 and Fig. 38, when mails get delivered to the remote mail server, more processing time are needed due to the overheads for local and remote servers and the networks in-between. Note that the unusual large variances for files “mars.eml” and “careful.eml” are due to the intensive mail deliveries to the remote mail server. Since too many mails get spooled on remote mail server, the processing time gets larger.

Table 5 shows the processing time for case 3.

File Name	Test.eml	Shutdown.eml	Bill.eml	Mars.eml	Careful.eml	Commercial.eml
Processing Time (s)	0.040269	0.043669	0.074207	1.92954	2.93179	14.761319
Variance	1.17E-05	1.28E-05	4.28E-05	6.226235	5.841845	0.2701019

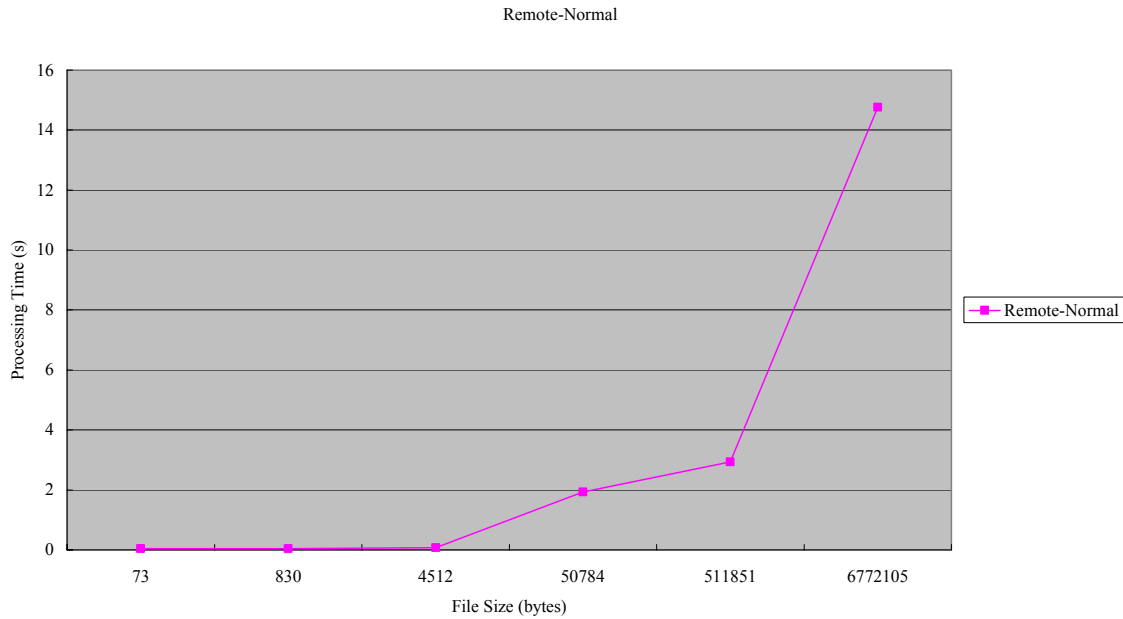


Fig. 38 shows the processing time for *remote* mail delivery to the *normal* mail server.

Case 4. Remote Delivery, LS Version:

As shown in Table 6 and Fig. 39, the processing time of mail delivery with the modifications of location server query and SMTP redirect is also constantly stable with respect to different file sizes. Note that the slight slower processing time for file “test.eml” is due to the caching overhead for LDAP server queries.

Table 6 shows the processing time for case 4.

File Name	Test.eml	Shutdown.eml	Bill.eml	Mars.eml	Careful.eml	Commercial.eml
Processing Time (s)	0.025897	0.023554	0.023378	0.023523	0.023353	0.024633
Variance	2.28E-05	9.95E-06	7.47E-06	1.18E-05	8.03E-06	9.07E-06

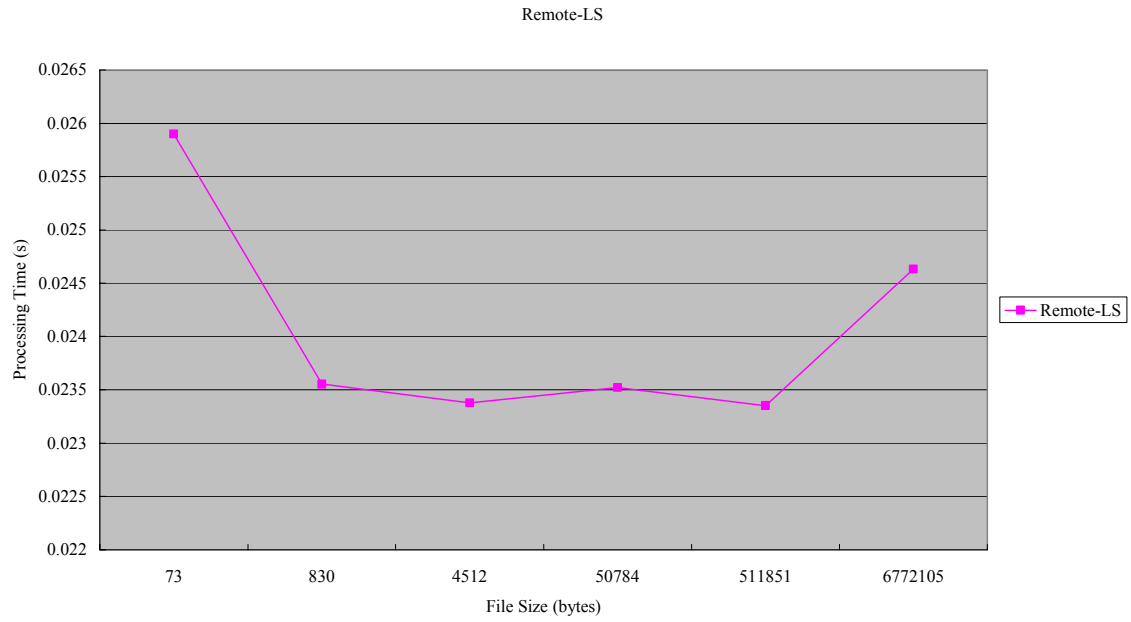


Fig. 39 shows the processing time for *remote* mail delivery to the *LS* version of mail server.

5.4.3 Discussions

As a conclusion of our experimental results, several comparisons are made between *normal* and *LS* versions of mail servers, and *local* and *remote* mail deliveries.

As shown in Fig. 40, we can see that under the processing of normal version of mail server, the processing time gets significantly larger when file size becomes larger than several megabytes regardless of local and remote mail deliveries.

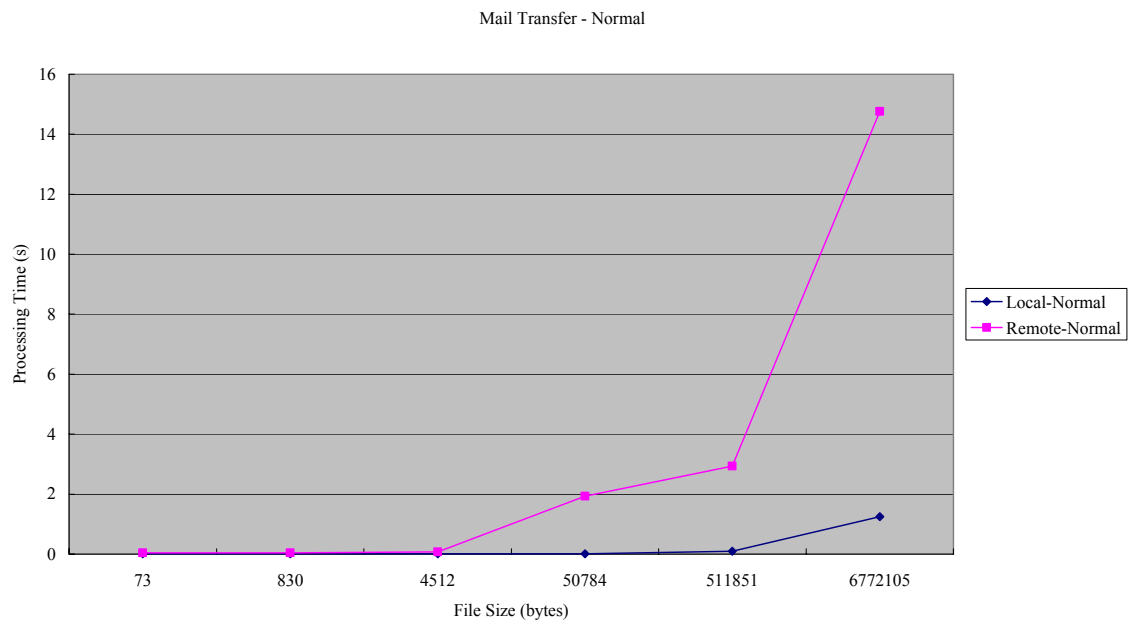


Fig. 40 shows the processing time for mail deliveries to *normal* version of mail server.

On the other hand, as shown in Fig. 41, the processing time doesn't increase with any significant level when file size gets very large. Therefore, the location server query and SMTP redirect mechanism can really significantly decrease server loads. Remote mail delivery will increase server load, but the impact is constantly stable.

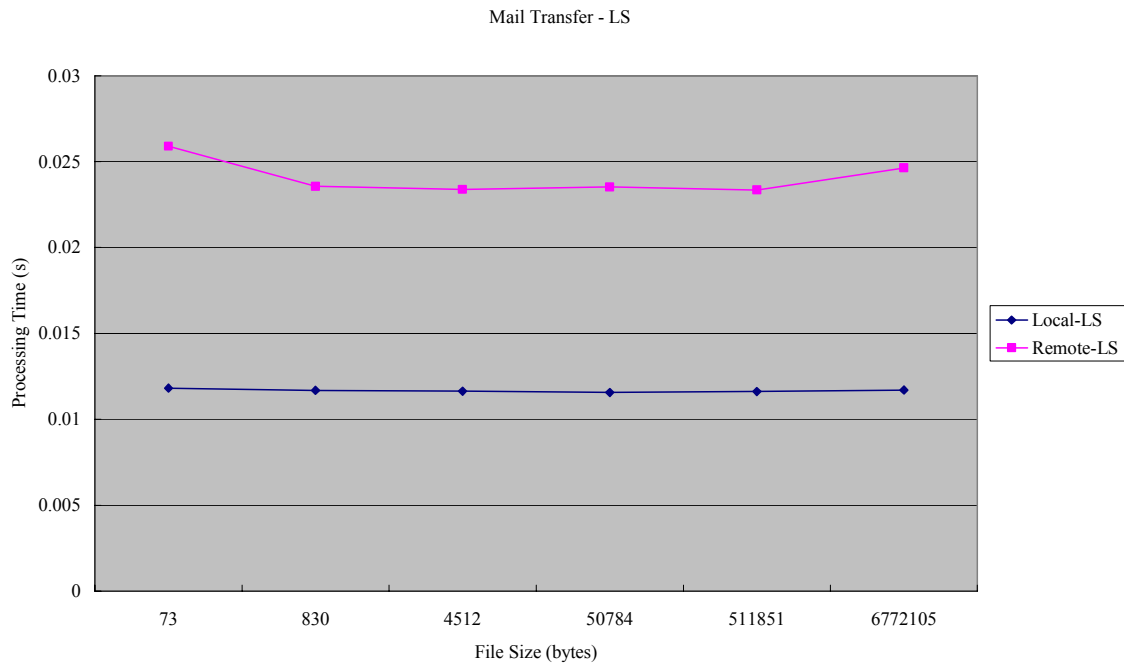


Fig. 41 shows the processing time for mail deliveries to *LS* version of mail server.

Finally, as shown in Fig. 42 and Fig. 43, we can see the benefits of *LS* version of mail server which adopts the location server query and SMTP redirect mechanism. No matter local or remote delivery is used, the scale of improvement is quite significant.

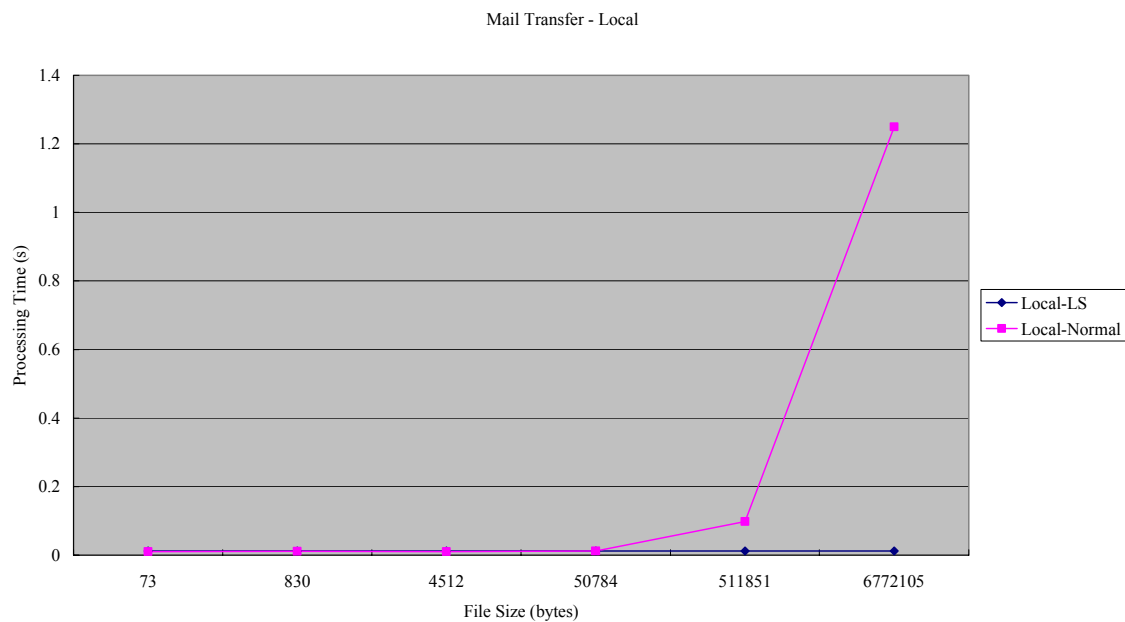


Fig. 42 shows the processing time for *local* mail delivery to both versions of mail servers.

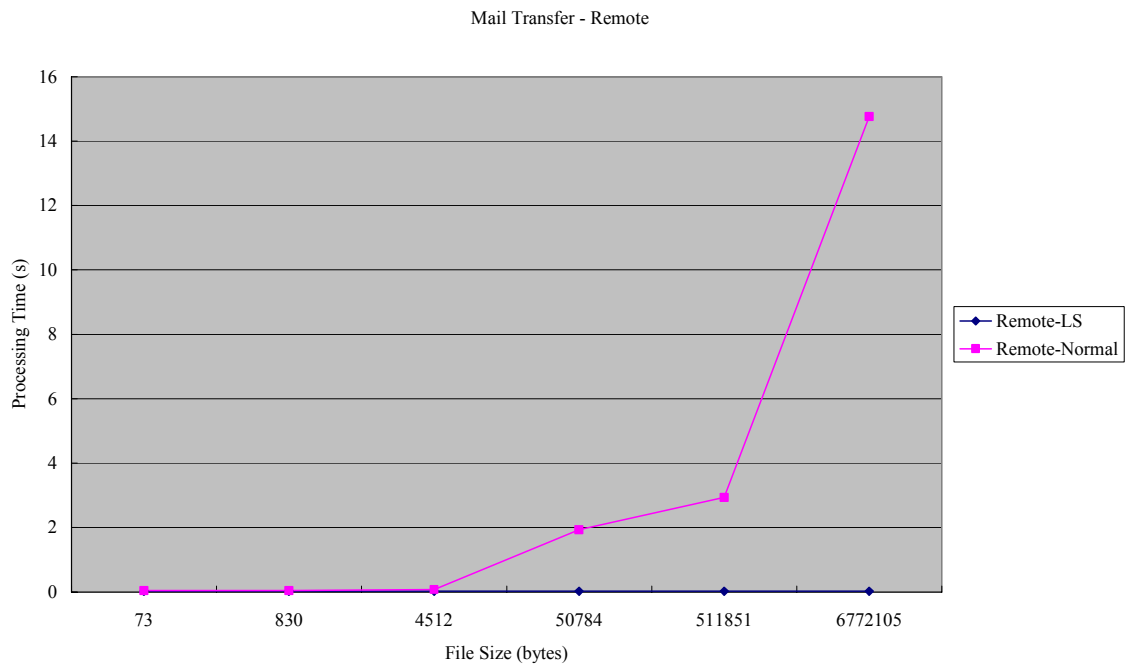


Fig. 43 shows the processing time for *remote* mail delivery to both versions of mail servers.

Chapter 6 Discussions and Conclusions

6.1 Discussions

In this section, several issues related to our infrastructure and management will be discussed. Firstly, in order to adopt our hybrid peer-to-peer architecture, several modifications have to be made to existing Internet protocols. For example, the functionality of DNS server can be extended by location server. One way is to add location records in DNS server. However, DNS protocol has to be modified to incorporate the processing of location records in DNS. The other way is to totally replace DNS by location service. The original DNS lookups are still available in location service queries. Moreover, new types of records like ULR and RLR also have to be provided.

In addition to existing protocols, the corresponding servers and clients may also have to be modified. For example in Chapter 5, mail transfer protocols like SMTP need to be modified to add peer-to-peer support. In order to bypass mail servers, DNS lookups in mail servers are changed into location server queries. Receiver MTA also needs to issue SMTP redirect messages to sender MTA. As another example, peer-to-peer support for file transfer and caching mechanism in Chapter 5 also requires modifications to HTTP/FTP/ICP protocols. Besides, HTTP/FTP/proxy servers also need modifications to support location service update and query.

As shown in Chapter 5, in order to add failover protection support for Internet applications, our architecture allows for precedence and fallback sequence configuration. Existing protocols such as SMTP are only a part of the universal messaging systems including e-mails, instant messages, and VoIP.

Secondly, the status of location server may not be completely up-to-date in situations when users disconnect abruptly from the network. In such circumstances, location server update is not possible since the sudden disconnection prevents clients from contacting location server. However, techniques like *polling* can be used for location server to ensure that a particular client is no longer alive.

However, even if there may be a short period of time when inconsistency exists, the operations of location service queries and updates will not be affected. For example, user *a* disconnects when user *b* tries to contact user *a*. From location service query, user *b* “thinks” that user *a* is still on-line at IP_a , so user *b* tries to connect to IP_a . But no reply will be returned since user *a* is actually off-line. If user *a* has configured a list of fallback sequence, then our infrastructure will not terminate the application immediately for not being able to contact user *a*. Instead, user *b* will try the next precedence from the list of fallback sequence for user *a*. And if no immediate way can be reached, e-mails will be sent as the last resort. Therefore, reliability is the benefit of our flexible failover protection mechanism.

Thirdly, will Location Server have single point of failure problem? The answer is no. The situation is different from that of a global centralized index server like Napster. In that case, server maintenance or failure causes global service disruption. In our architecture, since location server is deployed hierarchically as in the case of DNS server, a single server failure will not affect other parts of the world. No service disruption will ever result if replication of location server is available.

6.2 Concluding Remarks

In this dissertation, a hybrid peer-to-peer architecture was proposed to integrate and improve existing Internet application services. The focus of the architecture is on RAS (reliability, availability, and scalability). Reliability comes from the security mechanism deployed in the infrastructure, AAA (Authentication, Authorization, and Accounting) at resource allocation phase, and DHCP-based intranet management schemes. Availability comes from the failover protection mechanism where fallback sequence configuration is possible. Scalability comes from the hybrid peer-to-peer structure inherent in the architecture where dynamic addition, modification, and removal of mobile hosts are all managed in hierarchical location service and the flexibility of device capabilities, application types, and data attributes.

6.3 Future Works

By now, only several important applications like mail and file transfer are designed to work in our infrastructure. Since the architecture is flexible, in the future, more existing Internet applications can be augmented by peer-to-peer support like streaming audio/video and VoIP. Moreover, various services can be integrated into a universal messaging service. In this service, instant messaging, e-mails, VoIP, file transfer, and streaming audio/video playback are the individual part of the whole service where people can communicate, talk, and share files, etc. Since the architecture is also scalable, when new network technologies emerge, the transmission media could be changed, but the architecture is still able to operate, reliably and smoothly.

References

- [1] Klensin, J. Ed., “Simple Mail Transfer Protocol,” *RFC 2821*, IETF, April 2001.
- [2] Myers, J. and M. Rose, “Post Office Protocol – version 3,” *STD 53, RFC 1939*, IETF, May 1996.
- [3] Crispin, M., “Internet Message Access Protocol – Version 4rev1,” *RFC 2060*, IETF, December 1996.
- [4] Napster, available at: <http://www.napster.com/>.
- [5] GnuTella, available at: <http://www.gnutella.co.uk/>.
- [6] Milojevic, D.S., et. al., “Peer-to-Peer Computing,” *Technical Report HPL-2002-57*, HP Laboratories Palo Alto, March 2002.
- [7] Yang, B. and H. Garcia-Molina, “Comparing Hybrid Peer-to-Peer Systems,” *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB 2001)*, Roma, Italy, September 2001
- [8] IEEE, “IEEE Standard for Information Technology – Telecommunications and Information Exchange between System – Local and Metropolitan Area Networks – Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *ANSI/IEEE Std 802.11-1999 Edition*, 1999.
- [9] Perkins, C., “IP Mobility Support for IPv4,” *RFC 3220*, IETF, January 2002.
- [10] Clip2, “The GnuTella Protocol Specification v0.4, Document Revision 1.2,” *Internet article available at: <http://www.clip2.com/GnutellaProtocol04.pdf>*.
- [11] Mockapetris, P., “Domain Names – Implementation and Specification,” *STD 13, RFC 1035*, IETF, November 1987.
- [12] IEEE, “IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks,” *IEEE Std 802.1Q-1998*, December 1998.
- [13] IEEE, “Supplement to IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band,” *IEEE Std 802.11b-1999*, September 1999.
- [14] IEEE, “Supplement to IEEE Standard for Information Technology — Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks — Specific Requirements — Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band,” *IEEE Std 802.11a-1999*, September 2000.

- [15] Stubblefield, A., J. Ioannidis, and A. D. Rubin, "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP," *AT&T Labs Technical Report TD-4ZCPZZ*, available at: http://www.cs.rice.edu/~astubble/wep/wep_attack.pdf, August 2001.
- [16] Arbaugh, W.A., N. Shankar, and Y.C. Justin Wan, "Your 802.11 Wireless Network has No Clothes," available at: <http://www.cs.umd.edu/~waa/wireless.pdf>, March 2001.
- [17] Walker, J.R., "Unsafe at any Key Size; An Analysis of the WEP Encapsulation," *Technical Report 03628E, IEEE 802.11 Committee*, October 2000.
- [18] IEEE, "Draft Supplement to Standard for Telecommunications and Information Exchange between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)," *IEEE 802.11e Draft 2.0*, November 2001.
- [19] IEEE, "Draft Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation," *IEEE 802.11f Draft 2.0*, July 2001.
- [20] IEEE, "Draft Supplement to Standard for Telecommunications and Information Exchange between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical layer (PHY) Specifications: Specification for Enhanced Security," *IEEE 802.11i Draft 1.5*, August 2001
- [21] Guttman, E., C. Perkins, J. Veizades, and M. Day, "Service Location Protocol, Version 2," *RFC 2608*, IETF, June 1999.
- [22] Droms, R., "Dynamic Host Configuration Protocol," *RFC 2131*, IETF, March 1997.
- [23] Vixie, P., S. Thomson, Y. Rekhter, and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)," *RFC 2136*, IETF, April 1997.
- [24] Perkins, C. and D.B. Johnson, "Route Optimization in Mobile IP," *Internet Draft*, available at: <http://search.ietf.org/internet-drafts/draft-ietf-mobileip-optim-11.txt>, IETF, September 2001.
- [25] Leonhardt, U., J. Magee, and P. Dias, "Location Service in Mobile Computing Environments," *Computers and Graphics, Vol. 20, No. 5*, pp. 627-632, 1996.
- [26] Handley, M., H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol," *RFC 2543*, IETF, March 1999.
- [27] Moats, R., "URN Syntax," *RFC 2141*, IETF, May 1997.
- [28] T'Joens, Y., C. Hublet and P. De Schrijver, "DHCP Reconfigure Extension," *RFC 3203*, IETF, December 2001.
- [29] Wang, J.H. and T.L. Lee, "Enhanced Intranet Management in a DHCP-Enabled Environment," accepted and to appear in *Proceedings of the 26th Annual*

- International Computer Software and Applications Conference (COMPSAC 2002)*, Oxford, England, August 26-29, 2002.
- [30] IEEE, "IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Common Specifications – Part 3: Media Access Control (MAC) Bridges," *ANSI/IEEE Std 802.1D, 1998 Edition*, 1998.
- [31] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions," *RFC 2132*, IETF, March 1997.
- [32] Wimer, W., "Clarifications and Extensions for the Bootstrap Protocol," *RFC 1542*, IETF, October 1993.
- [33] Wang, J.H., T.L. Lee, and H.H. Lin, "Remote Host Configuration Protocol: Configuring a Remote Host in a User-Friendly Manner," *Proceedings of the 14th International Conference on Advanced Science and Technology (ICAST 98)*, pp. 303-314. Illinois, U.S.A., April 1998.
- [34] Wang, J.H. and T.L. Lee, "Extending DHCP with MAC-Layer User Authentication," *Proceedings of the 1st International Workshop on Software Engineering and Multimedia Applications (SEMA 99)*, pp. 151-155, Baden-Baden, Germany, August 1999.
- [35] Case, J., M. Fedor, M. Schoffstall, and J. Davin, "A Simple Network Management Protocol (SNMP)," *STD 15, RFC 1157*, IETF, May 1990.
- [36] 3Com Corporation, *SuperStack II Switch: Management Guide*, April 1999.
- [37] IEEE, "IEEE Standards for Local and Metropolitan Area Networks: Port based Network Access Control," *IEEE Std. 802.1X-2001*, June 2001.
- [38] Congdon, P., "IEEE 802.1X Overview: Port Based Network Access Control," *Internet article available at: <http://grouper.ieee.org/groups/802/1/mirror/8021/docs2000/P8021XOverview.PDF>*, March 2000.
- [39] Wang, J.H. and T.L. Lee, "Comparison between DHCP-based and IEEE 802.1x Network Access Control Mechanisms," accepted and to appear in *Proceedings of 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002)*, Orlando, USA, July 14-18, 2002.
- [40] Blunk, L. and J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)," *RFC 2284*, IETF, March 1998.
- [41] Mishra, A. and W.A. Arbaugh, "An Initial Security Analysis of the IEEE 802.1X Standard," *CS-TR-4328, UMIACS-TR_2002-10*, *Internet article available at: <http://www.cs.umd.edu/~waa/1x.pdf>*, February 2002.
- [42] Rigney, C., A. Rubens, W. Simpson, S. Willens, "Remote Authentication Dial In User Service (RADIUS)," *RFC 2865*, IETF, June 2000.
- [43] Congdon, P., et. al., "IEEE 802.1X RADIUS Usage Guidelines," *Internet Draft*

- available at: <http://www.ietf.org/internet-drafts/draft-congdon-radius-8021x-16.txt>, IETF, August 2001.
- [44] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," *RFC 2460*, IETF, December 1998.
- [45] Ranch, D.A., "Linux IP Masquerade HOWTO", *Linux Documentation Project article at: <http://www.tldp.org/HOWTO/IP-Masquerade-HOWTO/index.html>*, April 2002.
- [46] Rekhter, Y., B. Moskowitz, D. Karrenberg, G.J. de Groot, and E. Lear, "Address Allocation for Private Internets," *RFC 1918*, IETF, February 1996.
- [47] Vuksan, V., "DHCP mini-HOWTO", *Linux Documentation Project article at: <http://www.tldp.org/HOWTO/mini/DHCP/index.html>*, October 2000.
- [48] Grennan, M. "Firewall and Proxy Server HOWTO", *Linux Documentation Project article at: <http://www.tldp.org/HOWTO/Firewall-HOWTO.html>*, February 2000.
- [49] Cheswick, W.R. and S.M. Bellovin, *Firewalls and Internet Security: Repelling the Wily Hacker*, Addison-Wesley Pub. Co., June 1994.
- [50] Chapman, D.B. and E.D. Zwicky, *Building Internet Firewalls*, O'Reilly & Associates, Inc., November 1995.
- [51] Hornig, C., "Standard for the Transmission of IP Datagrams over Ethernet Networks," *RFC 894*, IETF, April 1984.
- [52] Comer, D.E., *Internetworking with TCP/IP, Vol. I: Principles, Protocols, and Architectures, Third Edition*, Prentice-Hall, Inc., 1995.
- [53] Stevens, W.R., *TCP/IP Illustrated, Vol. I: The protocols*, Addison-Wesley Pub. Co., 1994.
- [54] Postel, J. and J.K. Reynolds, "File Transfer Protocol," *RFC 959*, IETF, October 1985.
- [55] Horowitz, M. and S. Lunt, "FTP Security Extensions," *RFC 2228*, IETF, October 1997.
- [56] Wang, J.H. and T.L. Lee, "DHCP Enhancements for MAC-Layer User Authentication and Access Control," *Proceedings of the 9th IEEE International Conference on Telecommunications (ICT 2002)*, Beijing, China, June 23-26, 2002.
- [57] Russell, R., "Linux ipchains HOWTO, v1.0.8," *Internet article at: <http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html>*, July 2000.
- [58] Andreasson, O., "iptables Tutorial 1.1.9," *Internet article at: <http://people.unix-fu.org/andreasson/iptables-tutorial/iptables-tutorial.html>*, 2001.
- [59] Rusty et. al., "Kernel Hacking HOWTO v0.0.5," *Internet article at: <http://www.lisoleg.net/doc/Kernel-Hacking-HOWTO/>*, October 1999.
- [60] Johnson, M.K., "Linux Kernel Hackers' Guide," *Internet article at: <http://www.linuxdoc.org/LDP/khg/>*.

- [61] Beck, M., H. Boehme, M. Dziadzka, U. Kunitz, R. Magnus, and D. Verworner, *Linux Kernel Internals*, 2nd ed., Addison-Wesley, 1997.
- [62] Rubini, A., *Linux Device Drivers*, O'Reilly & Associates, Inc., 1998.
- [63] Jabber, available at: <http://www.jabber.org/>.
- [64] Oram, A., "Peer-to-Peer for Academia," *Internet article available at: http://www.openp2p.com/pub/a/p2p/2001/10/29/oram_speech.html*, October 2001.
- [65] Berg, S.R. and P. Guenther, "procmail," *available at: <http://www.procmail.org/>*.
- [66] Sendmail Inc., "Filtering mail with sendmail," *Internet article available at: http://www.sendmail.com/partner/resources/development/milter_api/*.
- [67] Sendmail, available at: <http://www.sendmail.org/>.
- [68] Costales, B. and E. Allman, *Sendmail*, 2nd ed., O'Reilly & Associates, Inc., January 1997.
- [69] Wahl, M., T. Howes, and S. Kille, "Lightweight directory access protocol (v3)," *RFC 2251*, IETF, December 1997.
- [70] OpenLDAP, available at: <http://www.openldap.org/>.
- [71] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," *RFC 3022*, IETF, January 2001.
- [72] Gaynor, M. and S. Bradner, "Firewall Enhancement Protocol," *RFC 3093*, IETF, April 2001.
- [73] Fielding, R. et. al., "Hypertext Transfer Protocol – HTTP/1.1," *RFC 2616*, IETF, June 1999.
- [74] Wessels, D. and K. Claffy, "Internet Cache Protocol (ICP), version 2," *RFC 2186*, IETF, September 1997.
- [75] Squid Web Proxy Cache, available at: <http://www.squid-cache.org/>.
- [76] Barish, G. and K. Obraczka, "World Wide Web Caching: Trends and Techniques," *IEEE Communication Magazine*, vol. 38, issue 5, pp. 178-185, May 2000.
- [77] Gauthier, P., J. Cohen, M. Dunsmuir, and C. Perkins, "Web Proxy Auto-Discovery Protocol (WPAD)," *Internet Draft, Internet article available at: <http://www.web-cache.com/Writings/Internet-Drafts/draft-ietf-wrec-wpad-01.txt>*, IETF, July 1999.
- [78] Netscape, "Navigator Proxy Auto-Config File Format," *Internet article available at: <http://wp.netscape.com/eng/mozilla/2.0/relnotes/demo/proxy-live.html>*, March 1996.
- [79] Cieslak, M., D. Forster, G. Tiwana, and R. Wilson, "Web Cache Communication Protocol V2.0," *Internet Draft*, IETF, April 2001.
- [80] Vixie, P. and D. Wessels, "Hyper Text Caching Protocol (HTCP/0.0)," *RFC 2756*, IETF, January 2001.
- [81] Valloppillil, V. and K.W.Ross, "Cache Array Routing Protocol v1.0," *Internet*

- Draft*, IETF, February 1998.
- [82] Russkov, A. and D. Wessels, "Cache Digests," *Proceedings of 3rd International WWW Caching Workshop*, April 1998.
 - [83] Fikouras, N.A. and C. Goerg, "Performance Comparison of Hinted and Advertisement Based Movement Detection Methods for Mobile IP Hand-offs," *Proceedings of the European Wireless 2000*, Dresden, Germany, September 2000.
 - [84] Perkins, C. and K.Y. Wang, "Optimized Smooth Handoffs in Mobile IP," *Proceedings of the fourth IEEE Symposium on Computers and Communications*, available at: <http://citeseer.nj.nec.com/perkins99optimized.html>, July 1999.
 - [85] El-Malki, K., et. al., "Low Latency Handoffs in Mobile IPv4," *Internet Draft, IETF Mobile IP Working Group, draft-ietf-mobileip-lowlatency-handoffs-v4-03.txt*, IETF, November 2001.
 - [86] Dommety, G., et. al., "Fast Handovers for Mobile IPv6," *Internet Draft, IETF Mobile IP Working Group, draft-ietf-mobileip-fast-mipv6-04.txt*, IETF, March 2002.
 - [87] De Carolis, A., L. Dell'Uomo, and F. Pugini, "QoS-Aware handover for Mobile IP: Secondary Home Agent," *Internet Draft, IETF Mobile IP Working Group, draft-decarolis-qoshandover-02.txt*, IETF, April 2000.
 - [88] Wedlund, E. and H. Schulzrinne, "Mobility Support Using SIP," *Proceedings of Second ACM/IEEE International Conference on Wireless and Mobile Multimedia (WoWMoM99)*, available at: <http://citeseer.nj.nec.com/486857.html>, Seattle, Washington, USA, August 1999.
 - [89] Deering, S., "ICMP Router Discovery Messages," *RFC 1256*, IETF, September 1991.
 - [90] Wang, J.H. and T.L. Lee, "An Enhanced DHCP-based Infrastructure for Intranet Management," *Proceedings of the 3rd International Conference on Internet Computing (IC 2002)*, Las Vegas, USA, June 24-27, 2002.